

Support d'apprentissage/ de formation

Siemens Automation Cooperates with Education (SCE) | A partir de la version V14 SP1

Module 031-300 TIA Portal Temporisations CEI et compteurs CEI Multi-instances pour le SIMATIC S7-1200

siemens.com/sce



Utilisation libre pour les instituts publics de formation et de R&D. © Siemens AG 2018. Tous droits réservés.

Packages SCE pour formateurs adaptés à ces supports d'apprentissage/de formation

- SIMATIC S7-1200 AC/DC/RELAIS (paquet de 6) "TIA Portal" N° d'article : 6ES7214-1BE30-4AB3
- SIMATIC S7-1200 DC/DC/DC (paquet de 6) "TIA Portal" N° d'article : 6ES7214-1AE30-4AB3
- SIMATIC STEP 7 BASIC V14 SP1 Upgrade (for S7-1200) (paquet de 6) "TIA Portal" N° d'article : 6ES7822-0AA04-4YE5

Veuillez noter que les packages pour formateurs ont parfois été remplacés par de nouveaux packages. Vous pouvez consulter les packages SCE actuellement disponibles sous : <u>siemens.com/sce/tp</u>

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

Le support d'apprentissage/de formation SCE pour une solution d'automatisation cohérente Totally Integrated Automation (TIA) ont été créés spécialement 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 SA n'assume aucune responsabilité quant au contenu.

Cette documentation ne peut être utilisée que pour une première formation aux produits/systèmes Siemens. Ce qui veut dire qu'elle peut être copiée, en partie ou dans son 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 copie de cette documentation, son exploitation et la communication de son contenu sont autorisés dans le cadre d'instituts publics de formation et de formation continue.

Toute exception requiert au préalable l'autorisation écrite de la part des interlocuteurs de Siemens SA : 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 la TU de Dresde, notamment le professeur Leon Urbas et l'entreprise Michael Dziallas Engineering ainsi que toutes les personnes ayant contribué à la réalisation de ce support d'apprentissage/de formation.

Sommaire

1	Obje	ectif	4
2	Con	dition	4
3	Con	figurations matérielles et logicielles requises	5
4	Part	ie théorique	6
	4.1	Notions d'instances et de multi-instances pour le SIMATIC S7-1200	6
	4.1.	Blocs de données d'instance / instances uniques	7
	4.1.	2 Multi-instances	8
5	Арр	lication à réaliser	10
6	Plar	ification	10
	6.1	Mode automatique – Moteur du convoyeur avec fonction de temporisation	10
	6.2	Schéma technologique	11
	6.3	Tableau d'affectations	12
7	Mar	che à suivre détaillée	13
	7.1	Désarchiver un projet existant	13
	7.2	Compléter le bloc fonctionnel FB1 "MOTOR_AUTO" par un temporisateur CEI type TP	15
	7.3	Actualiser l'appel de bloc dans le bloc d'organisation	22
	71		
	1.4	Enregistrer et compiler le programme	23
	7.5	Enregistrer et compiler le programme Charger le programme	23 24
	7.4 7.5 7.6	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme	23 24 25
	7.5 7.6 7.7	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme Archiver le projet	23 24 25 27
	7.4 7.5 7.6 7.7 7.8	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme Archiver le projet Check-list	23 24 25 27 28
8	7.4 7.5 7.6 7.7 7.8 Exe	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme Archiver le projet Check-list	23 24 25 27 28 29
8	7.4 7.5 7.6 7.7 7.8 Exe 8.1	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme Archiver le projet Check-list rcice Énoncé du problème – Exercice	23 24 25 27 28 29 29
8	7.5 7.6 7.7 7.8 Exe 8.1 8.2	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme Archiver le projet Check-list rcice Énoncé du problème – Exercice Schéma technologique	23 24 25 27 28 29 29 29
8	7.5 7.6 7.7 7.8 Exe 8.1 8.2 8.3	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme Archiver le projet Check-list rcice Énoncé du problème – Exercice Schéma technologique Tableau d'affectations	23 24 25 27 28 29 29 29 29 29 29
8	7.4 7.5 7.6 7.7 7.8 Exe 8.1 8.2 8.3 8.4	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme Archiver le projet Check-list rcice Énoncé du problème – Exercice Schéma technologique Tableau d'affectations Planification	23 24 25 27 28 29 29 29 29 29 29 23
8	7.4 7.5 7.6 7.7 7.8 Exe 8.1 8.2 8.3 8.4 8.5	Enregistrer et compiler le programme Charger le programme Visualiser des blocs de programme Archiver le projet Check-list rcice Énoncé du problème – Exercice Schéma technologique Tableau d'affectations Planification Check-list – Exercice	23 24 25 27 28 29 29 29 29 30 31

Temporisations CEI et compteurs CEI / Multi-instances pour le SIMATIC S7-1200

1 Objectif

Dans le présent chapitre, vous allez apprendre l'utilisation d'instances uniques et de multiinstances pour programmer le SIMATIC S7-1200, grâce au logiciel de programmation TIA Portal.

Ce module décrit les différents types de blocs de données d'instance et montre les différentes étapes à suivre pour étendre un bloc de programme avec des temporisations et des compteurs CEI.

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

2 Condition

Ce chapitre s'appuie sur la programmation de blocs fonctionnels pour CPU1214C SIMATIC S7. Pour l'étude de ce chapitre, vous pouvez par ex. recourir au projet suivant :

031-200_Programmation de FB_S7-1200....zap14

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 Basic dans TIA Portal à partir de V14
- 3 Automate SIMATIC S7-1200, par exemple CPU 1214C DC/DC/DC avec Signal Board ANALOG OUTPUT SB1232, 1 AO à partir du firmware V4.2.1

Remarque : les entrées TOR doivent être mises en évidence sur un pupitre.

4 Connexion Ethernet entre la station d'ingénierie et l'automate



4 Partie théorique

4.1 Notions d'instances et de multi-instances pour le SIMATIC S7-1200

L'appel d'un bloc de fonction est appelé **instance**. A chaque appel d'un bloc fonctionnel est affecté une **instance** qui sert de mémoire de données. On y enregistre les paramètres actuels 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 sous forme de 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 de fonction 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 et des règles de bibliothèque, il peut être appelé un nombre quelconque de fois.

Cependant, vous devez affecter un bloc de données d'instance différent à chaque appel sous forme d'instance unique.

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 de démarrage, temps total de fonctionnement) 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

Il est possible que vous ne vouliez ou que vous ne puissiez allouer qu'un nombre limité de blocs de données pour des instances et cela à cause de la capacité mémoire de la CPU utilisée.

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

Il suffit pour cela de sélectionner dans les options d'appel 'Multi-instance'.

Call options	×
Single instance	Multiple instance Name in the interface IEC_Timer_overrun The called function block saves its data in the instance data block of the calling function block and not in its own instance data block. This allows you to concentrate the instance data in a single block and to get by with fewer instance data blocks in your program. MORE
	OK Cancel

Remarques : les multi-instances permettent d'enregistrer les données du bloc 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, vous pouvez ainsi 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 **multiinstances** dans le bloc de données d'instance DB1 du bloc fonctionnel FB1 appelant.



5 Application à réaliser

Dans le présent chapitre, le bloc fonctionnel du chapitre "SCE_FR_031-200 Programmation de FB S7-1200" va être complété par une temporisation CEI.

