

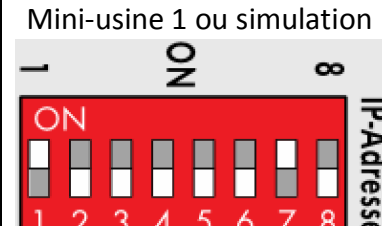
MANIP 6 : Exercices de gestion des sorties

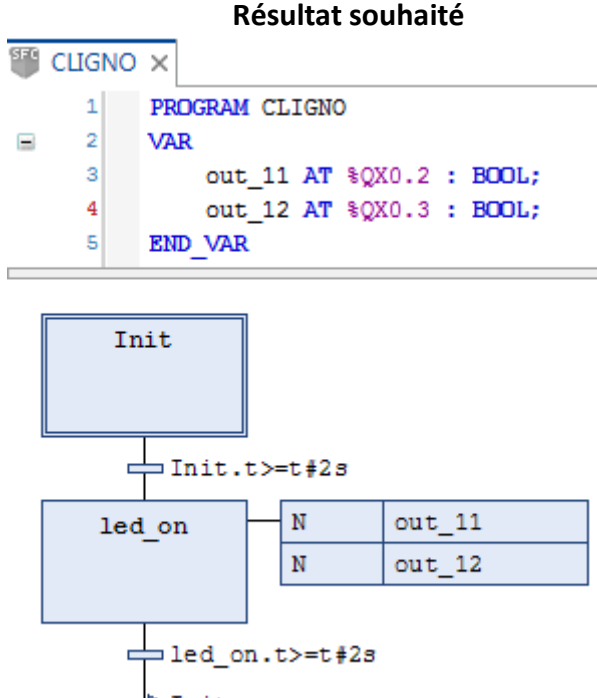
Objectifs

- Se familiariser avec l'environnement de programmation.
- Mettre en œuvre des notions supplémentaires liées au SFC.
- Être capable de résoudre des exercices simples agissant sur les sorties uniquement et de vérifier leurs bons fonctionnements grâce aux LEDS et aux outils de débogage.

A. Refaire la manipulation 5 à partir de zéro

Refaire la manipulation précédente par vous-même en créant un nouveau projet. Essayez, si possible, de ne pas vous aider des feuilles de la manip 5 mais de n'utiliser que les éléments ci-dessous.

Aide-mémoire		Adresse IP par défaut du PC	Adresse IP de l'automate en simulation	Codes d'accès par défaut
		192.168.1.192	192.168.1.65	admin wago

Résultat souhaité	Quelques explications
	<p>Pour les variables :</p> <ul style="list-style-type: none"> Q correspond à une sortie. X indique qu'il s'agit d'un bit. 0 correspond dans notre cas au port de sortie. .2 et .3 sont les bits 2 et 3 du port. <p>L'étape « Init » n'a pas d'action associée</p> <p>« Init.t » correspond à la durée écoulée depuis la dernière activation de « Init »,</p> <p>« Init.t >= t#2s » est une condition-transition qui deviendra vraie lorsque « Init » aura été active au moins 2 secondes.</p> <p>L'étape « led_on » est associée à deux actions qui seront effectuées en continu tant que l'étape est active.</p> <p>La séquence reboucle sur « Init ».</p>

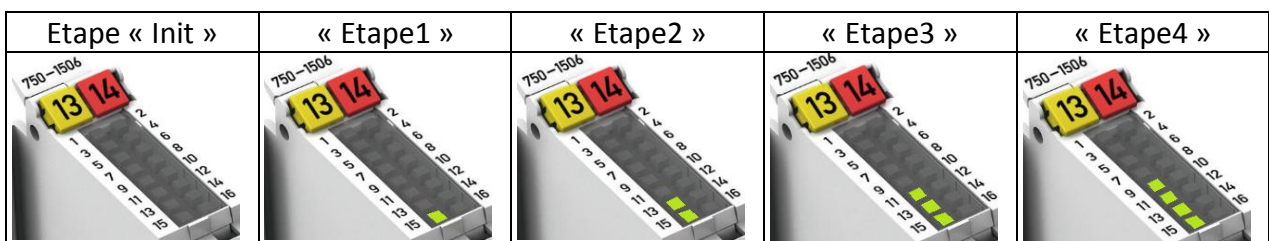
MANIP 6 : Exercices de gestion des sorties

