

Siemens Automation Cooperates with Education | 05/2017

Module 032-300 TIA Portal

Temporisations et compteurs CEI Les multi-instances dans SIMATIC S7-1500





Packages SCE pour formateurs adaptés à ces dossiers de formation

Automates SIMATIC

■ SIMATIC ET 200SP Open Controller CPU 1515SP PC F et HMI RT SW

N° d'article: 6ES7677-2FA41-4AB1

SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety

N° d'article: 6ES7512-1SK00-4AB2

SIMATIC CPU 1516F PN/DP Safety

N° d'article: 6ES7516-3FN00-4AB2

SIMATIC S7 CPU 1516-3 PN/DP

N° d'article: 6ES7516-3AN00-4AB3

SIMATIC CPU 1512C PN avec logiciel et PM 1507

N° d'article : 6ES7512-1CK00-4AB1

SIMATIC CPU 1512C PN avec logiciel, PM 1507 et CP 1542-5 (PROFIBUS)

N° d'article: 6ES7512-1CK00-4AB2

SIMATIC CPU 1512C PN avec logiciel

N° d'article: 6ES7512-1CK00-4AB6

SIMATIC CPU 1512C PN avec logiciel et CP 1542-5 (PROFIBUS)

N° d'article: 6ES7512-1CK00-4AB7

SIMATIC STEP 7 Software for Training

SIMATIC STEP 7 Professional V14 SP1- Licence monoposte

N° d'article: 6ES7822-1AA04-4YA5

SIMATIC STEP 7 Professional V14 SP1 - Licence salle de classe 6 postes

N° d'article: 6ES7822-1BA04-4YA5

SIMATIC STEP 7 Professional V14 SP1- Licence de mise à niveau 6 postes

N° d'article: 6ES7822-1AA04-4YE5

SIMATIC STEP 7 Professional V14 SP1 - Licence salle de classe 20 postes

N° d'article: 6ES7822-1AC04-4YA5

Veuillez noter que les packages pour formateurs ont parfois été remplacés par de nouveaux packages.

Vous pouvez consulter les packages SCE actuellement disponibles sous : siemens.com/sce/tp

Formations

Pour les formations Siemens SCE régionales, contactez votre interlocuteur SCE régional siemens.com/sce/contact

Plus d'informations sur le programme SCE

siemens.com/sce

Remarque d'utilisation

Les dossiers de formation SCE pour la solution d'automatisation cohérente Totally Integrated Automation (TIA) ont été spécialement créés pour le programme "Siemens Automation Cooperates with Education (SCE)" à des fins de formation pour les instituts publics de formation et de R&D. Siemens AG n'assume aucune responsabilité quant au contenu.

Cette documentation ne peut être utilisée que pour une première formation aux produits/systèmes Siemens. Autrement dit elle peut être copiée, en partie ou en intégralité, pour être distribuée aux participants à la formation afin qu'ils puissent l'utiliser dans le cadre de leur formation. La diffusion et la duplication de cette documentation, l'exploitation et la communication de son contenu sont autorisées au sein d'instituts publics de formation et de formation continue.

Toute exception requiert au préalable l'autorisation écrite de la part des interlocuteurs Siemens AG : Monsieur Roland Scheuerer <u>roland.scheuerer@siemens.com</u>.

Toute violation de cette règle expose son auteur au versement de dommages et intérêts. Tous droits réservés, en particulier en cas de délivrance de brevet ou d'enregistrement d'un modèle déposé.

Il est expressément interdit d'utiliser cette documentation pour des cours dispensés à des clients industriels. Tout usage de cette documentation à des fins commerciales est interdit.

Nous remercions l'Université technique de Dresde, en particulier Prof. Dr.-Ing. Leon Urbas et l'entreprise Michael Dziallas Engineering ainsi que toutes les personnes ayant contribué à la réalisation des dossiers de formation.

Sommaire

1	Ob	bjectif	5
2	Co	onditions requises	5
3	Co	onfigurations matérielles et logicielles requises	6
4	Th	néorie	7
	4.1	Instances et multi-instances dans SIMATIC S7-1500	7
	4.	1.1 Blocs de données d'instance/Instances uniques	8
	4.	1.2 Multi-instances	9
5	Ér	noncé du problème	11
6	Pla	anification	11
	6.1	Mode automatique - Moteur du convoyeur avec fonction de temporisation	11
7	Ins	structions structurées par étapes	12
	7.1	Désarchiver un projet existant	12
	7.2	Ajout d'une temporisation CEI TP dans le bloc fonctionnel FB1 "MOTOR_AUTO"	14
	7.3	Mise à jour de l'appel de bloc dans le bloc d'organisation	20
	7.4	Enregistrer et compiler le projet	21
	7.5	Charger le programme	. 22
	7.6	Visualiser les blocs de programme	23
	7.7	Archivage du projet	25
8	Lis	ste de contrôle	26
9	Ex	xercice	27
	9.1	Énoncé du problème - exercice	27
	9.2	Planification	27
	9.3	Liste de contrôle - Exercice	. 28
1() Inf	formations complémentaires	. 29

LES TEMPORISATIONS ET COMPTEURS CEI MULTI-INSTANCES DANS SIMATIC S7-1500

1 Objectif

Ce chapitre présente l'utilisation des instances simples et multiples lors de la programmation de SIMATIC S7-1500 avec l'utilitaire TIA PORTAL.