6 Planification

Le temporisation CEI est programmée comme une extension du bloc fonctionnel MOTOR-AUTO [FB1] du projet "031-200_Programmation de FB_S7-1200.zap14". Ce projet doit être désarchivé afin d'insérer le temporisation CEI de type TP (impulsion à mémoire). Une multi-instance est créée comme mémoire pour la temporisation.

6.1 Mode automatique – Moteur du convoyeur avec fonction de temporisation

Mémoire_Automatique_Démarrage_Arrêt est déclenché et mémorisé par la Commande_ Démarrage, mais à la seule condition qu'aucun signal de réinitialisation ne soit actif.

Mémoire_Automatique_Démarrage_Arrêt est réinitialisé lorsque le signal Commande_Arrêt est actif, le circuit de protection est activé ou que le mode automatique n'est pas activé (mode manuel).

La sortie Automatique_Moteur est commandée lorsque le signal Mémoire_Automatique_ Démarrage_Arrêt est mis à 1, les conditions de validation sont remplies et le signal Mémoire_Convoyeur_Démarrage_Arrêt est mis à 1.

Pour des raisons d'économie d'énergie, le convoyeur ne doit marcher que lorsqu'une pièce est présente.

C'est pourquoi Mémoire_Convoyeur_Démarrage_Arrêt est mis à 1 lorsque Capteur_Glissière_ occupée signale la présence d'une pièce et réinitialisé lorsque Capteur_fin de convoyeur génère un front descendant ou le circuit de protection est activé ou que le mode automatique n'est pas activé (mode manuel).

Extension par une fonction de temporisation :

Comme le Capteur_fin de convoyeur ne pouvait être directement installé sur l'extrémité du convoyeur, une prolongation du signal Capteur_fin de convoyeur est requise.

Pour cela, une impulsion mémorisée est insérée entre le signal Capteur_fin de convoyeur et la détection de front descendant.

6.2 Schéma technologique

La figure ci-dessous montre le schéma technologique pour l'application à réaliser.



Figure 1 : Schéma technologique

Schalter der Sortieranlage	Automatikbetrieb		Handbetrieb / Manual mode
Switches of sorting station	Automatic mode		-S3 Tippbetrieb -M1 vorwärts/
-P1 ein/on	-P5 gestartet/started		Manual -M1 forwards
-Q0 Hauptschalter/Main switch	-S1 Start/start		-S4 Tippbetrieb -M1 rückwärts/
-P4 aktiviert/active			Manual -M1 backwards
-A1 NOTHALT/Emergency stop	-S2 Stopp/stop		-P7 ausgefahren/extended
.P2 Hand/manual .P3 Auto/auto			-S6 Zylinder -M4 austahren/
-S0 Betriebcart/operating mode			-P6 eingefahren/retracted
			cylinder -M4 retract
		I I	

Figure 2 : Pupitre de commande

6.3 Tableau d'affectations

DE	Туре	Code	Fonction	NC/NO
E 0.0	BOOL	-A1	Message ARRET D'URGENCE ok :	NC
E 0.1	BOOL	-K0	Installation "Marche"	NO
E 0.2	BOOL	-S0	Commutateur mode Manuel (0)/ Automatique (1)	Manuel = 0 Auto=1
E 0.3	BOOL	-S1	Bouton démarrage automatique	NO
E 0.4	BOOL	-S2	Bouton arrêt automatique	NC
E 0.5	BOOL	-B1	Capteur tige du vérin -M4 rentrée	NO
E 1.0	BOOL	-B4	Capteur glissière occupée	NO
E 1.3	BOOL	-B7	Capteur pièce à la fin du convoyeur	NO

Cette application requiert les signaux suivants comme opérande global.

DA	Туре	Code	Fonction	
A 0.0	BOOL	-Q1	Moteur du convoyeur M1 avance à vitesse fixe	

Legende zur Belegungsliste

DE	Entrée TOR	DA	Sortie TOR
AE	Entrée analogique	AA	Sortie analogique

E Entrée A Sortie

NC Normally Closed (contact à ouverture)

NO Normally Open (contact à fermeture)

7 Marche à suivre détaillée

Vous trouverez ci-après une description étape par étape de la marche à suivre pour la planification. Si vous vous en sortez déjà bien, vous pouvez vous contenter des numéros correspondant aux étapes pour réaliser l'application. Sinon, il vous suffit de suivre la procédure détaillée décrite ci-dessous.

7.1 Désarchiver un projet existant

® Avant de compléter le bloc fonctionnel "MOTOR_AUTO [FB1]", nous devons désarchiver le projet "031-200_Programmation de FB_S7 1200.zap14" du chapitre "SCE_FR_031-200 Programmation de FB S7-1200". Pour désarchiver un projet existant, vous devez sélectionner l'archive correspondant sous ® Projet ® Désarchiver dans la vue du projet. Confirmez votre sélection avec Ouvrir. (® Projet ® Désarchiver ® Sélection d'une archive .zap ® 031-200 Programmation de FB S7-1200.zap14 ® Ouvrir)

Project I	Edit	View	Insert	Online	Options	Тс
📑 New) ±
👌 Open					Ctrl+O	4
Migrate	proj	ect				
Close					Ctrl+W	
Save					Ctrl+S	
Save as	5			Ctrl	+Shift+S	
Delete	proje	ct			Ctrl+E	1
Archive						1
Retriev	e					
Manag	e mul	tiuser s	erver pro	jects		
TCard Re	ader	USB me	emory		•	
T Memor	ycard	l file			•	
Start ba	asic ii	ntegrity	check			
Upgrad	e					

Ensuite, vous pouvez s
 électionner le r
 épertoire cible dans lequel vous souhaitez enregistrer
 le projet d
 ésarchivé. Confirmez votre s
 élection avec "OK".

Le projet ouvert est enregistré sous 031-300_Temporisations_CEI _compteurs (
 Project (Projet)
 Save as (Enregistrer sous))
 031-300_Temporisations_CEI _compteurs
 Save (Enregistrer)



7.2 Compléter le bloc fonctionnel FB1 "MOTOR_AUTO" par un temporisateur CEI type TP



® Ouvrez d'abord le bloc fonctionnel "MOTOR_AUTO [FB1]" en double-cliquant dessus.

Insérez un nouveau réseau au début du bloc fonctionnel "MOTOR_AUTO [FB1]" en sélectionnant d'abord le
 "Titre de bloc", puis cliquez sur l'icône
 Motor licône insérer réseau.



Réseau 1:" avec toutes
 informations utiles.

