

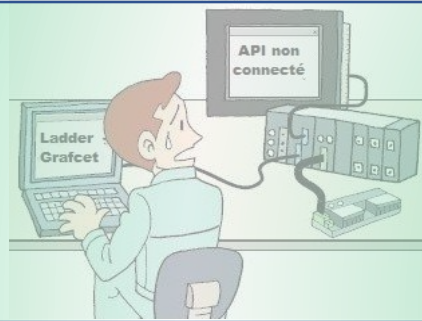
SEMAINE 5

INITIATION AUX AUTOMATES SIEMENS

FICHE 41 : PROGRAMMATION D'UN CAHIER DES CHARGES EN LADDER



Automation & Sense



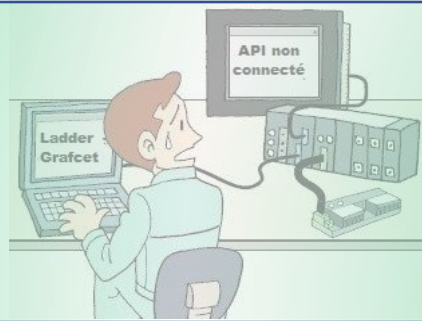
Nota Bene :



Cette fiche est une proposition de solution au cahier des charges donné au niveau de la fiche 39.

Elle a pour objectif de vous initier à la programmation en langage Ladder sur TIA Portal.

Vu que c'est une proposition de solution, elle pourra être différente de la vôtre.