B. Réalisation d'un bargraphe

Créez un nouveau projet au sein d'un dossier dédié que vous nommez par exemple « manip6b ». Sauvez ce projet avec un nom évocateur tel que « bargraphe ».

Le programme sera réalisé au sein d'un SFC comprenant 5 étapes.

Nommez les étapes telles qu'indiquées ci-dessous et arrangez vous pour que votre séquence active successivement ces étapes dans l'ordre avec une durée de 1 seconde chacune.

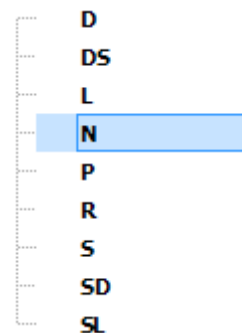


Par souci de clarté, nommez les sorties 15, 13, 11 et 9 par les identifiants respectifs led1, led2, led3 et led4.

C. Bargraphe avec qualificatifs R et S

Lorsque nous avons associé des actions à une étape, nous avons utilisé des actions précédées de la lettre « N ».

Il est cependant possible d'utiliser d'autres qualificatifs pour les actions comme illustré ci-contre.



La lettre « N » est le qualificatif le plus courant, il précise que l'action demeure active aussi longtemps que l'étape demeure active.

Nous allons essayer de garder exactement la même fonctionnalité que l'exercice précédent mais en utilisant cette fois les qualificatifs R et S :

- Le S (SET) activera l'action lors de l'activation de l'étape associée et maintiendra cette action active jusqu'à l'apparition d'un reset.
- Le R (RESET) désactivera l'action au moment de l'activation de l'étape associée.

Si on regarde par exemple la led1 du bargraphe, on remarque qu'elle doit rester allumée au sein des étapes 1, 2, 3 et 4. On pourrait alors appliquer un SET de la led1 à l'étape 1 et un RESET après l'étape4.

On choisira d'ajouter une étape5 (suivie d'une condition-transition « TRUE ») pour réaliser le reset ; on évite ainsi d'associer une action à l'étape initiale « Init ».

MANIP 6 : Exercices de gestion des sorties

Description des qualificatifs :

Qualificatif		
Pour associer des actions à des étapes CEI, vous disposez des qualificatifs (qualifier) suivant:		
N	Non-stored (non mémorisé)	l'action demeure active aussi longtemps que l'étape demeure active
R	overriding Reset (remise à zéro prioritaire)	désactivation de l'action
S	Set (Stored) (positionné (mémorisé))	activation de l'action, qui demeure ensuite active jusqu'au prochain reset
L	time Limited (limite dans le temps)	activation de l'action pendant une durée déterminée
D	time Delayed (temporisé)	activation de l'action après un certain temps, pour autant que l'étape demeure active
P	Pulse (impulsion)	l'action est exécutée exactement une fois, lorsque l'étape est activée
SD	Stored and time Delayed (mémorisé et temporisé)	activation de l'action après un certain temps; l'action demeure ensuite active jusqu'au prochain reset
DS	Delayed and Stored (temporisé et mémorisé)	activation de l'action après un certain temps, pour autant que l'étape demeure active; l'action demeure alors active jusqu'au prochain reset
SL	Stored and time Limited (mémorisé et limité dans le temps)	activation de l'action pendant une durée déterminée

Les qualificatifs L, D, SD, DS et SL doivent être accompagnés d'une spécification de temps sous format de constante TIME, p.ex. L T#5s.