03	1-300_	IEC_1	imer	s_Co	unter	s 🕨 C	PU_1214C [CPU 1214C DC/DC/DC] → Program blocks → MOTOR_AUTO [FB1]	_ # =×
I	<mark>⊮X</mark> ∰	ř 🖹	ii,			1	월±월±월ː글(*) @ 60 년 년 왕 두 날 놀 61 년 & ?? 🔒	
							Block interface	
8	>=1	7?	-	-01	-	-[=]		
•	Block ti	itle:	Motor	contr	olinau	utomat	ic mode	~
TI a If F TI a O A a F	he bit M re not fune bit M ctivated Memory lemory_cor reason or reason enerefore nd reset r if the a s the Se n additio or that p	lemory filled. or if the auto convey ms of e with a utoma ns or_e onal time ourpos	_auto _auto matic_ vor_sta nergy ory_col a nega tic mo end_of me un e we u	matic, omatic, start_start_storefficien nveyo tive e ode is 	_start_ _start_ c mod _stop is op is se ency th r_start dge at not ac reyor c can st extension	stop is stop is e is not s set, the e conv _stop i : Senso tivated ouldn't op the ded pu	set with the input Start, but only if the reset conditions reset with the input Start, but only if the reset conditions activated (manual mode). te enable conditions are granted and utput Conveyor_motor_automatic_mode is activated. eyor motor should only run if a part is present. is set if there is a part detected in font of Sensor_Bide r_end_of_conveyor or if the safety shutoff is activated (manual mode). be assembled directly at the end of the conveyor we need motor. se timer in between the Sensor_end_of_conveyor	
•	Netv	vork 1	: O	/errun	time e	end of	conveyor pulse 2 seconds	
Ĩ	Comm	nent						

® Les fonctions de temporisation se trouvent dans votre fenêtre de programmation, à droite, dans la liste des instructions. Recherchez sous
 Instructions de base
 Temporisation, la fonction
 TP (Créer impulsion) et faites-la glisser dans votre Réseau 1 (la ligne verte apparaît, pointeur de la souris avec le +).

(@	Instructions ®	Instructions de base	® Terr	porisations ®	💶 TP)	
•						

									Options	
i kă 🖈 🖆 🐛 🔚 🚍 🔚	🖻 📲 ± 📲 ± 📲	± 🖃 😥 🎨 🤇	. 🖑 🐄 🕹	⊊ I = ³ =	SI CI	0 00 b	e E		- 164 ·	
MOTOR_AUTO									> Favorites	
Name	Data type	Default value	Retain	Accessible f	Writa	Visible in	Setp		Resis instructions	
🕣 🔻 Input								~	Name	Deccr
- Automatic_mode_act	Bool	alse false	Non-ret							Descr
- Start	Bool	false	Non-retain					=		
📲 Stop	Bool	false	Non-retain						Imer operations	
- Enable_OK	Bool	false	Non-retain							Gener
- Safety_shutoff_active	Bool	false	Non-retain						TON	Gener
📹 🔹 Sensor_slide	Bool	false	Non-retain						TOF	Gener
Sensor_end_of_conve	Bool	false	Non-retain							Time
🕣 🔻 Output										Start
Conveyor_motor_aut	Bool	false	Non-retain					~	IT -ITONI-	Start
<		111					>		I -[TOF]-	Start
								-	I -[TONR]-	Time
>=1 ??? -I -oI →	-[=]								I -[RT]-	Reset
Block title: Motor control in auto	matic mode								FT -[PT]-	Load .
Conveyor motor in automatic mode	initiatic mode							÷	Counter operations	
								_	Comparator operations	
Network 1: Overrun time en	d of conveyor pulse	2 seconds						-	Math functions	
Comment								-	Move operations	
Comment									Conversion operations	
									Program control operati	
									Word logic operations	
									Shift and rotate	

La fonction temporisation requiert une mémoire. Elle lui est fournie dans le bloc de données d'instance du bloc de fonction sans création d'un nouveau DB d'instance. Activez l'option
® "Multi-instance". Attribuez un nom à la multi-instance et confirmez la saisie avec ® "OK".
(® Multi-instance ® IEC_Timer_Attente ® OK)

Call options	×
Single instance	Multiple instance Name in the interface If you call the function block as a multi-instance, it saves its data in the instance data block of the calling function block and not in its own instance data block. This allows you to concentrate the instance data in a single block and to get by with fewer instance data blocks in your program.
Parameter instance	more OK Cancel

 R Cela crée une variable structurée de type "Static" correspondant à la temporisation TP dans la description de l'interface.

MOTOR AUTO								
Name	Data type	Default value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment
🕣 🔻 Static								
Memory_automatic_start_stop	Bool	false	Non-retain					Memory used for start/ stop automatic
Memory_conveyor_start_stop	Bool	false	Non-retain					Memory used for start/ stop of convey
- Memory_edge_detection	Bool	false	Non-retain					Memory used for edge detection
🕣 🔹 🔻 IEC_Timer_overrun	TP_TIME		Non-retain					
🚭 🔹 PT	Time	T#Oms	Non-retain		 Image: A start of the start of	Image: A start of the start		
🕣 🔹 ET	Time	T#Oms	Non-retain			V		
🕣 = IN	Bool	false	Non-retain		 Image: A start of the start of	V		
🕣 = Q	Bool	false	Non-retain			Image: A start and a start		
🕣 🔻 Temp								
<								
>=1 [??] → -ol → -[=] Block title: Motor control in automatic n	node							
Network 1: Overrun time end of con	veyor pulse :	2 seconds						
#IEC_Timer_ overrun								

Remarque : une multi-instance ne peut être utilisée que pour la programmation dans un bloc fonctionnel, car ce n'est que là qu'il existe des variables de type « Static ».

® Maintenant, faites glisser le paramètre d'entrée #Capteur_fin de convoyeur et déposez-le sur <??.?> en regard du paramètre "IN" de la temporisation TP pour que celle-ci soit déclenchée en cas d'un front montant. La meilleure méthode pour sélectionner un paramètre dans la description de l'interface consiste à le prendre par l'icône bleue

MOTOR_AUTO								
Name	Data type	Default value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment
a v Input								
Automatic_mode_active	Bool	false	Non-retain					Automatic mode activated
🖬 = Start	Bool	false	Non-retain					Pushbutton automatic start
🖬 = Stop	Bool	false	Non-retain					Pushbutton automatic stop
Enable_OK	Bool	false	Non-retain					All enable conditions OK
Safety_shutoff_active	Bool	false	Non-retain					Safety shutoff active e.g. emergency
Sensor_slide	Bool	false	Non-retain					Sensor part at slide
Sensor_end_of_conveyor	Bool 🔳	false	Non-ret 💌					Sensor part at end of conveyor
🛛 👻 Output								
Conveyor_motor_automatic	Bool	false	Non-retain					Control of the conveyor motor in aut
						and the second se		
			1.*					
>=1 ⑦ol → lock title: Motor control in automatic r nveyor motor in automatic mode: Network 1: Overrun time end of con Comment	mode Iveyor pulse 2	seconds		m 				

(
 Capteur_fin de convoyeur)

 ® Saisissez la durée d'impulsion souhaitée en regard du paramètre "PT", c'est-à-dire 2 s. (® 2s)





Puis, faites glisser la sortie "Q" de la variable structurée "IEC_Timer_Attente" sur l'entrée "CLK" du front descendant "N_TRIG" dans Réseau 2. La variable sur l'entrée #Capteur_fin de convoyeur est alors remplacée et le convoyeur, arrêté par un front descendant de l'impulsion IEC_Timer_Attente. (® Réseau 2 ® IEC_Timer_Attente ® Q ® #Capteur_fin de convoyeur)



N'oubliez pas de cliquer sur Save project à la fin de chaque étape. Ci-après, le bloc de fonction "MOTOR_AUTO [FB1] terminé, incluant le temporisateur représenté dans le logigramme (LOG).