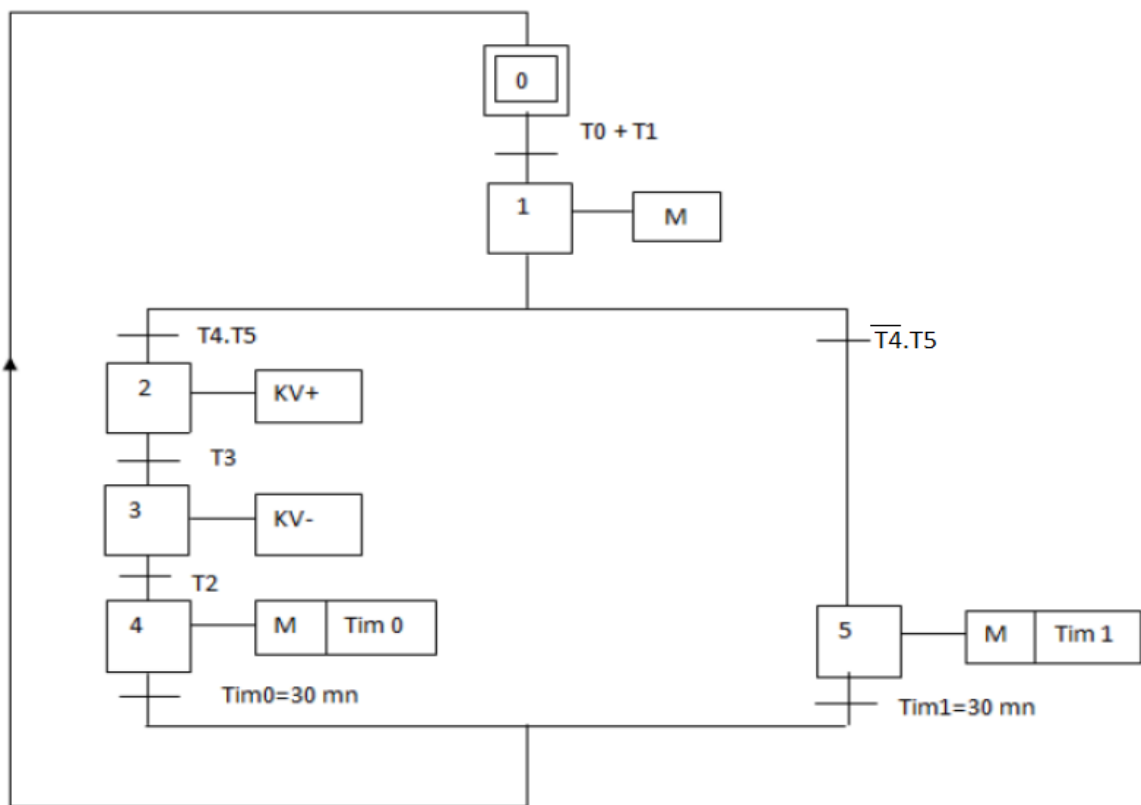


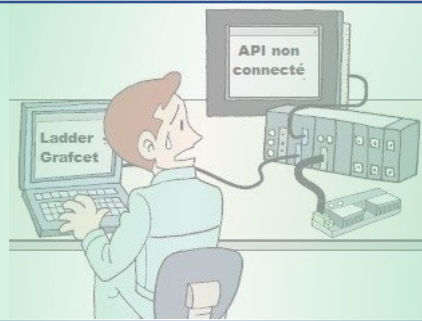
Dans la précédente fiche, nous avons vu comment programmer un cahier des charges sur TIA Portal en utilisant le langage Grafcet, dans cette fiche, nous programmerons le même cahier des charges mais cette fois-ci en langage Ladder.

Pour cela, nous établirons dans un premier temps les équations logiques du système puis dans un second temps nous créerons un nouveau projet et concevrons le programme Ladder.

I) Etablissement des équations logiques

Le grafcet du cahier des charges a été légèrement modifié (concernant la transition permettant d'activer l'étape 5 du grafcet). Cette modification permet d'avoir un grafcet plus propre. En effet, l'ancien grafcet pouvait présenter un petit problème vu que les deux transitions juste après la divergence en OU pouvaient être vraies en même temps. Ci-dessous le nouveau grafcet.





Nous avons vu dans les précédentes fiches comment établir les équations logiques d'un cahier des charges à partir d'un grafcet. Nous allons pour concevoir notre programme Ladder établir les équations des conditions d'activation des différentes étapes du grafcet. Soit **A_Ei** la condition d'activation de l'étape **i** du grafcet, nous aurons les équations suivantes :