Document Version: 3.2, CoDeSys V2.3.7.0. - Edition: 29.09.2006. - p. 30 / 468

Un autre qualificatif intéressant est le P (PULSE) qui précise que l'action ne sera exécutée qu'une seule fois au moment de l'activation de l'étape associée.

Cette situation est nécessaire par exemple si l'on souhaite incrémenter une variable lors de l'activation d'une étape ; il faut bien sûr que l'incrémenter ne se fasse pas en continu !

Il est recommandé d'éviter d'utiliser les qualificatifs SET et RESET pour des actions classiques sur les sorties.

Mis à part que l'usage de SET/RESET mobilise de l'espace mémoire supplémentaire, il faut garder à l'esprit que cela rend le débogage plus difficile. En effet, si on a par exemple une sortie à « 0 » alors qu'elle devrait être à « 1 », on ne sait pas si on a omis un SET ou si on a placé un RESET à un endroit qui ne devrait pas être. De façon similaire, le débogage est plus difficile si la sortie est à « 1 » alors qu'elle devrait être à « 0 ».

MANIP 6 : Exercices de gestion des sorties

D. Bargraphes gauche et droit avec une structure en ET

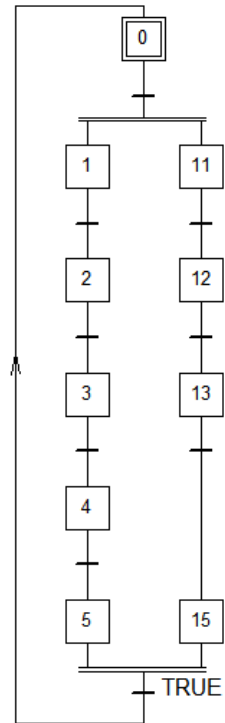
L'exercice consiste à mettre en œuvre la structure ci-contre.

La branche contenant les étapes 1 à 5 se charge de réaliser un bargraphe de 4 leds situées à gauche du module d'entrée/sortie. Il s'agit, comme pour l'exercice « B », des sorties 15, 13, 11 et 9. On les nommera respectivement led1g, led2g, led3g et led4g.

La branche contenant les étapes 11 à 15 se charge de réaliser un bargraphe de 3 leds situées à droite du module d'entrée/sortie. Il s'agit des sorties 16, 14 et 12 qu'on nommera respectivement led1d, led2d et led3d.

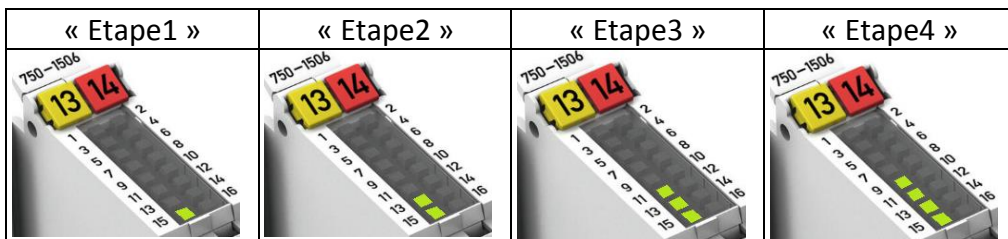
Les étapes 5 et 15 sont des étapes d'attente. D'ailleurs un ET convergent apparaît qu'après des étapes et non des conditions-transition.

Le nombre de sorties pilotées par chaque branche est volontairement différent afin d'avoir des durées différentes et bien constater le fonctionnement de la structure en ET.

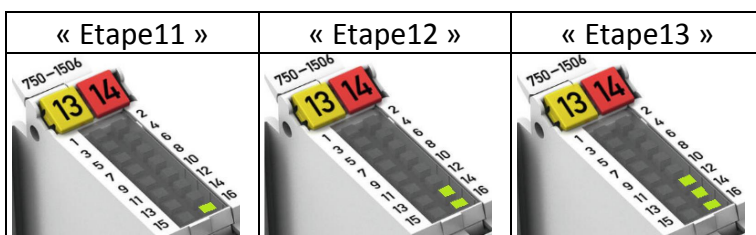


L'étape « 0 » correspond à la situation initiale où toutes les leds sont éteintes. Vous devez prendre en compte une durée de 1 seconde pour les conditions-transition.