Ce module explique les différents types de blocs de données d'instance et montre étape par étape comment ajouter des temporisations et compteurs CEI dans un bloc de programmation.

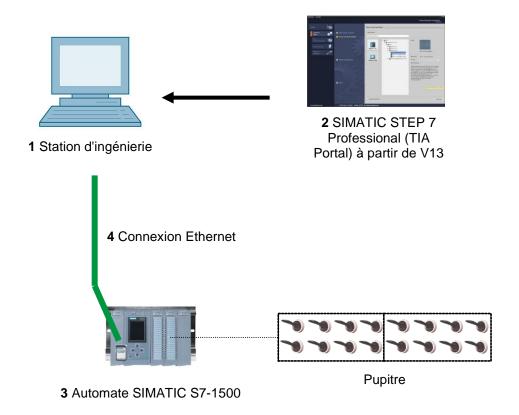
Les automates SIMATIC S7 énumérés au chapitre 3 peuvent être utilisés.

2 Conditions requises

Ce chapitre s'appuie sur la programmation de FB avec SIMATIC S7 CPU1516F-3 PN/DP. Pour ce chapitre, vous pouvez par ex. utiliser le projet suivant : 032-200_Programmation de FB_R1503.zap13

3 Configurations matérielles et logicielles requises

- 1 Station d'ingénierie : Le matériel et le système d'exploitation sont la condition de base (pour plus d'informations, voir le fichier Lisezmoi sur les DVD d'installation de TIA Portal)
- 2 Logiciel SIMATIC STEP 7 Professional dans TIA Portal à partir de V13
- 3 Automate SIMATIC S7-1500/S7-1200/S7-300, par exemple CPU 1516F-3 PN/DP à partir du firmware V1.6 avec carte mémoire et 16DI/16DO ainsi que 2AI/1AO Remarque : les entrées TOR doivent être mises en évidence sur un pupitre.
- 4 Connexion Ethernet entre la station d'ingénierie et l'automate



4 Théorie

4.1 Instances et multi-instances dans SIMATIC S7-1500

L'appel d'un bloc fonctionnel est appelé **instance**. A chaque appel d'un bloc fonctionnel est affectée une instance qui sert à enregistrer les données. On y enregistre les paramètres effectifs et les données statiques du bloc fonctionnel.

Les variables déclarées dans le bloc fonctionnel déterminent la structure du bloc de données d'instance.

Utilisation d'instances uniques et de multi-instances

Les instances peuvent être affectées comme suit :

Appel en tant qu'instance unique

- Un DB d'instance différent pour chaque instance d'un FB

Appel en tant que multi-instance

- Un seul DB d'instance pour plusieurs instances d'un ou plusieurs FB

4.1.1 Blocs de données d'instance/Instances uniques

L'appel d'un bloc fonctionnel auquel on attribue son propre bloc de données d'instance est appelé instance unique.

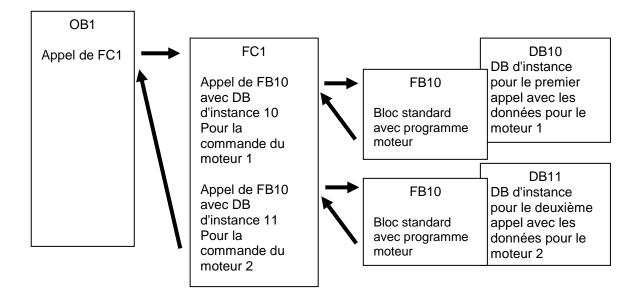
Si le bloc fonctionnel a été créé selon les règles des blocs standards réutilisables, il peut être appelé un nombre quelconque de fois.

Cependant, pour chaque appel en tant qu'instance unique, vous devez assigner un bloc de données d'instance différent à chaque fois.

Exemple d'instances uniques :

La figure ci-dessous montre la commande de deux moteurs par un bloc de fonction FB10 et deux blocs de données différents.

Les différentes données de chaque moteur (par exemple la vitesse, temps d'allumage, temps total de mise en marche) sont enregistrées dans les différents blocs de données d'instance DB10 et DB11.



Remarque : certaines commandes comme les temporisations et les compteurs se comportent comme des blocs de fonction. S'ils sont appelés, ils nécessitent également une zone de mémoire associée, sous la forme d'un DB d'instance, par exemple.

4.1.2 Multi-instances

La capacité mémoire de la CPU utilisée peut limiter le nombre de blocs de données alloué aux instances.