Name Data type Default value Retain Accessible f. Writz Visible in Septint Comment 4 • Automatic_mode_active Bool false Non-retain • • Automatic_mode active de	MOTOR	AUTO								
1 Imput Im	Name		Data type	Default value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment
2 4 Automatic_mode_active Bool false Non-retain Automatic mode activated 3 5 Start Bool false Non-retain Pushbutton automatic start 4 4 5 Start Bool false Non-retain Pushbutton automatic start 4 4 5 Stap Bool false Non-retain Pushbutton automatic start 6 4 Staft Bool false Non-retain Pushbutton automatic start 6 4 Safety_shutoff_active Bool false Non-retain Pushbutton automatic start 7 5 Sensor_galde Bool false Non-retain Pushbutton automatic start 8 4 Sensor_galde_conveyor Bool false Non-retain Pushbutton automatic start 10 4 Conveyor_motor_automatic	1 🕣 🕶 Ing	put								^
3 Q Start Bool false Non-retain Pushbutton automatic start 4 Q Stop Bool false Non-retain Q Pushbutton automatic start 5 Q Enable_OK Bool false Non-retain Q All enable conditions OK 6 Q Safety_shutoff_active Bool false Non-retain Q All enable conditions OK 7 Q Sensor_slide Bool false Non-retain Q Sensor part at slide 8 Q Sensor_end_of_conveyor Bool false Non-retain Q Sensor part at end of conveyor 9 Q Output Conveyor_motor_automatic Bool false Non-retain Q Sensor part at end of conveyor 9 Q Indux Bool false Non-retain Q Sensor part at end of conveyor 9 Q Indux Bool false Non-retain Q Sensor part at end of conveyor 10 Q Memory_automatic_start_stop Bool false Non-retain	2 🕣 =	Automatic_mode_active	Bool	false	Non-retain					Automatic mode activated
4 Q Stop Bool false Non-retain Q Pushbutton automatic stop 5 Q Enable_OK Bool false Non-retain Q All enable conditions OK 6 Q Safety.shutoff_active Bool false Non-retain Q Safety.shutoff active e.g. emergency: 7 Q Sensor_end_of_conveyor Bool false Non-retain Q Q Sensor part at end of conveyor 9 Q Output Sensor_end_of_conveyor Bool false Non-retain Q Q Sensor part at end of conveyor 9 Q Output Sensor_automatic_star_star Bool false Non-retain Q Q Control of the conveyor motor in auto 10 Q Conveyor_motor_automatic_star_stop Bool false Non-retain Q Q Control of the conveyor motor in auto 11 Q Non-retain Q Q Memory_used for start/ stop automatic Memory used for start/ stop automatic 12 V Memory_edge_detection Bool false Non-retain Q<	3 🕣 🗉	Start	Bool	false	Non-retain					Pushbutton automatic start
5 G1 Enable_OK Bool false Non-retain Image: Control of Conveyor Safety shutoff active e.g. emergency: 7 G1 Sensor_slide Bool false Non-retain Image: Conveyor Safety shutoff active e.g. emergency: 9 G1 Output Sensor_end_of_conveyor Bool false Non-retain Image: Conveyor_motor_automatic Bool false Non-retain Image: Conveyor_motor_automatic Bool false Non-retain Image: Conveyor_motor_automatic Bool false Non-retain Image: Conveyor_motor automatic Memory used for start/ stop automatic <td>4 🕣 🕷</td> <td>Stop</td> <td>Bool</td> <td>false</td> <td>Non-retain</td> <td></td> <td></td> <td></td> <td></td> <td>Pushbutton automatic stop</td>	4 🕣 🕷	Stop	Bool	false	Non-retain					Pushbutton automatic stop
6 Qi = Safety_shutoff_active Bool false Non-retain Qi = Safety_shutoff_active e.g. emergency: 7 Qi = Sensor_slide Bool false Non-retain Qi = Sensor part at slide 8 Qi = Sensor_end_of_conveyor Bool false Non-retain Qi = Qi = Sensor part at end of conveyor 9 Qi = Conveyor_motor_automatic Bool false Non-retain Qi = Qi = Conveyor motor in automatic 10 Qi = Conveyor_motor_automatic Bool false Non-retain Qi = Qi = Conveyor motor in automatic in automatic 11 Qi = Static Bool false Non-retain Qi = Qi = Conveyor_motor_automatic_start_stop Bool false Non-retain Qi =	5 🕣 =	Enable_OK	Bool	false	Non-retain					All enable conditions OK
7 C1 Sensor_slide Bool false Non-retain Image: Sensor_send_of_conveyor Bool false Non-retain Image: Sensor_send_of_conveyor Sensor part at slide 9 C1 Output Image: Sensor_motor_automatic Bool false Non-retain Image: Sensor_motor_automatic Bool false Non-retain Image: Sensor_automatic Bool false Non-retain Image: Sensor_automatic Bool false Non-retain Image: Sensor_automatic Control of the conveyor motor in automatic 11 C1 Image: Sensor_automatic_star_stop Bool false Non-retain Image: Sensor_automatic_star_stop Bool false Non-retain Image: Sensor_automatic Memory used for start/ stop automatic 12 Image: Sensor_cnucyor_start_stop Bool false Non-retain Image: Sensor_cnucyor_start/ stop automatic Image: Sensor_automatic_start/ stop automatic 13 C1 Memory_cnucyor_start_stop Bool false Non-retain Image: Sensor_automatic Memory used for start/ stop automatic 16 C1 Image: Sensor_automatic Memory used for start/ stop automa	6 🕣 =	Safety_shutoff_active	Bool	false	Non-retain					Safety shutoff active e.g. emergency st
8 C1 Sensor_end_of_conveyor Bool false Non-retain Image: Sensor part at end of conveyor 9 C1 Output Bool false Non-retain Image: Sensor part at end of conveyor 10 C2 Conveyor_motor_automatic Bool false Non-retain Image: Sensor part at end of conveyor 11 C2 Conveyor_motor_automatic_surt_automatic Bool false Non-retain Image: Sensor part at end of conveyor 12 Image: Sensor part at end of conveyor Bool false Non-retain Image: Sensor part at end of conveyor 13 C2 Static Image: Sensor part at end of conveyor Sensor part at end of conveyor 13 C2 Static Image: Sensor part at end of conveyor Sensor part at end of conveyor 14 C2 Memory_endpart Bool false Non-retain Image: Sensor part at end of conveyor 15 C2 Memory_endpart Bool false Non-retain Image: Sensor part at end of conveyor 17 C2 FT Time TPOms Non-retain Image: Sensor part at end of conveyor Memory u	7 📶 🖷	Sensor_slide	Bool	false	Non-retain					Sensor part at slide