Les sorties pilotées par la branche de gauche doivent évoluer comme ci-dessous :



Les sorties pilotées par la branche de droite doivent évoluer comme ci-dessous :



Vous devriez constater une exécution simultanée des différentes branches.

MANIP 6 : Exercices de gestion des sorties

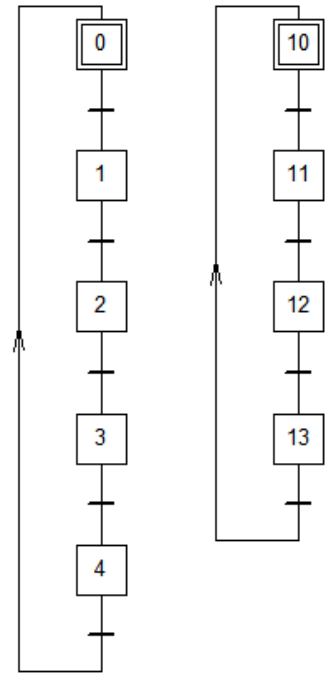
E. Bargraphes gauche et droit par des séquences indépendantes

L'exercice consiste à mettre en œuvre la structure ci-contre.

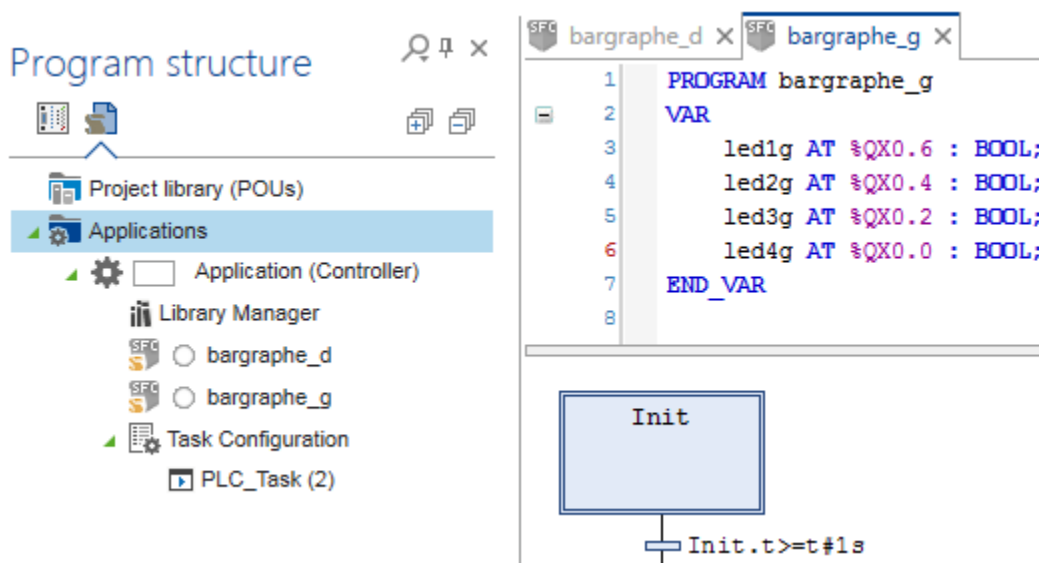
La branche contenant les étapes 0 à 4 se charge de réaliser un bargraphe de 4 leds situées à gauche du module d'entrée/sortie. La branche contenant les étapes 10 à 13 se charge de réaliser un bargraphe de 3 leds situées à droite du module d'entrée/sortie.

Les séquences s'exécuteront toutes les deux de manière indépendante. On pourrait s'attendre à un comportement similaire à l'exercice « D »... cependant, si différence il y a, tâchez de la constater et de l'expliquer.