Si d'autres blocs fonctionnels existants comme les temporisations ou les compteurs sont appelés dans un bloc fonctionnel de votre programme utilisateur, il est possible d'appeler ces FB supplémentaires sans leur propre DB d'instance.

Il suffit pour cela de sélectionner dans Call option (options d'appel) « Multi-instances ».



Remarques : les multi-instances permettent d'enregistrer les données du bloc de fonction appelé dans le bloc de données d'instance du bloc de fonction appelant.

Dans ce cas, le bloc qui appelle doit toujours être un bloc de fonction.

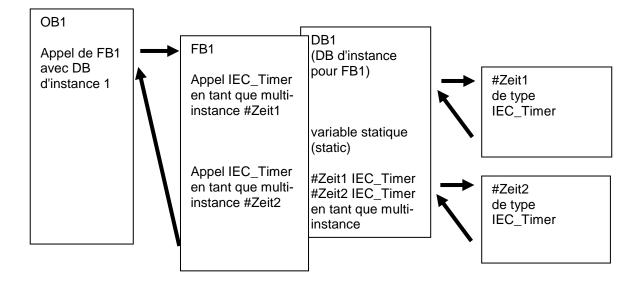
De cette façon, vous concentrez les données d'instance dans un seul bloc de données d'instance et vous pouvez utiliser le nombre de DB disponibles plus efficacement.

Ceci doit toujours être le cas si le bloc appelant doit être réutilisable comme un bloc standard.

Exemple de multi-instances :

La figure suivante montre deux appels dans un bloc fonctionnel d'une temporisation CEI de type TP (impulsion).

Les données qui sont différentes, des deux compteurs, sont stockées en tant que **multi-**instances dans le bloc de données d'instance DB1 du bloc fonctionnel FB1 appelant.



5 Énoncé du problème

Dans ce chapitre, on va compléter le bloc fonctionnel du chapitre "SCE_FR_032-200 Programmation de FB" en ajoutant une temporisateurs CEI.

6 Planification

La programmation de la temporisation CEI en tant qu'extension s'effectue dans le bloc fonctionnel MOTOR-AUTO [FB1] du projet « 032-200_Programmation de FB.zap13". Ce projet doit être désarchivé pour que la temporisation CEI TP (impulsion avec mémorisation) puisse être ajoutée. Une multi-instance a été créée pour servir de mémoire à la temporisation.

6.1 Mode automatique - Moteur du convoyeur avec fonction de temporisation

La fonction Memory_automatic_start_stop (mémoire_automatique_marche_arrêt) est activée avec mémorisation par la commande de démarrage, mais seulement si les conditions de réinitialisation ne sont pas remplies.

Memory_automatic_start_stop (mémoire_automatique_marche_arrêt) est mis à 0 si la commande d'arrêt est active, si le disjoncteur est actif ou si le mode automatique n'est pas activé (mode manuel).

La sortie moteur_automatique est activée si Memory_automatic_start_stop (mémoire_automatique_marche_arrêt) est mis à 1, les conditions de validation sont remplies et Memory_conveyor_start_stop (mémoire_convoyeur_marche_arrêt) mis à 1.

Par économie d'énergie, le convoyeur ne doit fonctionner que si une pièce se trouve dessus.

De ce fait, Memory_conveyor_start_stop (mémoire_convoyeur_marche_arrêt) est mis à 1 si capteur_toboggan_occupé signale une pièce et mis à 0 si Sensor_end_of_conveyor (capteur_fin_de_convoyeur) crée un front descendant, si le disjoncteur est actif ou si le mode automatique n'est pas activé (mode manuel).

Ajout d'une fonction de temporisation :

Comme sensor_end_of_conveyor (capteur_fin_de_convoyeur) n'est pas monté directement en bout de tapis, il est nécessaire de prévoir une extension du signal sensor_end_of_conveyor (capteur_fin_de_convoyeur).

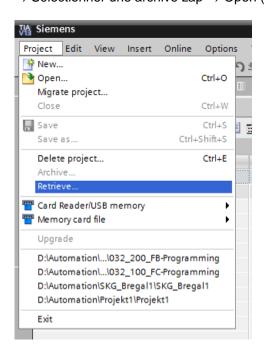
Pour cela, on va ajouter une impulsion avec mémorisation entre sensor_end_of_conveyor (capteur_fin_de_convoyeur) et la détection de front descendant.

7 Instructions structurées par étapes

Vous trouverez ci-après des instructions pour réaliser la planification. Si vous êtes déjà expérimenté, les étapes numérotées vous suffisent. Sinon, suivez les étapes détaillées des instructions.

