

Support d'apprentissage/ de formation

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

Module 031-100 TIA Portal

Principes de base de la programmation de FC avec SIMATIC S7-1200

siemens.com/sce



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	Obj	ectif	5
2	Con	dition	5
3	Con	figurations matérielles et logicielles requises	6
4	Part	ie théorique	7
	4.1	Système d'exploitation et programme utilisateur	7
	4.2	Blocs d'organisation	8
	4.3	Mémoire image et traitement cyclique du programme	9
	4.4	Fonctions	.11
	4.5	Blocs fonctionnels et blocs de données d'instance	.12
	4.6	Blocs de données globaux	.13
	4.7	Blocs de code compatibles avec la bibliothèque	.14
	4.8	Langages de programmation	.15
5	Арр	lication à réaliser	.16
6	Plar	nification	.16
	6.1	ARRÊT D'URGENCE	.16
	6.2	Mode manuel – Moteur du convoyeur en marche par impulsion	.16
	6.3	Schéma technologique	.17
	6.4	Tableau d'affectations	.18
7	Mar	che à suivre détaillée	.19
	7.1	Désarchiver un projet existant	.19
	7.2	Création d'une nouvelle table des variables	.20
	7.3	Création de nouvelles variables dans une table de variables	.22
	7.4	Importation de la "Table des variables_installation de tri"	.23
	7.5	Création de la fonction FC1 "MOTOR_HAND" pour le moteur du convoyeur en marche par impulsion	27
	7.6	Définition de l'interface de la fonction FC1 "MOTOR_HAND"	.29
	7.7	Programmation du FC1 : MOTOR_HAND	.32
	7.8	Programmation du bloc d'organisation OB1 – commande de l'avance du convoyeur en mode manuel.	.39
	7.9	Enregistrer et compiler le programme	.44
	7.10	Charger le programme	.45
	7.11	Visualiser des blocs de programme	.46
	7.12	Archiver le projet	.48
	7.13	Check-list	.49
8	Exe	rcice	.50
	8.1	Énoncé du problème – Exercice	.50
	8.2	Schéma technologique	.50
	8.3	Tableau d'affectations	.51
	8.4	Planification	.51

8	5.5	Check-list – Exercice	52
9	Info	prmations complémentaires	53

Principes de base de la programmation de FC

1 Objectif

Dans le présent chapitre, vous allez vous familiariser avec les éléments de base d'un programme : les *blocs d'organisation (OB)*, les *fonctions (FC)*, les *blocs fonctionnels (FB)* et les *blocs de données (DB)*. Par ailleurs, nous vous montrerons comment programmer les fonctions et blocs fonctionnels compatibles avec la *bibliothèque*. Vous allez vous familiariser avec le langage de programmation *logigramme (LOG)* et utiliser celui-ci pour programmer une fonction FC1 et un bloc d'organisation OB1.

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

2 Condition

Ce chapitre s'appuie sur la configuration matérielle de la CPU1214C SIMATIC S7. Toutefois, il peut également être travaillé avec toute autre configuration matérielle munie d'une carte d'entrée/sortie TOR. Pour l'étude de ce chapitre, vous pouvez par exmple recourir au projet suivant :

SCE_FR_011_101_Configuration matérielle_CPU1214C.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 Système d'exploitation et programme utilisateur

Chaque automate (CPU) contient un **système d'exploitation** qui organise toutes les fonctions et processus de la CPU n'étant pas liés à une tâche d'automatisation spécifique. Font partie des tâches du système d'exploitation :

- Déroulement du démarrage (à chaud)
- Actualisation de la mémoire image des entrées et de la mémoire image des sorties
- Appel cyclique du programme utilisateur
- Acquisition des alarmes et appels des OB d'alarme
- Détection et traitement des erreurs
- Gestion des zones de mémoire

Le système d'exploitation est un composant de la CPU et est déjà installé dans la CPU à la livraison.

Le **programme utilisateur** contient toutes les fonctions requises pour le traitement de tâches d'automatisation spécifiques. Font partie des fonctions du programme utilisateur :

- Vérification des conditions préalables au démarrage (à chaud) à l'aide d'OB de démarrage
- Traitement des données du processus, c'est-à-dire commande des signaux de sortie en fonction des états des signaux d'entrée
- Réaction aux alarmes et entrées d'alarmes
- Traitement des perturbations dans l'exécution normale du programme

4.2 Blocs d'organisation

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation de l'automate (CPU) et le programme utilisateur. Ils sont appelés par le système d'exploitation et gère les opérations suivantes :

- Traitement cyclique du programme (par ex. OB1)
- Comportement au démarrage de l'automate
- Traitement du programme déclenché par alarme
- Traitement des erreurs

Un projet doit contenir au moins *un bloc d'organisation pour le traitement cyclique du programme*. Un OB est appelé par un *évènement déclencheur* comme représenté dans la Figure 1. Des priorités sont définies pour les différents OB afin que l'OB1 cyclique puisse par exemple être interrompu par un OB82 pour le traitement des erreurs.