Les durées à prendre en compte sont toujours d'une seconde. Les étapes initiales « 0 » et « 10 » ne sont associées à aucune action.



Il semblerait, à priori, qu'on ne puisse pas mettre plusieurs séquences indépendantes sur le même SFC (contrairement à Unity de Schneider par exemple). On optera alors pour la création d'un nouveau programme (POU) de manière à avoir une séquence au sein de chaque SFC (comme illustré ci-dessous avec bargraphe_g et bargraphe_d) :



```

1  PROGRAM bargraphe_g
2  VAR
3      led1g AT %QX0.6 : BOOL;
4      led2g AT %QX0.4 : BOOL;
5      led3g AT %QX0.2 : BOOL;
6      led4g AT %QX0.0 : BOOL;
7  END_VAR
8

```

Init

Init.t >= t #1s

On veillera à ce que le « PLC_Task » contienne bien l'appel des deux POU's !

MANIP 6 : Exercices de gestion des sorties

F. Contrôler l'ensemble des sorties à un seul endroit

On a constaté que, pour des raisons de débogage, il est plus facile de regrouper l'ensemble des commandes sur les sorties au sein d'un seul fichier.

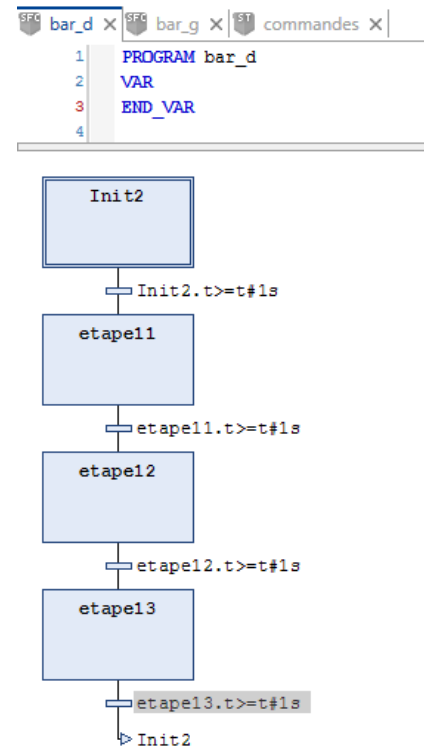
Ainsi par exemple, si une foreuse ne tourne pas, il suffit de consulter l'endroit où cette foreuse est commandée au sein du fichier. Il n'est plus nécessaire d'examiner toutes les actions associées aux étapes des différents SFC !

L'illustration ci-contre montre un onglet supplémentaire « commandes » qui contiendra les commandes des sorties.

On voit également un SFC modifié où l'on ne retrouve plus aucune action associée.

On peut remarquer également qu'on n'a plus de variables manipulées au sein du SFC, celles-ci seront manipulées au sein du programme « commandes ».

Nous avons choisi le langage ST pour les commandes » car c'est un langage concis mais nous aurions pu choisir le LD.