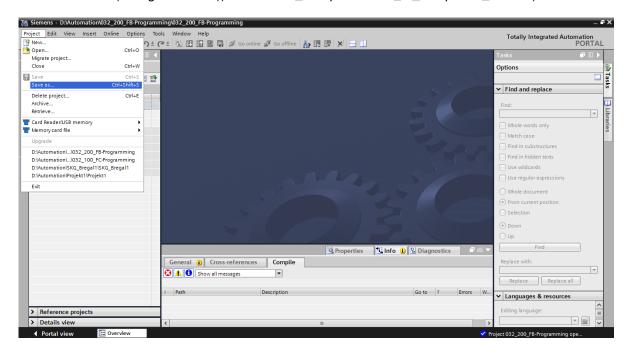
7.1 Désarchiver un projet existant

→ Avant de compléter le bloc fonctionnel "MOTOR_AUTO [FB1]", il faut désarchiver le projet « 032-200_Programmation de FB.zap13" du chapitre "SCE_FR_032-200 Programmation de FB". Pour désarchiver un projet existant, vous devez rechercher l'archive à partir de la vue de projet sous →Project (Projet)→Retrieve (Désarchiver).
 Confirmez votre choix avec "Open (Ouvrir)". (→ Project (Projet)→Retrieve (Désarchiver) → Sélectionner une archive zap → Open (Ouvrir))



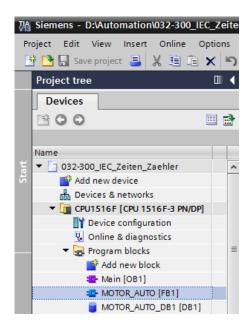
→ Sélectionner ensuite le répertoire cible pour enregistrer le projet désarchivé. Confirmez votre sélection par "OK". (→ Répertoire cible → OK) Enregistrer le projet sous 032-300_Temporisations et compteurs CEI (→ Project (Projet)

 \rightarrow Save as (Enregistrer sous)) \rightarrow 032-300_Temporisations_et_compteurs_CEI \rightarrow)

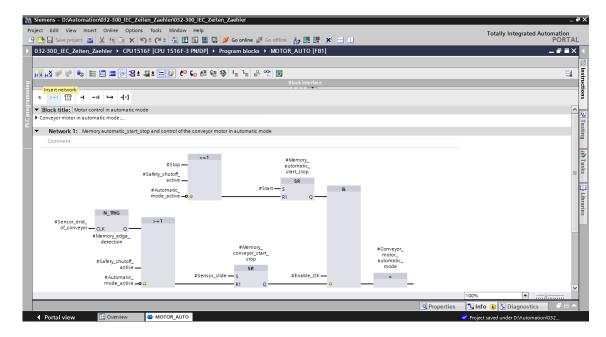


7.2 Ajout d'une temporisation CEI TP dans le bloc fonctionnel FB1 "MOTOR AUTO"

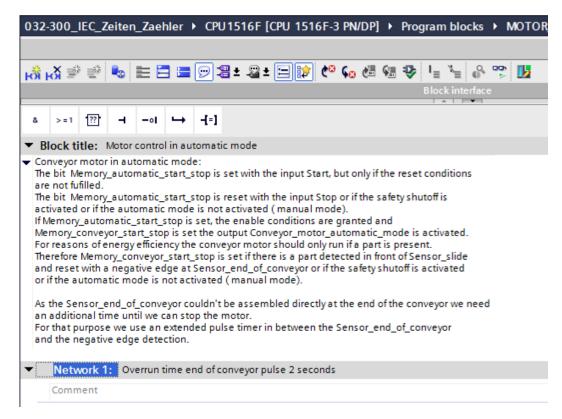
→ Ouvrir tout s'abord le bloc fonctionnel "MOTOR_AUTO [FB1]" par un double clic.



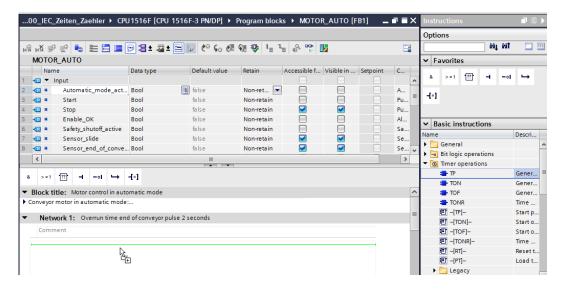
- → Au début du bloc fonctionnel "MOTOR_AUTO[FB1]", ajouter un réseau en sélectionnant
 - → "Block title (titre du bloc)" et en cliquant sur → III "insert network (insérer réseau)".



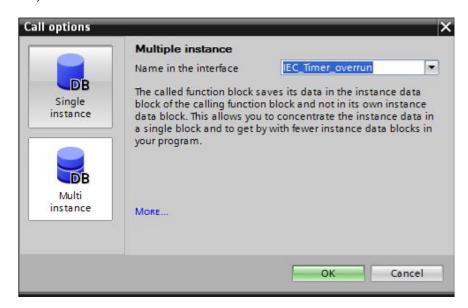
→ Renseigner le commentaire de bloc et le titre de réseau "Network 1 (réseau 1)".