Figure 1 : Évènements déclencheurs dans le système d'exploitation et appel d'OB

Les réactions suivantes sont possibles après qu'un évènement déclencheur s'est produit :

- Si vous avez affecté un OB à l'événement, il déclenchera l'exécution de l'OB affecté. Si la priorité de l'OB affecté est plus élevée que celle de l'OB en cours d'exécution, celui-ci est immédiatement exécuté (interruption). Si ce n'est pas le cas, le système attend d'abord jusqu'à ce que l'exécution de l'OB avec la priorité plus élevée soit terminée.
- Si l'événement n'est affecté à aucun OB, la réaction système par défaut est exécutée.

Le tableau 1 ci-dessous montre différents exemples d'évènements déclencheurs pour un SIMATIC S7-1200. Il contient aussi des numéros d'OB possibles et les réactions système prédéfinies qui sont exécutées lorsque le bloc d'organisation (OB) correspondant n'est pas présent dans l'automate.

Évènement déclencheur	Numéros d'OB possibles	Réaction système prédéfinie
Mise en route	100, ³ 123	Ignorer
Programme cyclique	1, ³ 123	Ignorer
Alarme horaire	10 à 11	-
Alarme de mise à jour	56	Ignorer
Temps de cycle imparti dépassé une fois	80	Ignorer
Temps de cycle imparti dépassé deux fois	80	STOP
Alarme de diagnostic	82	Ignorer

Tableau 1 : Numéros d'OB pour différents évènements déclencheurs

4.3 Mémoire image et traitement cyclique du programme

Si les entrées (I) et sorties (Q) sont adressées dans le programme utilisateur cyclique, les états des signaux ne sont pas interrogés directement par les modules d'entrées/sorties mais il est accédé à la zone de mémoire de la CPU. Cette zone de mémoire contient une image des états des signaux et est appelée **mémoire image**.

Le traitement cyclique du programme se déroule comme suit :

- Au début du programme cyclique, le système vérifie si chacune des entrées est sous tension ou non. L'état de ces entrées est enregistré dans la mémoire image des entrées (MIE). Si l'entrée est sous tension, l'information 1 ou "High" sera enregistrée. Si l'entrée n'est pas sous tension, l'information 0 ou "Low" sera enregistrée.
- Le processeur exécute alors le programme stocké dans le bloc d'organisation cyclique. L'information d'entrée requise à cet effet est prélevée dans la mémoire image des entrées (MIE) lue auparavant et les résultats logiques sont écrits dans une mémoire image des sorties (MIS).
- 3. A la fin du cycle, la **mémoire image des sorties** (**MIS**) est transférée comme état logique aux modules de sorties et celles-ci sont activées ou désactivées. La procédure reprend ensuite à partir du point 1.



3. Transmettre l'état de la MIS aux sorties.

Figure 2 : Traitement cyclique du programme

Remarque : le temps requis par le processeur pour l'exécution du programme s'appelle le temps de cycle. Ce dernier dépend entre autres du nombre et du type d'instructions ainsi que de la puissance du processeur de l'automate.

4.4 Fonctions

Les fonctions (FC) sont des blocs de code sans mémoire. Elles **n'ont pas de mémoire de données** dans laquelle il est possible d'enregistrer les valeurs de paramètres de bloc. C'est pourquoi tous les paramètres d'interface doivent être interconnectés lors de l'appel d'une fonction. Des blocs de données globaux doivent être crées pour stocker durablement les données.

Une fonction contient un programme qui est toujours exécuté quand un autre bloc de code appelle cette fonction.

Les fonctions peuvent par exemple servir dans les cas suivants :

- Retourner un résultat dépendant des valeurs d'entrée pour les fonctions mathématiques.
- Exécuter des fonctions technologiques comme des commandes uniques avec combinaisons binaires.

Une fonction peut également être appelée plusieurs fois à divers endroits du programme.



Figure 3 : Fonction avec appel d'un bloc d'organisation Main [OB1]

4.5 Blocs fonctionnels et blocs de données d'instance

Les blocs fonctionnels sont des blocs de code qui mémorisent durablement leurs variables d'entrée, de sortie et d'entrée/sortie ainsi que leurs variables statiques dans des blocs de données d'instance afin qu'il soit possible d'y accéder même **après le traitement de blocs**. Pour cette raison, ils sont aussi appelés blocs avec mémoire.

Les blocs fonctionnels peuvent aussi travailler avec des variables temporaires. Cependant, les variables temporaires ne sont pas enregistrées dans le DB d'instance mais disponibles uniquement tout le temps d'un cycle.

Les FB sont utilisés pour des tâches qui ne peuvent être mises en œuvre avec des fonctions :

- Toujours quand les temporisations et les compteurs sont nécessaires dans un bloc ou
- toujours quand une information doit être enregistrée dans le programme. Par ex. un indicatif de mode de fonctionnement avec un bouton.

Les FB sont toujours exécutés quand un bloc fonctionnel est appelé par un autre bloc de code. Un FB peut aussi être appelé plusieurs fois à divers endroits du programme. Ceci facilite la programmation de fonctions complexes et répétitives.