$$A_E0 = (A_E4.Tim0 + A_E5.Tim1 + Init)$$

$$A_E1 = (A_E0 . (T0 + T1))$$

$$A_E2 = (A_E1.T4.T5)$$

$$A_E3 = (A_E2.T3)$$

$$A_E4 = (A_E3.T2)$$

$$A_E5 = (A_E1.\overline{T4}.T5)$$

$$M = A_E1 + A_E4 + A_E5$$

$$KV+ = A_E2$$

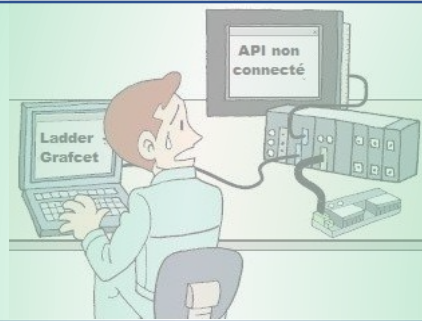
$$KV- = A_E3$$

$$TIM0 = A_E4$$

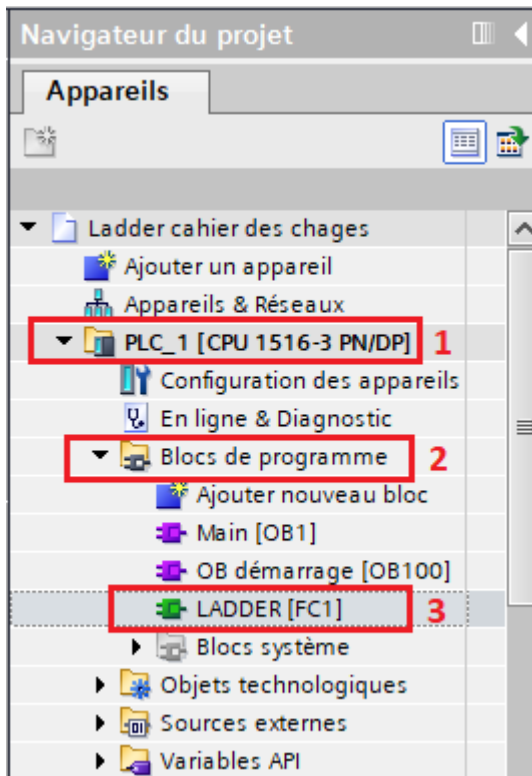
$$TIM1 = A_E5$$

II) Implémentation du programme Ladder dans l'automate

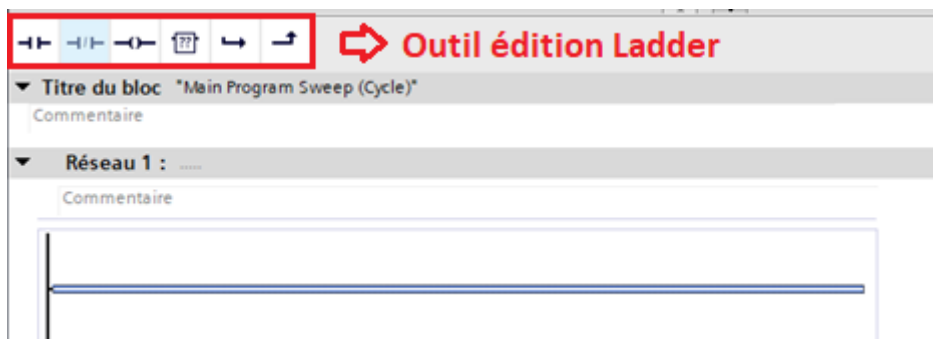
Pour implémenter le programme Ladder dans notre automate, nous allons créer un nouveau projet. Cette fois ci, nous allons choisir un CPU de type **S7-1500**. Pour cela, après avoir ouvert TIA Portal, nous allons cliquer sur « **Ajouter un nouveau appareil** », choisir notre CPU et lui donner un nom. (Voir les fiches précédentes).

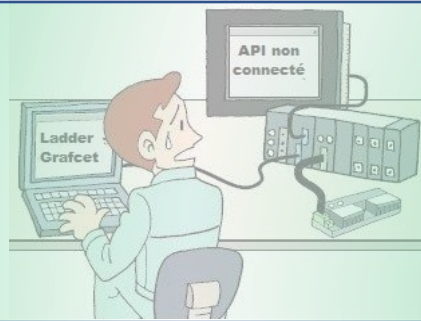


Une fois le CPU choisi, dans la vue projet, nous allons cliquer sur celui-ci, puis sur « **Blocs de programme** », nous ajouterons un bloc FC que l'on appellera LADDER. C'est dans ce bloc FC que l'on rentrera notre programme Ladder

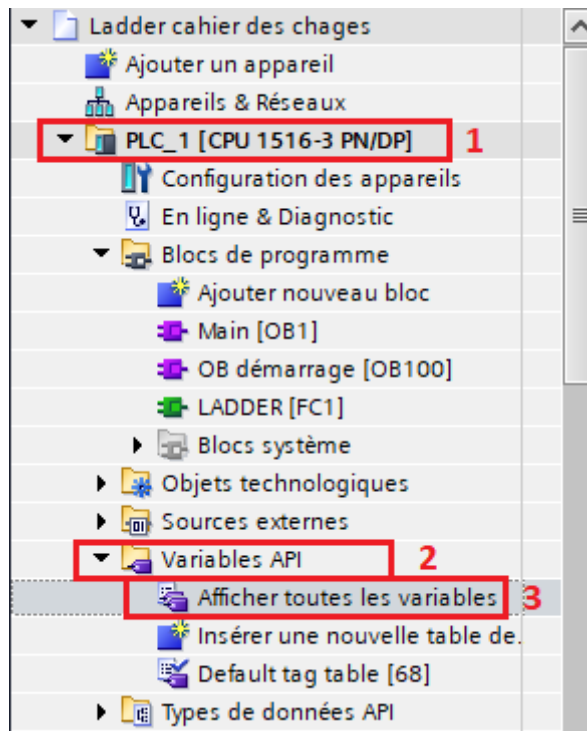


Par la suite, nous allons cliquer sur « **LADDER [FC1]** » et utiliser l'outil d'édition de Ladder pour rentrer notre programme.



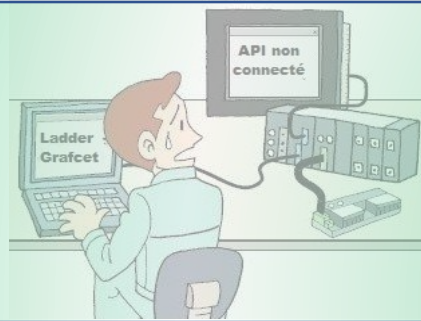


Vu que nous aurons à utiliser des variables dans notre programme Ladder, nous allons donc les déclarer au préalable au niveau de la table des variables. Pour cela, nous allons cliquer sur « **Variables API** » puis sur « **Afficher toutes les variables** ».