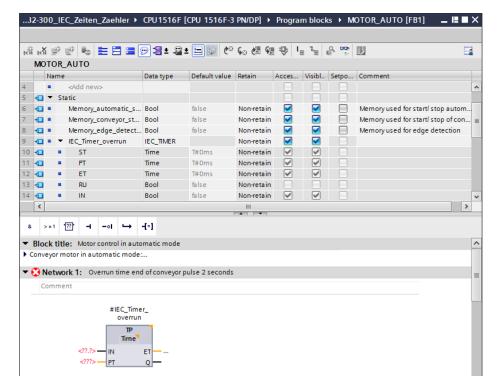
- → Dans la partie droite de la fenêtre de programmation se trouve la liste des instructions, dont les fonctions de temporisation. Sous → Basic instructions (instructions de base) → Timer operations (temporisations), rechercher la fonction TP (Generate pulsation (Générer impulsion)) et la faire glisser sur le réseau 1 (une ligne verte apparait, pointeur avec symbole +).
 - $(\rightarrow$ Instructions \rightarrow Basic instructions (Instructions de base) \rightarrow Timer operations (temporisations) \rightarrow TP)



→ La fonction temporisation a besoin d'une mémoire. Elle lui est fournie dans le bloc de données d'instance du bloc fonctionnel sans création d'un nouveau DB d'instance.
 Activez l'option → "Multiple instance (multi-instance)". Donner un nom à la multi-instance et valider avec → "OK". (→ Multiple instance (multi-instance) → IEC_Timer_overrun → OK)

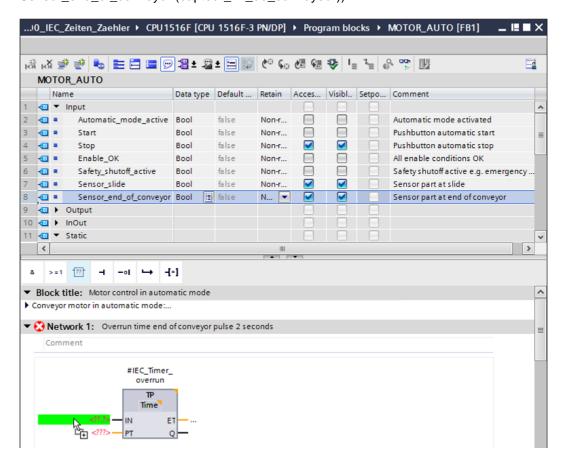


→ On crée ainsi dans la description de l'interface une structure de variables de type "static" adaptée à la temporisation TP.

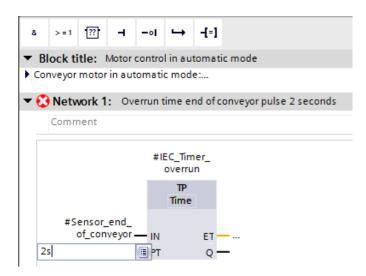


Remarque: une multi-instance ne peut être utilisée que pour la programmation dans un bloc de fonction car c'est le seul qui contient des variables statiques.

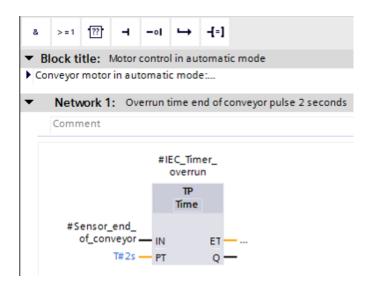
→ Faire glisser le paramètre d'entrée #Sensor_end_of_conveyor (capteur_fin_de_convoyeur) sur <??.?> devant le paramètre "IN" de la temporisation TP, afin qu'il soit activé en cas de front montant sur l'entrée #Sensor_end_of_conveyor (capteur_fin_de_convoyeur). Pour sélectionner plus facilement un paramètre dans la description de l'interface, le saisir sur le symbole bleu. (→ Sensor_end_of_conveyor (capteur_fin_de_convoyeur))



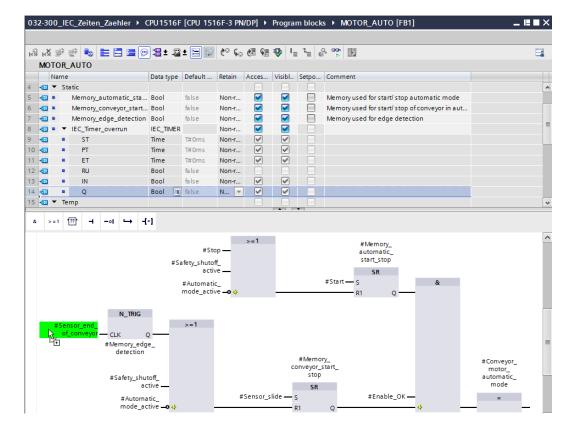
→ Devant le paramètre "PT", saisir la durée d'impulsion 2 secondes. (→ 2s)