Un appel d'un bloc fonctionnel est désigné par le terme "instance". Pour chaque instance d'un FB, une zone mémoire lui est affectée, contenant les données utiles au traitement du bloc. Cette mémoire est fournie par des blocs de données que le logiciel génère automatiquement.

Il est également possible de fournir de la mémoire pour plusieurs instances dans un bloc de données sous forme de **multi-instance**. La taille maximale des DB d'instance varie selon la CPU. Les variables déclarées dans le bloc fonctionnel déterminent la structure du bloc de données d'instance.



Figure 4 : Bloc fonctionnel et instance avec appel d'un bloc d'organisation Main [OB1]

4.6 Blocs de données globaux

Contrairement aux blocs de code, les blocs de données ne contiennent pas d'instructions, mais ils sont utilisés pour enregistrer les données utilisateur.

Les blocs de données contiennent donc des données variables qui sont utilisées dans le programme utilisateur. La structure des blocs de données globaux peut être définie au choix.

Les blocs de données globaux stockent des données qui peuvent être utilisés *par tous les autres blocs* (voir figure 5). L'accès aux blocs de données d'instance doit être réservé au bloc fonctionnel correspondant. La taille maximale des blocs de données varie selon la CPU.



Figure 5 : Différence entre bloc de données global et bloc de données d'instance.

Exemples d'application pour les blocs de données globaux :

- Enregistrement des informations pour la gestion d'un magasin. "Où se trouve quel produit ?"
- Enregistrement des recettes de produits donnés.

4.7 Blocs de code compatibles avec la bibliothèque

Un programme utilisateur peut être créé de façon linéaire ou structurée. La **programmation linéaire** consiste à écrire le programme utilisateur complet dans l'OB de cycle. Cela n'est toutefois recommandé que pour des programmes simples pour lesquels on utilise désormais d'autres systèmes de commande plus économique telle que LOGO!

Une *programmation structurée* est recommandée pour des programmes plus complexes. Vous pouvez subdiviser la tâche d'automatisation complexe en plusieurs petites tâches partielles à réaliser par des fonctions et blocs fonctionnels.

Il convient de créer des blocs de code compatibles avec la bibliothèque pour cela. Autrement dit, les paramètres d'entrée et les paramètres de sortie d'une fonction ou d'un bloc fonctionnel sont définis de manière générale et les variables globales actuelles (entrées/sorties) ne leurs sont attribuées que lors de l'utilisation du bloc.



Figure 6 : Fonction compatible de bibliothèque avec appel dans OB1

4.8 Langages de programmation

Les langages de programmation logigramme (LOG), schéma à contacts (CONT) et Structured Control Language (SCL) sont disponibles pour la programmation de fonctions et blocs fonctionnels pour SIMATIC S7-1200.

Le langage de programmation logigramme (LOG) est expliqué ci-après.

LOG est un langage de programmation graphique. La représentation est inspirée des systèmes de circuits électroniques. Le programme est représenté dans divers réseaux. Un réseau contient un ou plusieurs chemins logiques. Les signaux binaires et analogiques sont combinés entre eux par des boîtes. Pour représenter la logique, on utilise les symboles logiques graphiques connus de l'algèbre booléenne.

Avec les fonctions binaires, vous pouvez interroger les opérandes binaires et combiner leurs états logiques. Les instructions "Opération logique ET", "Opération logique OU" et "Opération logique OU EXCLUSIF" sont des exemples de fonctions binaires comme représenté dans la Figure 7 ci-dessous.



Figure 7 : Fonctions binaires dans LOG et table logique correspondante

Les instructions simples permettent par ex. de forcer des sorties binaires, d'évaluer les fronts ou d'exécuter des fonctions de saut dans le programme.

Des éléments de programme comme des temporisations CEI et des compteurs CEI mettent à disposition des instructions plus complexes.

La boîte vide est un emplacement réservé dans lequel vous pouvez sélectionner l'instruction voulue.

Mécanisme d'entrée de validation EN (enable)/sortie de validation ENO (ENable Output) :

- Une instruction sans mécanisme EN/ENO est exécutée indépendamment de l'état logique au niveau des entrées de la boîte.
- Des instructions avec mécanisme EN/ENO ne sont exécutées que si l'état logique de l'entrée de validation EN est "1". Si le traitement de la boîte est correct, la sortie de validation ENO est à l'état logique "1". Si des erreurs se produisent en cours de traitement, la sortie de validation ENO est remise à zéro. Si l'entrée de validation EN n'est pas imbriquée, la boite est toujours exécutée.

5 Application à réaliser

Dans le présent chapitre, nous voulons planifier, programmer et tester les fonctions de processus de l'installation de tri suivante :

 Mode manuel – commande de l'avance du convoyeur en mode manuel / marche par impulsion.

6 Planification

Il n'est pas conseillé de programmer toutes les fonctions dans l'OB1 par souci de clarté et pour ne pas restreindre les possibilités de réutilisation. C'est pourquoi le code du programme est principalement contenu dans des fonctions (FC) et des blocs fonctionnels (FB). La décision visant à déterminer les fonctions à affecter aux FC et celles à exécuter dans l'OB1 sera planifié ci-après.