Par la suite nous déclarerons toutes nos variables. Il faut noter que parmi nos variables, nous avons des entrées digitales et des sorties digitales. Ce sont des variables de types booléennes et qui ne peuvent prendre que deux valeurs soit « 0 », soit « 1 ». Sur TIA Portal, les entrées digitales sont adressées suivant cette forme : **%IB.i** alors que les sorties digitales sont adressées suivant la forme **%QB.i**. Avec **B** désignant le numéro de l'octet dans lequel est située la variable et le **i** la position exacte de la variable au sein de l'octet.

Pour rappel, un octet est composé de 8 bits. Ce qui fait que si on prend par exemple l'octet 0 comme octet comportant les entrées physiques de l'automate, on aura les variables suivantes %I0.0, %I0.1, %I0.2 etc... jusqu' à %I0.7. Même chose pour les sorties digitales, on aura de %Q0.0 à %Q0.7. Les sorties digitales sont constituées des actionneurs comme les électrovannes



servant à commander le vérin et le moteur du tapis alors que les entrées digitales sont constituées des capteurs et des commutateurs.

Variables API									
	Nom	Table des variables ▲	Type de données	Adresse	Réma...	Acces...	Ecritu...	Visibl...	Commentaire
1	M	Default tag table	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Moteur Tapis
2	KV+	Default tag table	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Electrovanne sortie vérin
3	KV-	Default tag table	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Electrovanne rentrée vérin
4	T0	Default tag table	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Commutateur cycle auto
5	T1	Default tag table	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Bouton poussoir cycle manu
6	T2	Default tag table	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Capteur tige vérin rentrée
7	T3	Default tag table	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Capteur tige vérin sortie
8	T4	Default tag table	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Capteur position haute
9	T5	Default tag table	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Capteur position basse
10	INIT	Default tag table	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Initialisation grafcet
11	Tim0	Default tag table	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Timer 0
12	Tim1	Default tag table	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Timer 1
13	ETAPEG7	Default tag table ▼	Int	%MW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Mot contenant position grafcet
14	<Ajouter>								

Identificateur d'opérande : M

Type d'opérande : W

Adresse : 2

Nous allons maintenant passer à l'édition du programme Ladder. Pour cela, nous allons nous baser sur les équations logiques que nous avons précédemment établies.

Remarques :

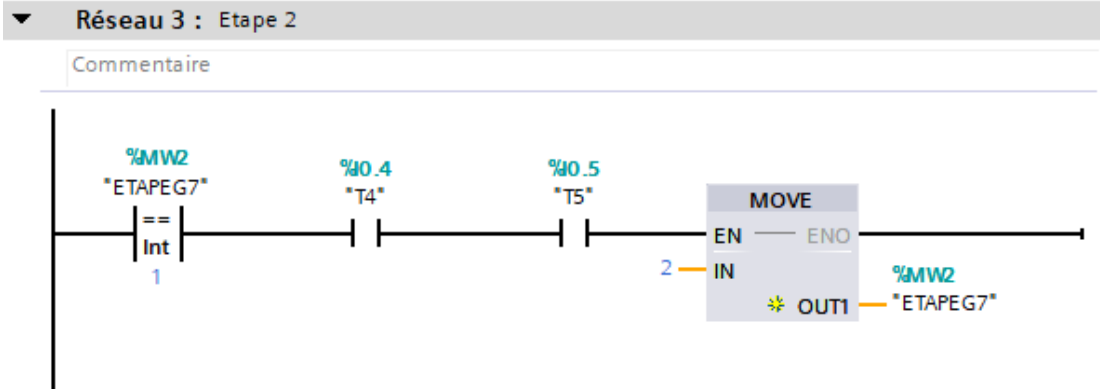
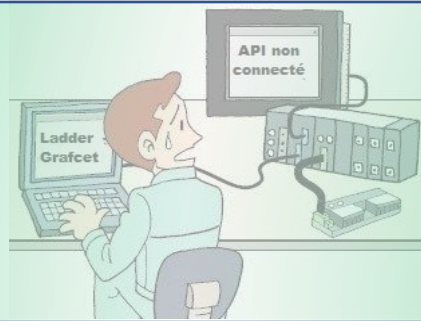
Nous avons utilisé un « momento » nommée « Init » pour l'initialisation du grafcet étant donné que l'étape initiale d'un grafcet est active au repos.

Aussi, nous avons utilisé un mot (Word) nommé **ETAPEG7** dans lequel nous allons enregistrer l'étape active du grafcet grâce à l'instruction **MOVE**.