9 € V Output 10 € Conveyor_motor_automatic Bool false Non-retain ♥ ♥ C Control of the conveyor motor in automatic 11 € V InOut 12 € Add new> 13 € Static 14 € Memory_satomatic_start_stop Bool false Non-retain ♥ ♥ Ø Memory used for start's top automati 15 € Memory_conveyor_start_stop Bool false Non-retain ♥ Ø Ø Memory used for start's top of convey 16 € I € Memory_conveyor_start_stop Bool false Non-retain ♥ Ø Ø Memory used for start's top of convey 17 € I EC_Time_overrun IP_IIME Non-retain ♥ Ø Ø Memory used for edge detection 18 € I € PT Time T#Oms Non-retain Ø Ø Ø Ø 19 € I EC_Time_overrun IP_IIME Non-retain Ø Ø Ø Ø 10 0 I € ET Time T#Oms Non-retain Ø Ø Ø Ø 10 0 I € I IN Bool false Non-retain Ø Ø Ø Ø 10 0 I € I IN Bool false Non-retain Ø Ø Ø Ø 10 0 I € I ET Time T#Oms Non-retain Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø 10 0 I € I ET Time T#Oms Non-retain Ø Ø Ø Ø 10 0 I € I IN Bool false Non-retain Ø Ø Ø Ø 10 0 I € I ET Time T#Oms Non-retain Ø Ø Ø Ø 10 0 I € I ET Time T#Oms Non-retain Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø Ø 10 0 I € IN Bool false Non-retain Ø Ø Ø Ø Ø 10 0 I € IN Ø Ø I € I IN Ø Ø Ø Ø Ø 10 0 I € IN Ø Ø Ø Ø Ø 10 0 I € I € I IN Ø Ø Ø Ø 10 0 I € I € I € I € I € I € I € I € I €	8 🐔	Sensor_end_of_conveyor	Bool	false	Non-retain					Sensor part at end of conveyor
10 Conveyor_motor_automatic_ Bool false Non-retain Image: Static conveyor motor in automatic mote 10 Convergence Static Static Static conveyor motor in automatic conveyor motor in automatic conveyor motor in automatic conveyor	9 📶 🔻 Ou	utput								
11 G InOut ol ol <t< td=""><td>10 🕣 =</td><td>Conveyor_motor_automatic</td><td>Bool</td><td>false</td><td>Non-retain</td><td></td><td></td><td></td><td></td><td>Control of the conveyor motor in auto</td></t<>	10 🕣 =	Conveyor_motor_automatic	Bool	false	Non-retain					Control of the conveyor motor in auto
12 • <add new=""> 13 I Static Memory_automatic_start_stop Bool false Non-retain Memory_conveyor_start_stop Sool false Non-retain M V V V</add>	11 🕣 🔻 Ind	Dut								
13 41 ✓ Non-retain ✓ ✓ Memory used for start/ stop automatic 13 41 ✓ Memory_automatic_start_stop Bool false Non-retain ✓ ✓ Memory used for start/ stop automatic 15 41 ✓ Memory_conveyor_start_stop Bool false Non-retain ✓ ✓ Memory used for start/ stop automatic 15 41 ✓ Memory_conveyor_start_stop Bool false Non-retain ✓ ✓ Memory used for start/ stop of convey 16 41 ✓ Memory_conveyor_start_stop Bool false Non-retain ✓ ✓ Memory used for start/ stop of convey 17 41 ✓ Hero Non-retain ✓ ✓ ✓ Memory used for start/ stop of convey 18 41 ● PT Time T#Oms Non-retain ✓ ✓ ✓ ✓ 19 41 ● Q Bool false Non-retain ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓<	12 -	<add new=""></add>								
14 G Memory_automatic_start_stop Bool false Non-retain Image: Conveyor_start_stop Memory_conveyor_start_stop Bool false Non-retain Image: Conveyor_start_stop Memory_used for start/ stop automatic 15 G1 Memory_conveyor_start_stop Bool false Non-retain Image: Conveyor_start_stop Memory used for start/ stop automatic 16 G1 Memory_conveyor_start_stop Bool false Non-retain Image: Conveyor_start_stop Memory used for start/ stop automatic 17 G1 Image: Conveyor_start_stop Bool false Non-retain Image: Conveyor_start_stop Memory used for start/ stop automatic 17 G1 Image: Conveyor_start_stop Bool false Non-retain Image: Conveyor_start_stop Memory used for edge detection 19 G1 ET Time T#Oms Non-retain Image: Conveyor_start_stop Image: Conveyor_start_stop Memory used for edge detection 20 G2 Image: Conveyor_start_stop Bool false Non-retain Image: Conveyor_start_stop Image: Conveyor_start_stop 21 G2 Q Bool<	13 🕣 🔻 Sta	atic								
15 G Memory_conveyor_start_stop Bool false Non-retain Image: Conveyor conveyor_start_stop Memory used for start/stop of conveyor	14 🕣 🔳	Memory_automatic_start_stop	Bool	false	Non-retain					Memory used for start/ stop automatic
16 G Memory_edge_detection Bool false Non-retain Image: Control in automatic mode 17 G IEC_Timer_overrun TP_IM/E Non-retain Image: Control in automatic mode Image: Control in automatic mode Image: Control in automatic mode	15 🕣 🖷	Memory_conveyor_start_stop	Bool	false	Non-retain					Memory used for start/ stop of convey
17 41 a Vicc_Timer_overrun TP_TIME Non-retain ✓ ✓ ✓ 18 41 a PT Time T#Oms Non-retain ✓ ✓ ✓ 19 41 a PT Time T#Oms Non-retain ✓ ✓ ✓ 20 42 ET Time T#Oms Non-retain ✓ ✓ ✓ 21 42 a Q Bool false Non-retain ✓ ✓ ✓ 21 42 a Q Bool false Non-retain ✓ ✓ ✓ 21 42 a Q Bool false Non-retain ✓ ✓ ✓ 21 42 a Q Bool false Non-retain ✓ ✓ ✓ 23 5 Temm. ✓ ✓ ✓ ✓ ✓ ✓ 4 >=1 17 -ot -ot -ot -ot ✓ ✓ ✓ 8 >=1 17 -ot -ot -ot -ot ✓ <td< td=""><td>16 🕣 =</td><td>Memory_edge_detection</td><td>Bool</td><td>false</td><td>Non-retain</td><td></td><td></td><td></td><td></td><td>Memory used for edge detection</td></td<>	16 🕣 =	Memory_edge_detection	Bool	false	Non-retain					Memory used for edge detection
18 42 ■ PT Time T#0ms Non-retain ♥ ♥ ♥ 19 43 ■ ET Time T#0ms Non-retain ♥ ♥ ♥ 20 43 ■ IN Bool false Non-retain ♥ ♥ ♥ 20 43 ■ Q Bool false Non-retain ♥ ♥ ♥ 20 43 ■ Temm 4 >=1 12ol ↦ -f=]	17 🕣 = 🔻	IEC_Timer_overrun	TP_TIME		Non-retain					
19 C1 ■ ET Time T#0ms Non-retain ✓ ✓ 20 C2 ■ IN Bool false Non-retain ✓ ✓ ✓ 21 C3 ■ Q Bool false Non-retain ✓ ✓ ✓ 21 C3 ■ Q Bool false Non-retain ✓ ✓ ✓ 21 C3 ■ Q Bool false Non-retain ✓ ✓ ✓ ✓ 22 4T Temm. ✓ ✓ ✓ ✓ ✓ ✓ 22 4T Temm. ✓ ✓ ✓ ✓ ✓ ✓ 3 >=1 T1 -o1 → -j=j ✓ ✓ ✓ 3 >=1 T1 -o1 → -j=j ✓ ✓ ✓ 4 >=1 T1 -o1 → -j=j ✓ ✓ ✓	18 📶 💻	PT	Time	T#Oms	Non-retain		\checkmark	V		
20 42 IN Bool false Non-retain ✓ ✓ 21 42 4 Q Bool false Non-retain ✓ ✓ 21 42 • Bool false Non-retain ✓ ✓ ✓ 21 • • Bool false Non-retain ✓ ✓ 22 • • • ✓ ✓ ✓ ✓ 22 • • • ✓ ✓ ✓ ✓ 23 • • • ✓ ✓ ✓ ✓ 23 • • • • ✓ ✓ ✓ 24 • • • • ✓ ✓ ✓ 24 • • • • • ✓ ✓ 25 • • • • • ✓ ✓ 25 • • • • • ✓ ✓ 26 • • <td>19 🕣 🔹</td> <td>ET</td> <td>Time</td> <td>T#Oms</td> <td>Non-retain</td> <td>V</td> <td></td> <td></td> <td></td> <td></td>	19 🕣 🔹	ET	Time	T#Oms	Non-retain	V				
21 €21 ♥ Q Bool false Non-ret ♥ ♥ ♥ 22 60 ♥ Temm. ★ >=1 127 → -ol ↦ f=1 ♥ Block title: Motor control in automatic mode	20 🕣 🔹	IN	Bool	false	Non-retain	V	 Image: A start of the start of	Image: A start and a start		
272	21 🕣 🔹	Q	Bool	false	Non-ret					
8 >=1 1 -ol → -[=] ▼ Block title: Motor control in automatic mode	22 📶 🔻 Te	mp			_					×
a >=1 100 → -[+] ➡ Block title: Motor control in automatic mode	<				-					2
Block title: Motor control in automatic mode	& >=1	??]ol → -[=]								
block dife. motor condom automatic mode	Block titl	e: Motor control in automatic r	node							
Conveyor motor in automatic mode:	Conveyor m	otor in automatic mode:								
	. conveyor m	out an automotic modela.								
Network 1: Overrun time end of conveyor pulse 2 seconds	 Netwo 	rk 1: Overrun time end of con	veyor pulse	2 seconds						