6.1 ARRÊT D'URGENCE

L'ARRÊT D'URGENCE ne requiert pas une fonction propre. Tout comme le mode de fonctionnement, l'état actuel du relais ARRÊT D'URGENCE peut être utilisé directement par les blocs.

6.2 Mode manuel – Moteur du convoyeur en marche par impulsion

Nous voulons intégrer la marche par impulsion du moteur du convoyeur dans une fonction (FC) "MOTOR_HAND". D'une part, cela permet d'assurer la clarté de l'OB1 et, d'autre part, de conserver les possibilités de réutilisation en cas d'extension de l'installation avec un convoyeur supplémentaire. Le Tableau 2 ci-après indique les paramètres planifiés.

Input	Type de données	Commentaire
Manuel_activé	BOOL	Modes de fonctionnement manuel activé
Bouton_marche par impulsion	BOOL	Bouton-poussoir pour activer la marche par impulsion du moteur du convoyeur.
Validation_OK	BOOL	Toutes les conditions de validation sont remplies
Arrêt automatique de sécurité_activé	BOOL	Dispositif d'arrêt automatique de sécurité, par ex. arrêt d'urgence actionné
Output		
Moteur du convoyeur_marche par impulsion	BOOL	Commander le moteur du convoyeur en marche par impulsion

Tableau 2 : Paramètre pour FC "MOTOR_HAND"

La sortie Moteur du convoyeur_marche par impulsion est activée tant que le Bouton_marche par impulsion est enfoncé, le mode de fonctionnement manuel est activé, la validation est donnée et que le dispositif d'arrêt automatique de sécurité n'est pas actif.

6.3 Schéma technologique

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



Figure 8 : Schéma technologique

Schalter der Sortieranlage Switches of sorting station	Automatikbetrieb Automatic mode	Handbetrieb / Manual mode -S3 Tippbetrieb -M1 vorwärts/ Manual -M1 forwards
-P1 ein/on -Q0 Hauptschalter/Main switch -P4 aktivier/active	-P5 gestartet/started	-S4 Tippbetrieb -M1 rückwärts/ Manual -M1 backwards
-A1 NOTHALT/Emergency stop	S2 Stopp/stop	-P7 ausgefahren/extended -S6 Zylinder -M4 ausfahren/ cylinder -M4 extend -P6 eingefahren/retracted
-S0 Betriebsart/operating mode		-S5 Zylinder -M4 einfahren/ cylinder -M4 retract

Figure 9 : Pupitre de commande

6.4 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 0.5	BOOL	-B1	Capteur tige du vérin -M4 rentrée	NO
E 1.4	BOOL	-S3	Bouton-poussoir Avance en marche par à-coups du convoyeur M1	NO
E 1.5	BOOL	-S4	Bouton-poussoir Recule en marche impulsion du convoyeur M1	NO

Cette application requiert les signaux suivants comme opérande.

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 correspondants 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 commencer la programmation de la fonction (FC) "MOTOR_HAND", nous avons besoin d'un projet avec une configuration matérielle (par ex. SCE_FR_011-101_Configuration matérielle_CPU1214C.zap14). Pour désarchiver un projet existant, vous devez sélectionner l'archive correspondant sous ® Project (Projet) ® (Retrieve) Extraire dans la vue du projet. Confirmez votre sélection avec Ouvrir. (® Project (Projet) ® (Retrieve) Extraire ® Sélection d'une archive .zap ® Open (Ouvrir))



® Ensuite, vous pouvez sélectionner le répertoire cible dans lequel vous souhaitez enregistrer le projet désarchivé. Confirmez votre sélection avec "OK". (® Répertoire cible ® OK)

7.2 Création d'une nouvelle table des variables

B Dans la vue du projet, naviguez jusqu'aux
 PLC tags (Variables API) de votre automate et créez une nouvelle table des variables en double-cliquant sur
 Add new tag table (Ajouter nouvelle table des variables).



Renommez la table de variables que vous venez de créer en "Tag table_sorting station" (Table de variables_installation de tri). (R Clic droit sur "Tag table_1" (Table des variables_1)
Renommer" R Tag table_sorting station (Table de variables_installation de tri)



Ensuite, ouvrez-la en double-cliquant dessus. (
 Tag table_sorting station (Table des variables_installation de tri))