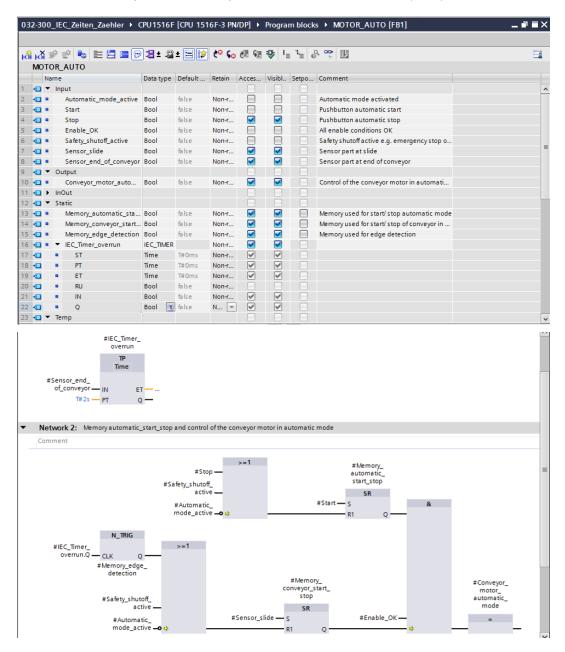


→ La saisie 2s est convertie automatiquement au format IEC-Time et affichée sous forme de constante "T#2s".



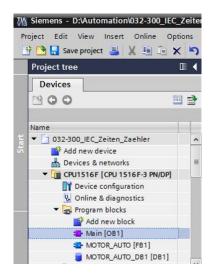
- → Dans la structure de variables "IEC_Timer_overrun" faire glisser la sortie "Q" sur l'entrée "CLK" du front descendant "N_TRIG" dans le réseau 2. La variable d'entrée préalablement saisie #Sensor_end_of_conveyor (capteur_fin_de_convoyeur) est remplacée et le convoyeur est stoppé par un front descendant de l'impulsion IEC_Timer_Overrun.
 - $(\rightarrow$ Réseau 2 \rightarrow IEC_Timer_Overrun \rightarrow Q \rightarrow #Sensor_end_of_conveyor (capteur_fin_de_convoyeur))



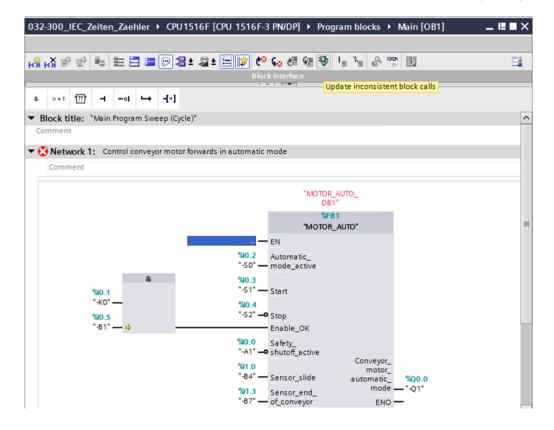


7.3 Mise à jour de l'appel de bloc dans le bloc d'organisation

→ Ouvrir le bloc d'organisation "Main [OB1]" par double clic.

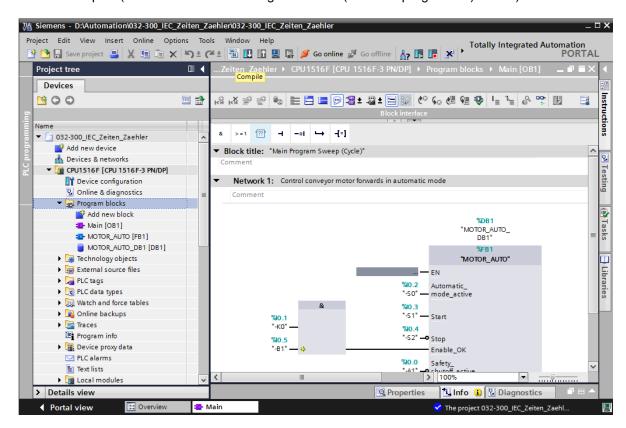


→ Dans le réseau 1 du bloc d'organisation "Main[OB1]", le bloc de données d'instance "MOTOR_AUTO_DB1" du bloc de fonction "MOTOR_AUTO [FB1]" semble incorrect car la mémoire supplémentaire pour la temporisation TP n'a pas encore été créée. Cliquer sur → "" Update inconsistent block calls (mettre à jour les appels de bloc incohérents). Le bloc de données d'instance "MOTOR_AUTO_DB1" est recréé correctement. (→ "")