 ® Vous pouvez régler le "Langage" sur CONT (schéma à contacts) dans l'onglet "General" (Général) des propriétés du bloc. (® Properties (Propriétés) ® General (Général) ® Language (Langage) : CONT)

MOTOR_AUTO [FB1]		cs 🗖 🗖 🗖 🗖				
General FB supervision	definitions					
General Information Time stamps Compilation Protection Attributes Download without reinitializati	General	Name: Type: Language: Number:	MOTOR_AUTO FB FBD LAD EBD)		
			Manual Manual Automatic 	:		

® Les réseaux 1 et 2 sont représentés dans le schéma à contacts (CONT) comme suit.



7.3 Actualiser l'appel de bloc dans le bloc d'organisation

® Ouvrez le bloc d'organisation "Main [OB1]" en double-cliquant dessus.



® Dans le réseau 1 du bloc d'organisation "Main [OB1]", le bloc de données d'instance "MOTOR_AUTO_DB1" du bloc fonctionnel "MOTOR_AUTO [FB1]" n'apparaît pas correctement car la mémoire supplémentaire pour la temporisation TP n'y est pas encore créée. Cliquez sur l'icône ® "¹ "Actualiser appels de bloc incohérents". Cela permet de recréer le bloc de données d'instance "MOTOR_AUTO_DB1" correctement. (® ¹)

kă ∌ ∉ 🐛 🖿 🚍 💬 🕾	±월±월±달 🈥 🥙 🐅 徳 锡 🕸 💷 🏣 📢 🕴 🔗 🥲 '	' 🗖
Block title: "Main Program Sweep (Cycl	e)"	
ommenc		
Network 1: Control conveyor motor	forwards in automatic mode	
Comment		
	DB*	
	%FB1	
	"MOTOR_AUTO"	
	EN EN	
	%40.2 Automatic_ "-50" — mode active	
&	%0.3	
%JO.1	"-51" — Start	
-ко —	%0.4	
%0.5	-S2" -O Stop	
·•• · · · · · · · · · · · · · · · · · ·	Enable_OK	
	"-A1" — shutoff_active	
	%11.0 Conveyor_	
	"-B4" — Sensor_slide automatic_ %Q0.0	
	%11.3 Sensor_end_ mode — "-Q1"	
	-B7 — of conveyor ENO —	

Utilisation libre pour les instituts publics de formation et de R&D. © Siemens AG 2018. Tous droits réservés. SCE_FR_031-300 IEC-Times and Counters S7-1200_R1709.docx

7.4 Enregistrer et compiler le programme

ject Edit View Insert Online Options 🔁 🔒 Save project 📑 💥 🗎 🗎 🗶 🏷	Tools Window Help 🛨 (# ± 🛅 🛄 🛐 🔛 🏹 🂋 Goonline 🔉	🖉 Go offline 🛔 🖪 🖪 🗶 📥 💷 🤇 Search in projects 🔒	Totally Integrated Automation PORT/
Project tree 🔲 🛛	✓)0_IEC_Timers_Counters → CPU_121		💶 🖥 🗮 🗙 Instructions 📰 🔳
Devices			Options
	🗄 🗟 🖉 🛒 🐛 🖿 🚍 💬 😂 :	± 🛛 ± 🖼 ± 🚍 🌮 伦 🕼 🖑 🖓 🕼 📲 🐂 📢 🔗	🖞 😳 ' Thi thi
		Block interface	> Eavorites
O31-300_IEC_Timers_Counters	Block title: "Main Program Sween (Cycle)* * =*	A M Basic instructions
Add new device	Comment		Viene Description
Devices & networks			Name Description
CPU_1214C [CPU 1214C DC/DC/DC]	 Network 1: Control conveyor motor f 	forwards in automatic mode	General
Device configuration	Comment		HI Bit logic operations
& Online & diagnostics	Connert		Imer operations
🔻 🔂 Program blocks			+1 Counter operations
Add new block		%DB1	Comparator operations
Main [OB1]		DB*	the math functions
MOTOR_AUTO [FB1]		YER1	Move operations
MOTOR_AUTO_DB [DB1]		"MOTOR AUTO"	Conversion operations
Technology objects		- 14	Program control operati
External source files			Word logic operations
PLC tags		40.2 Automatic_	Shift and rotate
PLC data types		-50 — mode_active	
Watch and force tables	à	%0.3	
Online backups	%0.1	-start	
Traces		940.4	
Device proxy data	%10.5	-32 - Stop	
Program info	-61 - 4	Enable_OK	
PIC alarm text lists		%10.0 Safety_	
Local modules		-AI -O shutom_active Conveyor	
		%1.0 motor_	
Common data		-84 — Sensor_slide automatic_ %Q0.0	
Documentation settings		%1.3 Sensor_end_ mode — "-Q1"	
languages & resources		B/ — of_conveyor ENO —	
Card Reader/USB memory	Vetwork 2:		< II
Car cord neodenoso meniory	Technik 2.		Extended instructions
	Comment		V > Technology
	<	> 100%	Communication
Details view		Properties	Contional packages
Details view		Sino Sino Magnostics	V Optional packages

General (1) Cross-reference	es Compile Energy Suite Syntax				
🕄 🛓 🕕 Show all messages					
Compiling finished (errors: 0; warni	ngs: 0)				
! Path	Description	Go to	? Errors	Warnings	Time
✓ ▼ CPU_1214C		~	0	0	4:48:34 PM
📀 🔻 Program blocks		~	0	0	4:48:34 PM
MOTOR_AUTO (FB1)	Block was successfully compiled.	~			4:48:34 PM
Main (OB1)	Block was successfully compiled.	~			4:48:37 PM
v	Compiling finished (errors: 0; warnings: 0)				4:48:37 PM
- m.					