Siemens - C:\Users\mde\Documents\Auto	omatis	ierung\011-101	1_CPU1214C\011-10	01_CPU1214C			_	_		_		-
oject Edit View Insert Online Optic P 🎦 🔒 Save project 昌 💥 🗐 🗊 🗙	ns T	C ^{al} ± 🗄 🛄	Help	online 🖉 Go offline	år 🖪 🖬 🗶	3	earch in	project>	9a	1	Fotally Integrated Autor	PORTA
Project tree		011-101_CPL	U1214C → CPU_12	214C [CPU 1214C D	C/DC/DC] > PLC t	ags 🕨 Tag	table_s	orting sta	ation [0] 🛛 🗕 🗖	■× Ta	isks	
Devices							-	🖬 Tags	User constan	nts O	ptions	
194 E		🥩 🥐 🕞 I	🛨 😤 🖬									
		Tag table	_sorting station							~	Find and replace	
 011-101_CPU1214C 	^	Nam	e	Data type	Address	Retain A	Acces	Writa Vi	sibl Comment			
Add new device		1 <	Add new>				V		Image: A start and a start	F	Find:	
Devices & networks												-
CPU_1214C [CPU 1214C DC/DC/DC]											Whole words only	
Device configuration	-											
😧 Online & diagnostics										E	Match case	
Program blocks										E	Find in substructures	
Technology objects										E	Find in hidden texts	
External source files	- 11										Use wildcards	
PLC tags	- 11											
Show all tags	- 11											
Add new tag table	- 11									(Down	
Default tag table [29]	- 11									0) Up	
BIC data timer										- C	Eind	
Watch and force tabler	- 11										Fillu	
Online backups	- 11										Replace with:	
Traces	~	1										
Details view	1000				lid n		1				S	1000
Details view	-				S Prope	erties [3	Info	U C Dia	gnostics		Whole document	
	_	General								() From current position	
		Tag	[Tee						^	Selection	
Name Data type	D			Tay							Replace Replace	all
	-			General								
			-		Name:							
		l			Data type:							
					Address:							
c	>	l				Retained					Languages & resource	s
•		1										-

7.3 Création de nouvelles variables dans une table de variables

® Ajoutez le nom Q1 et confirmez la saisie avec la touche Entrée. Si vous n'avez pas encore créé de nouvelles variables, TIA Portal attribue automatiquement "Bool" comme type de données et l'adresse %E0.0 (I 0.0). (® <Add new> (Ajouter) ® Q1 ® Enter)

						🕣 Tags	s 🔳 l	Jser constants
🖗 🧉	🖹 🕾 😤 🛍							
Tag ta	able_sorting_station							
	Name	Data type	Address	Retain	Acces	Writa	Visibl	Comment
10-10	01	Bool	%10.0					
			and the second sec					

® Modifiez l'adresse en %A0.0 (Q0.0) en saisissant directement cette valeur ou en cliquant sur la flèche de déroulement pour ouvrir le menu d'adressage. Modifiez le type d'opérande à A et confirmez avec Enter ou en cliquant sur la coche. (® %E0.0 ® Operand identifier (Type d'opérande) ® A ®

								🕣 Tag		Jser consta	ints
Ý	ř.	🕈 🕂 🚏 🛍									E
Т	ag ta	ble_sorting_station									
	1	lame	Data type	Address		Retain	Acces	Writa	Visibl	Comment	
1		Q1	Bool	%10.0	-						
2		<add new=""></add>			Opera	nd ident	ifier: 1			-	
					c	perand	type:				
						Add	Pessi M				
						Ditaun	hav 0			_	
						bit nun	iber: 0				

 Entrez le commentaire "conveyor motor M1 forwards fixed speed" (moteur du convoyeur M1 avance à vitesse fixe) pour la variable.

011	-101	_CPU12	14C → CPU	J_121	4C [CPU	1214C	DC/DC/	DC] 🕨 F	PLC tags	▶ Tag	table_sorting sta	ation [1]	_∎≡×
											🕣 Tags	🗉 User c	onstants
*	1. A		°° ii									1.50	
1	ag t	able_so	rting statio	n									
		Name	Data type		Address		Retain	Acces	Writa	Visibl	Comment		
1		Q1	Bool		%Q0.0						conveyor motor -M	11 forwards fixe	ed speed
2		<add i<="" td=""><td></td><td></td><td></td><td></td><td></td><td></td><td>v</td><td> Image: A start of the start of</td><td></td><th></th><th></th></add>							v	 Image: A start of the start of			

(® <Add new> (Ajouter) ® Q2 ® Enter ® Comment (Commentaire) ® conveyor motor M1 backwards fixed speed (moteur du convoyeur M1 en sens inverse à vitesse fixe))

									🕣 Tags	User constants
1	The second secon	•••	r 🗊							a
	Tag t	table_sor	ting station							
		Name 🔺	Data type	Address	Retain	Acces	Writa	Visible in	Comment	
1	-	-Q1	Bool	%Q0.0					conveyor motor -M1 f	forwards fixed speed
2	-	-Q2	Bool	%Q0.1					conveyor motor -M1 b	backwards fixed speed

7.4 Importation de la "Table_des_variables_installation_de_tri"

Pour insérer une table des mnémoniques déjà existante, cliquez avec le bouton droit de la souris sur un champ vide de la "Tag_table_sorting_station" (table_des_variables_ installation_de_tri) créée. Sélectionnez "Import file" (Fichier d'importation) dans le menu contextuel.