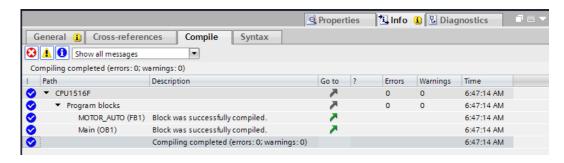


7.4 Enregistrer et compiler le projet

→ Pour enregistrer le projet, sélectionner " aver project dans le menu. Pour compiler tous les blocs, cliquer sur le dossier "Program blocks (Blocs de programme)" et dans le menu sur Compile. (→ Save project → Program blocks (Blocs de programme) → (Dans)

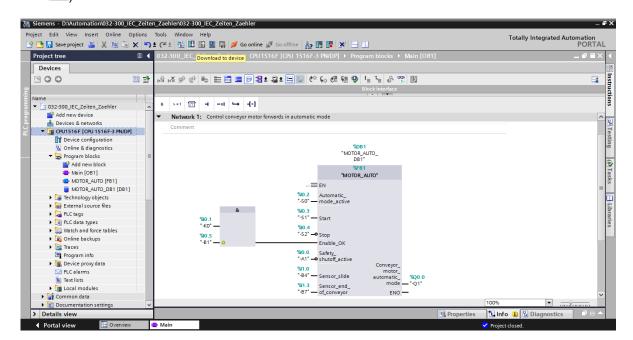


→ Dans la zone "Info" "Compile" les blocs compilés avec succès sont affichés.



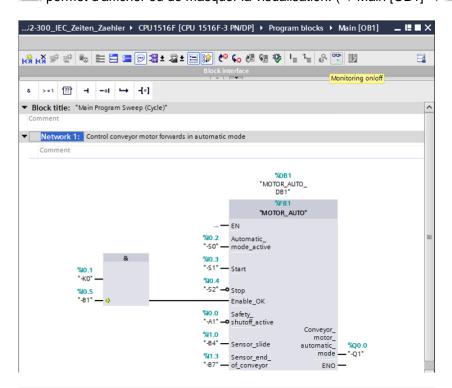
7.5 Charger le programme

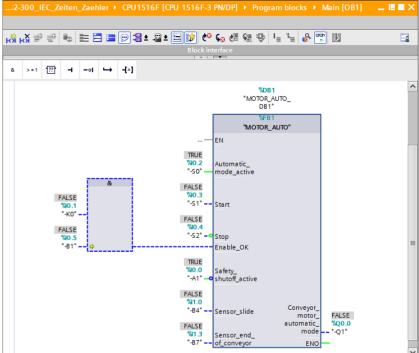
→ Une fois la compilation terminée avec succès, le programme créé y compris la configuration matérielle peut être chargé dans l'automate comme décrit auparavant. (→



7.6 Visualiser les blocs de programme

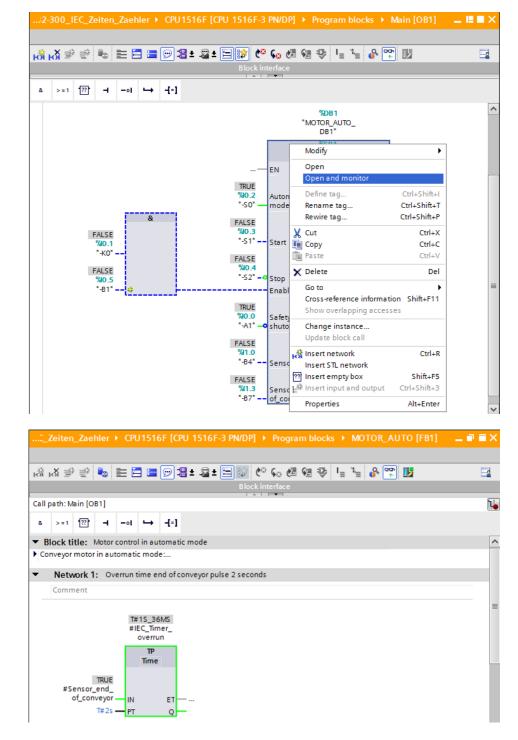
→ Pour visualiser le programme chargé, le bloc voulu doit être ouvert. Ensuite, un clic sur
 permet d'afficher ou de masquer la visualisation. (→ Main [OB1] → □)





Remarque : la visualisation s'effectue par signal et par automate. L'état des signaux sur la borne sont signalés par TRUE ou FALSE.

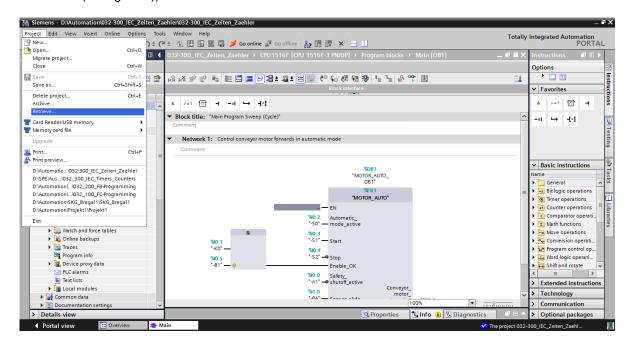
→ Le bloc d'organisation appelé "MOTOR_AUTO" [FB1] dans le bloc d'organisation "Main [OB1]" peut être ouvert et visualisé par clic droit ("Open and monitor") et le code du programme peut être visualisé dans le bloc de fonction avec la temporisation TP.
(→ "MOTOR_AUTO" [FB1] → Open and monitor (ouvrir et visualiser)



Remarque : la visualisation s'effectue par fonction et par automate. L'actionnement des capteurs et l'état de l'installation sont signalés par TRUE ou FALSE.