7.5 Charger le programme

Ine fois que la compilation s'est correctement déroulée, vous pouvez charger toute la commande avec le programme créé, y compris la configuration matérielle, comme cela a été décrit dans les modules précédents. (

Siemens - C:\Users\mde\Documents\Automa	on\031-300_IEC_Timers_Counters\031-300_IEC_Timers_Counters		_ 0
Project Edit View Insert Online Options	ools Window Help (# ± 🖥 🖳 🕼 🚆 🎧 💋 Goonline 🖉 Gooffiine 🎄 🖪 🖪	🛪 📃 🔢 <search in="" project=""> 🕌</search>	Totally Integrated Automation PORTAL
Project tree 🔲 🕯	031-300_IEC_Timers_Counters CPU_1214C [CPU 1214C DC/D	DC/DC] → Program blocks → Main [OB1] 🛛 🗕 🖬 i	🗆 🗙 Instructions 📰 🗈 🕨
Devices			Options
F# T		C. # CH 42 CH IL 14 CH M 28 00 Ch	F3
	Block interfa		A Foundation
▼ 1031-300 JEC Timers Counters		Real Providence of the International Providence of the Interna	7 Favorites
Add new device	 Block title: "Main Program Sweep (Cycle)" 		▲ Y Basic instructions
Devices & networks	Comment		Name Des
CPU 1214C [CPU 1214C DC/DC/DC]	 Network 1: Control conveyor motor forwards in automatic mode 		General
Device configuration	Comment		Bit logic operations
Q Online & diagnostics	comment		G Timer operations
Program blocks			the second
Add new block		%DB1	Comparator operations
Main [OB1]		DB"	the second
MOTOR_AUTO [FB1]		%FB1	Move operations
MOTOR_AUTO_DB [DB1]		"MOTOR_AUTO"	Conversion operations
Technology objects	- EN		Program control operation
External source files	910 2		Word logic operations
PLC tags	"-50" — mode	atic_	- F - Shift and rotate
PLC data types	8 940.2		
Watch and force tables	"-51" — Start		
Online backups	*K0*		
🕨 📴 Traces	\$40.5 *-52* -•• Stop		
Device proxy data	"-81" — # Enable	• OK	
Program info	%0.0 Safety		
PLC alarm text lists	"-A1" - • shutoff	factive	
Local modules	\$10	Conveyor_	
Lagrouped devices	*-84* — Sensor	r slide automatic %00.0	
Common data	\$13 Second	mode - "-Q1"	
Documentation settings	"-87" — of com	veyor ENO -	
Languages & resources	-		
Online access			
Card Reader/USB memory	Network 2:		> Extended instructions
-	Comment		> Taskasland
			V I lechnology
		> 100%	Communication
> Details view		Properties 🚺 Info 🔒 🔽 Diagnostics	Optional packages
Portal view Overview	Main (OB1)	🔜 🗸 T	he project 031-300_IEC_Timers_Coun

7.6 Visualiser des blocs de programme

® Pour visualiser le programme chargé, le bloc souhaité doit être ouvert. Vous pouvez ensuite

désactiver/activer la visualisation en cliquant sur l'icône 😬. (
 Main [OB1]



Remarque : ici, la visualisation s'effectue sur la base des signaux et elle dépend de la commande. Les états logiques des bornes sont représentés par TRUE ou FALSE.

Il est possible, après avoir fait un clic droit de la souris, de sélectionner directement le bloc fonctionnel "MOTOR_AUTO" [FB1] appelé dans le bloc d'organisation "Main [OB1]" à "Open and monitor" (Ouvrir et surveiller) et de surveiller ainsi le code du programme présent dans le bloc fonctionnel ayant le temporisateur TP. (® "MOTOR_AUTO" [FB1] ® Open and monitor (Ouvrir et surveiller))



Remarque : ici, la visualisation s'effectue selon les fonctions et elle dépend de la commande. L'actionnement des capteurs ou l'état de l'installation est représenté par TRUE ou FALSE.

7.7 Archiver le projet

Pour terminer, nous voulons archiver le projet complet. Sélectionnez dans le menu ® "Project" (Projet) la commande ® "Archive..." (Archiver). Choisissez un dossier dans lequel vous souhaitez archiver votre projet et enregistrez-le sous "Archives projets TIA Portal" comme type de fichier. (® Project (Projet) ® Archive... (Archiver) ® Archives projets TIA Portal ® SCE_FR_031-300_Temporisations_CEI _compteurs_S7-1200... ® Save (Enregistrer))

Ma Siemens - C:\Users\mde\Documents\A	mation/031-300_IEC_Timers_Counters/031-300_IEC_Timers_Counters	_ 0)
Project Edit View Insert Online Op	is Tools Window Help) ± (# ± 🖥 🗓 🛱 🔛 🧊 🖉 Go online 🖉 Go offline 🔐 🌆 🕼 🧩 🛃 🛄 🏼 Search in projects 🐐	Totally Integrated Automation PORTAL
Open Cti	✓300_IEC_Timers_Counters > CPU_1214C [CPU_1214C DC/DC/DC] > Program blocks > Main [OB1] ■	🗙 Testing 💼 🗈 🗈 🖉
Close Cti		Ontions
Save Ct		
Save as Ctrl+Shi	ᇍᅝᅊᄻᇍᇍᆐᄃᇊᆖᇟᇃᆂᄺᆂᇃᆋᅚᇊᅆᇈᇲᅆᇪᇫᄹᇩᇎᇎᆐᇉᅆᅶᆝ	
Delete project	Block interface	CPU operator panel
Archive	 Network 1: Control conveyor motor forwards in automatic mode 	No online connection
Retrieve	Comment	The second second second second
Manage multiuser server projects		-
To Card Reader/USB memory Card Reader/USB memory Memory card file	"MOTOR_AUTO_ BB"	l esting
Start basic integrity check Upgrade	%F81 "MOTOR_AUTO"	
Print Ct Print preview	" " "	Call environment No condition defined.
C:IUsersImdelDa1300_IEC_Timers_Counte C:IUsersImdelD1031-200_FB_trogrammin C:IUsersImdelDocume1011-101_CPU1214 C:IUsersImdelD1031-100_FC_Programmin C:IUsersImdelDocumen1011-100_CPU120		Change
Exit Alt	940.0 Safety	
Pevice providata	"-A1" - o shutoff_active	✓ Breakpoints
Program info	%1.0 Conveyor_ motor_	êx± e ³ ♥ ●> HI 6∃ 6∃ 5∃
PLC alarm text lists	-B4" Sensor_slide automatic_ %Q0.0	× Call hierarchy
Local modules	%1.3 Sensor_end_ FNO	can menerary
Generation Ungrouped devices	of <u>of</u> conveyor <u>eno</u>	
Common data		
Languages & resources	Network 2:	
Online access	Comment	No collection of the late
Card Reader/USB memory		No call path available
	K III 100%	~
> Details view	Properties 1 Info 🔒 V. Diagnostics	B 🔺
Portal view Dverview	A Main (081)	The project 031-300 JEC Timers Coup