(® Clic droit dans un champ vide de la table des variables ® Import file (Fichier d'importation))

11	-101	_CPU1214C	CPU_121	4C [CPU 1214C	DC/E	DC/D	C] 🕨 PL	C tags	Tag t	able_sorting_st	ation [2] 🛛 🗖 🗮 🗙
										🕣 Tags	User constants
Ý	*		it .								
1	lag t	able_sorting	_station								
-		Name	Data type	Address	Ret	ain	Acces	Writa	Visibl	Comment	
1	-	Q1	Bool	%10.0	E					conveyor motor -	M1 forwards fixed speed
2	-	Q2	Bool	%10.1						conveyor motor -	11 backwards fixed speed
3		<add news<="" td=""><td>🚔 Insert row 🚔 Add row</td><td></td><td></td><td></td><td></td><td>V</td><td>V</td><td></td><td></td></add>	🚔 Insert row 🚔 Add row					V	V		
			X Cut			Ctrl+)	K.				
			Сору			Ctrl+0	2				
			Paste			Ctrl+\	1				
			X Delete			De	1				
			Rename			F	2				
			X Cross-refe	rences		F11	1				
-			X Cross-refe	rence information	Shi	ft+F11	1				
			Monitor al	I							
			Import file	10							
			Export file								
	<		Q Properties				100				>

 Choisissez la table des mnémoniques voulue (par ex. au format xlsx), puis confirmez votre choix à l'aide du bouton "Ouvrir".

(
 SCE_FR_020-100_Table_des_variables_installation_de_tri...
 Open (Ouvrir))

® Une fois l'importation terminée, une fenêtre de confirmation s'ouvre pour vous donner la possibilité de consulter le fichier journal de l'importation. Cliquez sur ® OK.

Import co	ompleted. (0032:000001) X
	Import completed successfully.
	Detailed information is shown in the import log file.
	Click here to view the log file.
	ОК

- ® Vous constaterez que certaines adresses sont affichées en orange. Celles-ci existent à double et les noms des variables associées ont été numérotés automatiquement afin d'en garantir l'univocité.
- ® Pour supprimer les doublons, sélectionnez les lignes et appuyez sur la touche Suppr de votre clavier ou sélectionnez "Delete" (Supprimer) dans le menu contextuel.

									Tags 🔲 User constants
100	1		😤 🛍						E
1	lag t	able_so	orting station						
		Name	Data type	Address	Retain	Acces	Writa	Visibl	Comment
		Q1	Bool	%Q0.0					conveyor motor -M1 forwards fixed speed
		Q2	Bool	%Q0.1					conveyor motor -M1 forwards fixed speed
		-A1	Bool	%10.0					return signal emergency stop ok (nc)
	-	-K0	Bool	%10.1					main switch "ON" (no)
		-50	Bool	%10.2					mode selector manual(0) / automatic(1)
		-51	Bool	%10.3					pushbutton automatic start (no)
	-00	-52	Bool	%10.4					pushbutton automatic stop (nc)
	-0	-B1	Bool	%10.5					sensor cylinder -M4 retracted (no)
		-B2	Bool	%10.6					sensor cylinder -M4 extended (nc)
1		-B3	Bool	%10.7					sensor motor -M1 actice (pulse signal for .
		-B4	Bool	%11.0					sensor part at slide (no)
	-	-85	Bool	%11.1					sensor metal part (no)
5)).		-B6	Bool	%11.2					sensor part in front of cylinder -M4 (no)
		-B7	Bool	%11.3					sensor part at end of conveyor (no)
		-53	Bool	%11.4					pushbutton manual mode conveyor -M1
	-0	-54	Bool	%11.5					pushbutton manual mode conveyor -M1
		-55	Bool	%11.6					pushbutton manual mode cylinder -M4 re.
		-56	Bool	%11.7					pushbutton manual mode cylinder -M4 ex
	-	-Q1	Bool	%Q0.0					conveyor motor -M1 forwards fixed speed
	-	-Q2	Bool	%Q0.1					conveyor motor -M1 backwards fixed spee
	-	-Q3	Bool	%Q0.2					conveyor motor -M1 variable speed
		-M2	Bool	%Q0.3					cylinder -M4 retract
	-	-M3	Bool	%Q0.4					cylinder -M4 extend
	-	-P1	Bool	%Q0.5					display "main switch on"
	-	-P2	Bool	%Q0.6					display "manual mode"
;	-00	-P3	Bool	%Q0.7					display "automatic mode"

(
 Clic droit sur variables sélectionnées
 Delete (Supprimer))

® Maintenant, vous avez devant vous une table des mnémoniques complète des entrées et sorties digitales. Enregistrez votre projet sous 031-100_FC-Programmierung.

(
 Project (Projet)
 Save as (Enregistrer sous))
 031-100_Programmation de FC
 Save (Enregistrer))