Par exemple si l'étape 2 du grafcet est active, traduit par l'équation suivante :

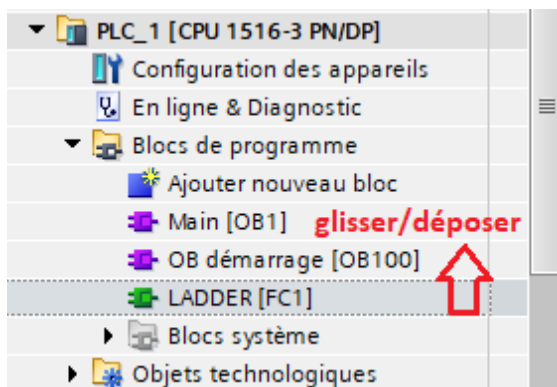
$$A_E2 = (A_E1.T4.T5)$$

Nous allons faire un transfert de la valeur 2 (étape 2) vers le mot **%MW2** comme représenté sur l'image ci-dessous.

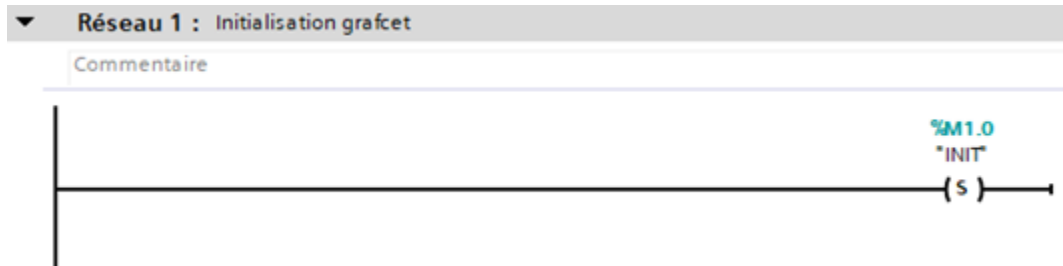
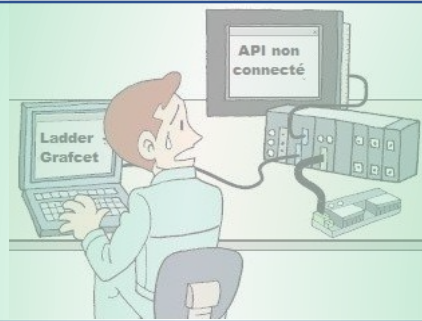


Nous allons faire pareil pour toutes les équations du Grafcet.

A la fin de l'édition du programme Ladder il ne faut pas oublier d'appeler le **FC1** au niveau de l'**OB1**. Pour cela, il faudra faire un simple glisser-déposer de FC1 vers OB1.



NB : Il faudra créer un OB de démarrage **OB100** qui sera exécuté une seule fois au démarrage de l'automate. Dans cet OB100, nous ferons un SET de « INIT » afin d'activer l'étape 0 du grafcet au début.



Puis tout à la fin du bloc OB1, nous ferons le RESET de « INIT »

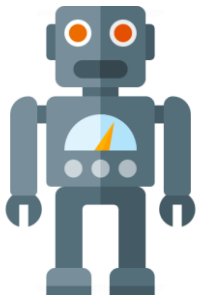
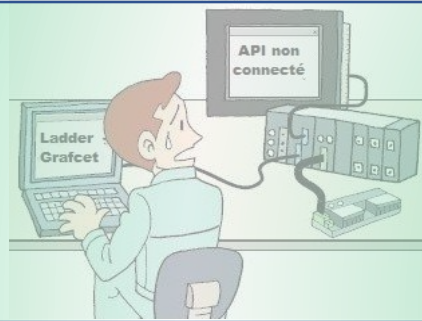


III) Simulation et test du programme

Pour simuler votre programme, démarrer PLCSIM puis chargez votre programme dans celui-ci. Vous pourrez ainsi utiliser la « **Table de visualisation et de forçage** » pour procéder à vos simulations.

Félicitations !!! Vous venez de créer votre premier programme Ladder sur TIA Portal

Vous trouverez sur l'espace de formation le programme Ladder complet en fichier PDF que vous pourrez télécharger.



Dans cette fiche, nous avons pu voir comment programmer les équations logiques d'un grafcet en langage Ladder.

Vous procéderez de la même manière pour tous les autres programmes que vous aurez à concevoir.