7.8 Check-list

Nº	Description	Contrôlé
1	La compilation s'est déroulée correctement et sans message de d'erreur	
2	Le chargement s'est déroulé correctement et sans message de d'erreur	
3	Mise en circuit de l'installation (-K0 = 1) Tige du vérin rentrée / signalisation en retour activée (-B1 = 1) ARRET D'URGENCE (-A1 = 1) pas activé Mode AUTOMATIQUE (-S0 = 1) Bouton d'arrêt automatique pas actionné (-S2 = 1) Appui bref sur le bouton de démarrage automatique (-S1 = 1) Capteur glissière occupée activé (-B4 = 1) Le moteur du convoyeur déclenche l'avance à vitesse fixe (-Q1 = 1) et reste activé.	
4	Capteur de fin de bande activé (-B7 = 1) $\ensuremath{\mathbb{B}}$ -Q1 = 0 (après 2 secondes	
5	Appui bref sur le bouton d'arrêt automatique (-S2 = 0) $\ensuremath{\mathbb{B}}$ -Q1 = 0	
6	Activation ARRET D'URGENCE (-A1 = 0) $\ensuremath{\mathbb{B}}$ -Q1 = 0	
7	Mode Manuel (-S0 = 0) ® -Q1 = 0	
8	Mise hors circuit de l'installation (-K0 = 0) \textcircled{B} -Q1 = 0	
9	Tige du vérin pas rentrée (-B1 = 0) ® -Q1 = 0	
10	Projet archivé correctement	

8 Exercice

8.1 Énoncé du problème – Exercice

Dans cet exercice, nous voulons compléter le bloc fonctionnel MOTOR_AUTO [FB1] avec un compteur CEI. Le bloc fonctionnel ainsi complété sera ensuite planifié, programmé et testé :

Le magasin pour plastique ne peut accueillir que 5 pièces. C'est pourquoi le comptage des pièces a lieu à la fin du convoyeur.

Le mode automatique doit être arrêté lorsque 5 pièces sont chargées dans le magasin.

Une fois le magasin vidé, le mode automatique est redémarré par une nouvelle Commande_Démarrage et le compteur, réinitialisé.

8.2 Schéma technologique

La figure ci-dessous montre le schéma technologique pour l'application à réaliser.



Figure 3 : Schéma technologique



Figure 4 : Pupitre de commande

8.3 Tableau d'affectations

DE	Туре	Code	Fonction	NC/NO
E 0.0	BOOL	-A1	Message ARRET D'URGENCE ok :	NC
E 0.1	BOOL	-K0	Installation "Marche"	NO
E 0.2	BOOL	-S0	Commutateur mode Manuel (0)/ Automatique (1)	Manuel = 0 Auto=1
E 0.3	BOOL	-S1	Bouton démarrage automatique	NO
E 0.4	BOOL	-S2	Bouton arrêt automatique	NC
E 0.5	BOOL	-B1	Capteur tige du vérin -M4 rentrée	NO
E 1.0	BOOL	-B4	Capteur glissière occupée	NO
E 1.3	BOOL	-B7	Capteur pièce à la fin du convoyeur	NO

Cette application requiert les signaux suivants comme opérande global.

DA	Туре	Code	Fonction	
A 0.0	BOOL	-Q1	Moteur du convoyeur -M1 avance à vitesse fixe	

Legende zur Belegungsliste

DE	Entrée TOR	DA	Sortie TOR
AE	Entrée analogique	AA	Sortie analogique
Е	Entrée	А	Sortie

- NC Normally Closed (contact à ouverture)
- NO Normally Open (contact à fermeture)

8.4 Planification

Et maintenant, planifiez vous-même la mise en œuvre de l'application à réaliser.

Remarque : vous trouverez des informations sur l'utilisation des compteurs CEI dans SIMATIC S7-1200 dans l'aide en ligne.

8.5 Check-list – Exercice

Nº	Description	Contrôlé
1	La compilation s'est déroulée correctement et sans message de d'erreur	
2	Le chargement s'est déroulé correctement et sans message de d'erreur	
3	Mise en circuit de l'installation (-K0 = 1) Tige du vérin rentrée / signalisation en retour activée (-B1 = 1) ARRET D'URGENCE (-A1 = 1) pas activé Mode AUTOMATIQUE (-S0 = 1) Bouton d'arrêt automatique pas actionné (-S2 = 1) Appui bref sur le bouton de démarrage automatique (-S1 = 1) Capteur glissière occupée activé (-B4 = 1) Le moteur du convoyeur déclenche l'avance à vitesse fixe (-Q1 = 1) et reste activé.	
4	Capteur de fin de bande activé (-B7 = 1) \textcircled{B} -Q1 = 0 (après 2 secondes	
5	Appui bref sur le bouton d'arrêt automatique (-S2 = 0) $\$ -Q1 = 0	
6	Activation ARRET D'URGENCE (-A1 = 0)	
7	Mode Manuel (-S0 = 0) ® -Q1 = 0	
8	Mise hors circuit de l'installation $(-K0 = 0)$ \textcircled{B} $-Q1 = 0$	
9	Tige du vérin pas rentrée (-B1 = 0) ® -Q1 = 0	
10	5^{e} pièce dans le magasin \mathbb{B} -Q1 = 0	
11	Projet archivé correctement	

9 Informations complémentaires

Pour vous aider à vous familiariser ou à approfondir vos connaissances, des informations complémentaires tels que mise en route, vidéos, didacticiels, applis, manuels, guide de programmation et logiciel/firmware de démonstration sont disponibles sous le lien suivant :

www.siemens.com/sce/s7-1200

Vue d'ensemble des "Informations complémentaires"

- Getting Started, Videos, Tutorials, Apps, Manuals, Trial-SW/Firmware
 - ↗ TIA Portal Videos
 - ↗ TIA Portal Tutorial Center
 - Getting Started
 - ↗ Programming Guideline
 - ↗ Easy Entry in SIMATIC S7-1200
 - > Download Trial Software/Firmware
 - ↗ Technical Documentation SIMATIC Controller
 - ↗ Industry Online Support App
 - TIA Portal, SIMATIC S7-1200/1500 Overview
 - ↗ TIA Portal Website
 - ↗ SIMATIC S7-1200 Website
 - ↗ SIMATIC S7-1500 Website

Plus d'informations

Siemens Automation Cooperates with Education siemens.com/sce

Supports d'apprentissage/de formation siemens.com/sce/documents

Packages SCE pour formateurs siemens.com/sce/tp

Partenaires SCE siemens.com/sce/contact

L'entreprise numérique siemens.com/digital-enterprise

Industrie 4.0 siemens.com/future-of-manufacturing

Totally Integrated Automation (TIA) siemens.com/tia

TIA Portal siemens.com/tia-portal

Automates SIMATIC siemens.com/controller

Documentation technique SIMATIC siemens.com/simatic-docu

Industry Online Support support.industry.siemens.com

Catalogue de produits et système de commande en ligne Industry Mall mall.industry.siemens.com

Siemens AG Digital Factory P.O. Box 4848 90026 Nuremberg Allemagne

Sous réserve de modifications et d'erreurs © Siemens AG 2018

siemens.com/sce