Siemens - C:\Users\m	de\Documents\Aut	omatis	sierun Tools	ng\011-10 Window	I_CPU1214CW	11-101_CPU121	4C						
😚 🎦 🔚 Save project	🛓 🗶 🗎 🗎 🗙	<u>ات</u>	(a)		li 🖳 📮 💋	🍠 Go online 🖉	Go offline	I I	×E		Search in project>	Totally In	tegrated Automation PORT
Project tree			011	-101_CPI	J1214C ► CP	U_1214C [CPU	1214C DC/DC	DC] 🕨	PLC tag	s 🕨 Tag	g table_sorting station [28]		_ 7 =
Devices												🕣 Tags	User constants
P3	1	1	1	🥐 🕞 I	+ 😤 🖬								E
				Tag table	sorting statio	on							
- 011-101_CPU121	4C	^		Nam	e Data type	Address	Retain	Acces	Writa	Visibl	Comment		
Add new devic	ce		1	- T	A1 Bool	11 %IO.0					return signal emergency stop ok (nc)		
h Devices & net	works		2	- I	0 Bool	%0.1					main switch _ON" (no)		
- CPU_1214C [C	PU 1214C DC/DC/DC]		3	a 4	0 Bool	%10.2					mode selector manual(0) / automatic(1)		
Device con	figuration	-	4	a 3	1 Bool	%10.3					pushbutton automatic start (no)		
V. Online & d	iagnostics		5	a 3	2 Bool	%10.4					pushbutton automatic stop (nc)		
+ 🛃 Program bl	locks		6	-	B1 Bool	%10.5					sensor cylinder -M4 retracted (no)		
Fight Technology	yobjects		7	-	Bool	%10.6					sensor cylinder -M4 extended (nc)		
External sc	ource files		8		Bool	%10.7					sensor motor -M1 actice (pulse signal for		
🕶 🚂 PLC tags			9	-	4 Bool	%11.0					sensor part at slide (no)		
Show a	ll tags		10		5 Bool	%11.1					sensor metal part (no)		
Add ner	w tag table		11	-	6 Bool	%11.2					sensor part in front of cylinder -M4 (no)		
😤 Default	tag table [29]		12		Bool	%11.3					sensor part at end of conveyor (no)		
💺 Tag tab	le_sorting station [28		13	-	3 Bool	%11.4					pushbutton manual mode conveyor -M1		
PLC data ty	/pes		14		4 Bool	%11.5					pushbutton manual mode conveyor -M1		
Natch and	force tables		15	•	5 Bool	%11.6					pushbutton manual mode cylinder -M4 re		
Online bac	kups		16	•	6 Bool	%11.7					pushbutton manual mode cylinder -M4 ex		
🕨 🔄 Traces		~	17	-	01 Bool	%Q0.0					conveyor motor -M1 forwards fixed speed		
Details view			18		2 Bool	%Q0.1					conveyor motor -M1 backwards fixed speed		
betans nen			19	-	3 Bool	%Q0.2					conveyor motor -M1 variable speed		
			20		M2 Bool	%O0.3					cylinder -M4 retract		
			21	- 10	VB Bool	%Q0.4					cylinder -M4 extend		
Name	Data type		22		1 Bool	%Q0.5					display main switch on"		
-Q1	Bool	91~	23		2 Bool	%Q0.6					display_manual mode"		
-Q2	Bool	9	24		Bool	%Q0.7					display_automatic mode"		
-Q3	Bool	%	25		4 Bool	%Q1.0					display_emergency stop activated"		
-S0	Bool	%≡	26	- II	5 Bool	%Q1.1					display automatic mode started"		
41 -S1	Bool	91	27	-	6 Bool	%Q1.2					display cylinder -M4 "retracted"		
-52	Bool	91~											
< 11		>									Properties	Info 追 🗓 Di	iagnostics
A Portal view	Cveniew	1	Tag	table so									CRUIDIAC une e

7.5 Création de la fonction FC1 "MOTOR_HAND" pour le moteur du convoyeur en marche par impulsion

Pour créer une nouvelle fonction, cliquez dans la vue du portail dans la section PLC programming (Programmation API) sur "Add new block" (Ajouter nouveau bloc).

(® PLC programming (Programmation API) ® Add new block (Ajouter nouveau bloc) ®



TIA Siemens - C:\Users\mde	Documents	Watomation 1031-100_FC_Programming 1031-1	00_FC_Programming	_ D X
		Device CPU_1214C	Add new block	PORTAL
Devices & networks	*	Show all objects Add new block	Name: Block_1	
PLC programming			Language: FBD Number: 1	
Motion & technology Drive	*		Organization Menuel	
		 Show cross-references Show program structure 	Description: Function block Function block	cated memory.
Online & Diagnostics	10		-	
			Function	
		e Help	Data block	
			> Additional information	
			Add new and open	Add
Project view		Opened project: C:\Users\mde\Do	uments\Automation\031-100_FC_Programming\031-100_FC_Programmi	ng

Renommez votre nouveau bloc en : "MOTOR_HAND", vérifiez que LOG est choisi comme langage et activez la numérotation automatique. Cochez "Add new and open" (Ajouter nouveau et ouvrir) pour que le bloc fonctionnel que vous avez créé s'ouvre automatiquement dans la vue du projet.Cliquez sur "Add" (Ajouter).

([®] Name (Nom) : MOTOR_HAND[®] Language (Langage) : LOG [®] Number (Numéro) : automatique [®] ^I Add new and open (Ajouter nouveau et ouvrir) [®] Add (Ajouter))

Add new block			
Name: MOTOR_MANUAL			
Organization block	Language: Number:	FBD	
Function block	Description: Functions are o	code blocks or subroutines without dedicated memor	у.
FC Function			
Data block	More		
> Additional inform	nation		
Add new and open			Add

7.6 Définition de l'interface de la fonction FC1 "MOTOR_HAND"

Lorsque vous avez cliquez sur "Add new and open" (Ajouter nouveau et ouvrir), la vue du projet s'ouvre avec une fenêtre pour vous permettre de créer le bloc que vous venez de générer.