7.7 Archivage du projet

→ Pour finir, nous voulons archiver le projet complet. Sous la commande de menu → "Project (Projet)" sélectionner → "Archive...". Choisir le dossier d'archivage du projet et l'enregistrer au format "Archive de projet TIA Portal". (→ Project (Projet) "Archive" → Archive de→ projet TIA Portal → 032-300_Temporisations_et_compteurs_CEI.... → Save (Enregistrer))



8 Liste de contrôle

N°	Description	Vérifié
1	Compilation réussie et sans message d'erreur	
2	Chargement réussi et sans message d'erreur	
3	Mettre en marche l'installation (-K0 = 1) Vérin rentré / Réponse activée (-B1 = 1) Arrêt d'urgence (-A1 = 1) non activé Mode AUTOMATIQUE (-S0 = 1) Bouton Arrêt Automatique non actionné (-S2 = 1) Actionner brièvement le bouton Démarrage automatique (-S1 = 1) Capteur toboggan affecté activé (-B4 = 1) puis moteur du convoyeur avant vitesse fixe (-Q1 = 1) s'enclenche (-Q1 = 1) et reste en marche.	
4	Capteur convoyeur fin (-B7 = 1) \rightarrow -Q1 = 0 (après 2 secondes	
5	Actionner brièvement le bouton arrêt automatique (-S2 = 0) → -Q1 = 0	
6	Activer l'arrêt d'urgence (-A1 = 0) → -Q1 = 0	
7	Mode manuel (-S0 = 0) \rightarrow -Q1 = 0	
8	Éteindre l'installation (-K0 = 0) → -Q1 = 0	
9	Vérin non rentré (-B1 = 0) \rightarrow -Q1 = 0	
10	Le projet a été archivé avec succès	

9 Exercice

9.1 Énoncé du problème - exercice

Dans cet exercice, vous allez compléter le bloc fonctionnel MOTOR_AUTO [FB1] en lui ajoutant un compteur CEI. Le bloc fonctionnel ainsi complété doit être planifié, programmé et testé :

Le compartiment pour plastique ne peut contenir que 5 pièces, c'est pourquoi les pièces sont comptées à la fin du convoyeur.

Si le compartiment contient 5 pièces, le mode automatique doit être interrompu.

Une fois que le compartiment a été vidé, le mode automatique reprend après une commande Démarrer et le compteur est remis à zéro.

9.2 Planification

Planifiez seul la réalisation de l'énoncé.

Remarque: voir l'aide en ligne pour l'utilisation des compteurs CEI dans SIMATIC S7-1500.

9.3 Liste de contrôle - Exercice

Nº	Description	Vérifié
1	Compilation réussie et sans message d'erreur	
2	Chargement réussi et sans message d'erreur	
3	Mettre en marche l'installation (-K0 = 1) Vérin rentré / Réponse activée (-B1 = 1) Arrêt d'urgence (-A1 = 1) non activé Mode AUTOMATIQUE (-S0 = 1) Bouton Arrêt Automatique non actionné (-S2 = 1) Actionner brièvement le bouton Démarrage automatique (-S1 = 1) Capteur toboggan affecté activé (-B4 = 1) puis moteur du convoyeur avant vitesse fixe (-Q1 = 1) s'enclenche et reste en marche.	
4	Capteur convoyeur fin (-B7 = 1) \rightarrow -Q1 = 0 (après 2 secondes	
5	Actionner brièvement le bouton arrêt automatique (-S2 = 0) → -Q1 = 0	
6	Activer l'arrêt d'urgence $(-A1 = 0) \rightarrow -Q1 = 0$	
7	Mode manuel (-S0 = 0) \rightarrow -Q1 = 0	
8	Éteindre l'installation (-K0 = 0) → -Q1 = 0	
9	Vérin non rentré (-B1 = 0) \rightarrow -Q1 = 0	
10	5e pièce dans le compartiment → -Q1 = 0	
11	Le projet a été archivé avec succès	

10 Informations complémentaires

Des informations complémentaires vous sont proposées afin de vous aider à vous exercer ou à titre d'approfondissement, par ex. : mises en route, vidéos, didacticiels, applis, manuels, guides de programmation et logiciel/firmware d'évaluation sous le lien suivant :

www.siemens.com/sce/s7-1500