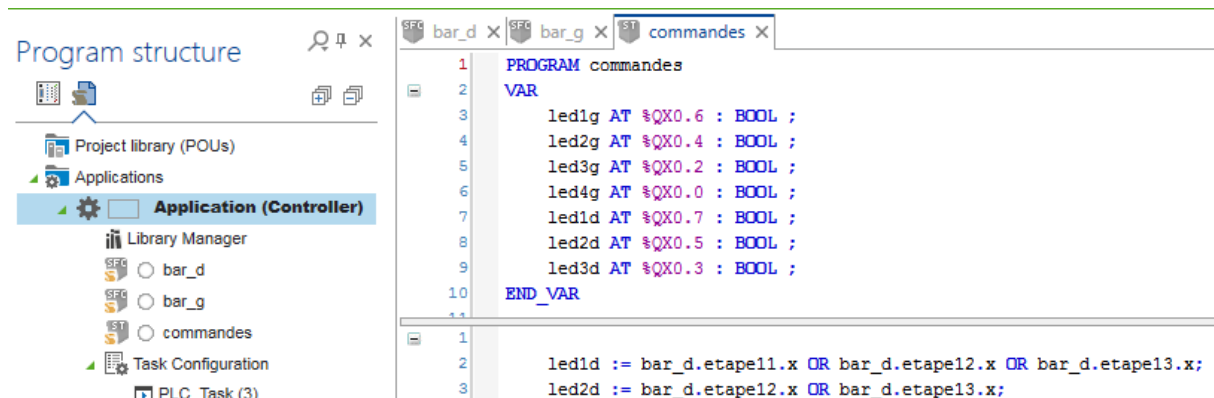


L'implémentation du programme « commande » :

Comme pour le SFC, le programme apparaît en deux parties :

- La partie supérieure → On y déclarera les variables manipulées
- La partie inférieure → On y écrira les commandes en ST (Structured Text)

Ci-dessous, une partie du programme ST gérant les sorties :



Tentez de comprendre la philosophie de la gestion des sorties afin modifier votre exercice précédent pour gérer toutes vos sorties à un seul endroit.

MANIP 6 : Exercices de gestion des sorties

Le langage ST est un langage littéral permettant, entre autres, d'affecter une valeur à une variable.

L'opérateur d'affectation est **:=**
Les lignes de commandes se terminent habituellement par un **;**
L'opérateur logique ou est noté **OR**

Etat actif ou non d'une étape :**Etape active**

Après l'appel du module SFC, l'action correspondant à l'étape initiale (entourée en double) sera ensuite exécutée. Une étape dont l'action sera exécutée, est appelée active. En mode En Ligne, les étapes actives sont représentées en bleu.

A chaque étape correspond un drapeau, lequel mémorise l'état de l'étape. Le drapeau d'étape (état actif ou inactif de l'étape) sera représenté par la valeur logique d'un élément booléen <nom_du_pas>.x. Cette variable booléenne a la valeur TRUE (vraie) quand l'étape correspondante est active et FALSE (fausse) quand elle est inactive. Cette variable est déclarée implicitement et peut être utilisée dans toutes les actions et transitions du module SFC.

Dans un cycle de commande, toutes les actions appartenant à des étapes actives sont exécutées. Ensuite, les étapes qui suivent les étapes actives deviennent actives quand les conditions de transition des étapes suivantes sont TRUE. Les étapes à présent actives ne sont exécutées qu'au prochain cycle.

Remarque : Si l'étape active contient une action de sortie, celle-ci ne sera exécutée qu'au prochain cycle, pour autant que la transition suivante ait la valeur TRUE.

- Pour les actions CEI : Les variables de temps implicites ne sont pas utilisées.

Ces « drapeaux » d'état peuvent être utilisés au sein de chaque action et transition du module SFC. Mais il est également possible d'accéder à ces flags à partir d'autres programmes :

Exemple: `boolvar1:=sfc.step1.x;`

`step1.x` est ici une variable booléenne implicite qui représente l'état de l'étape CEI `step1` dans le module `sfc1`.

Document Version: 3.2, CoDeSys V2.3.7.0. - Edition: 29.09.2006. - p. 28 / 468 et p. 31 / 468

Exemple d'expression booléenne utilisée dans notre cas : **bar_d.etape12.x**

Nous préférons gérer les commandes à l'aide d'un programme ST plutôt que le LD car il est plus concis et plus rapidement modifiable. Cependant si vous avez le temps, vous pouvez essayer de l'implémenter en LADDER comme exercice.

G. Implémentation des commandes en LADDER (LD)

Remplacez votre programme ST par un programme LD en gardant les mêmes fonctionnalités. La fonction OU se réalise en plaçant les éléments en parallèle.

Les sorties seront représentées par des bobines et les états de étapes par des contacts.