 La déclaration de l'interface de votre fonction se trouve dans la partie supérieure de votre vue de programmation.

юÏ	ы	K I	ê 🔮 🔍 🖿 🚍 🚍] 📲 ± 📲 ± 📲 ±	🖃 🎲 🍋 <table-cell></table-cell>	🖑 🖓 🗣 🖢 👘 🖓 🖓 🖓	
	MC	OTC	DR_MANUAL				
		Na	me	Data type	Default value	Comment	
1	-	•	Input				
2			<add new=""></add>				
3		•	Output				
4			<add new=""></add>				
5		•	InOut				
6			<add new=""></add>				
7	-00	٠	Temp				
8			<add new=""></add>				
9	-	•	Constant				
10			<add new=""></add>				
11		•	Return				
12	-0		MOTOR_MANUAL	Void			
	<				111		>
_	-						
8		> = 1	· ???•	=]			
•	RIO	ck	title:				
-	om	me	ot				
	.0111	1110	n.				=
•	1	Net	twork 1:				
	C	om	ment				
							- 11
							~
						100%	

I a commande du moteur du convoyeur requiert un signal de sortie binaire. C'est pourquoi nous créons d'abord la variable de sortie #Convoyeur_moteur_marche par impulsion de type "Bool". Nous faisons accompagner ce paramètre du commentaire "Commander le moteur du convoyeur en marche par à-coups". (® Output (Sortie) : Conveyor_motor_manual_mode (Convoyeur_moteur_marche par impulsion) ® Bool ® Control of the conveyor motor in manual mode (Commander le moteur du convoyeur en marche par impulsion)

5	C_F	ro	gramming + CPU_1214C [CPU 1	214C DC/DC/DC] >	Program block	us → Motor_Manual[FC1] 🛛 🗖 🗮 🗙
1.33		5 =	∮ ≉ ⊾ ⊨ ⊟ ≡ ⊡ 2 ±	2 ± 12 ± 🗆 😭	ල (2) (2) (2) (2)	19 G 由自然にが 11、 13
T.a	MC	ото	DR_MANUAL			
-		Na	me	Data type	Default value	Comment
1	-	•	Input			
2			<add new=""></add>			
З	-	•	Output			
4	-		Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode
5	ſ		<add new=""></add>			
6	-	•	InOut			
7			<add new=""></add>			
8	-	•	Temp			
9			<add new=""></add>			
10	-00	•	Constant			
11			<add new=""></add>			
12	-	•	Return			
13			MOTOR_MANUAL	Void		

- ® Sous Input ajoutez d'abord le paramètre #Manuel_activé comme interface d'entrée et confirmez la saisie avec la touche Entrée ou quittez le champ de saisie. Le type de données "Bool" est attribué automatiquement. Il est conservé. Saisissez ensuite le commentaire "Manual mode activated" (mode de fonctionnement manuel activé).
 - (
 Manual_mode_active (Manuel_activé)
 Enter
 Bool
 Manual mode activated (Mode de fonctionnement manuel activé)

® Sous Input ajoutez ensuite comme autres paramètres d'entrée binaires #Bouton_marche par impulsion, #Enable_OK (Validation) et #Safety_shutoff_active (arrêt automatique de sécurité_activé) et vérifiez leurs types de données. Complétez avec des commentaires utiles.

		Na	ime	Data type	Defau	Comment	
1	-	•	Input				1
2			Manual_mode_active	Bool		Manual mode activated	
3	-		Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on	
4	-		Enable_OK	Bool		All enable conditions OK	
5	-		Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated	
6			<add new=""></add>				
7		•	Output				
8	-		Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode	
9			<add new=""></add>				
10	-	•	InOut				
11			<add new=""></add>				
12	-	•	Temp				
13			<add new=""></add>				
14	-	•	Constant				
15			<add new=""></add>				
16	-	-	Return				5

Pour la documentation du programme, saisissez le titre du bloc, un commentaire de bloc et pour le réseau 1, un titre du réseau évocateur.

(
 Block title (Titre du bloc) : Motor control in manual mode (commande du moteur en mode manuel)
 R Network 1 (Réseau 1) : Control of the conveyor motor in manual mode (Commander le moteur du convoyeur en marche par impulsion)

		Na	me	Data type	Default value	Comment	
1	-	-	Input				
2			Manual_mode_active	Bool		Manual mode activated	
3			Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on	
4	-		Enable_OK	Bool		All enable conditions OK	
5	-		Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated	
6	-	•	Output				
7	-		Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode	
	<				1111		>
8		>=1	1??} → -01 ↦ -[=]				Г
•	Blo	ck	title: Motor control in manual mode				1
	Conv cond Conv	eyo itio eyo	r motor in manual mode: If the pushb ns are granted and the safety shutoff i r_motor_manual_mode is activated	utton_manu s not activat	al_mode is ope ed the output	rated, the enable	
•	1	let	work 1: Control of the conveyor mo	tor in manua	al mode		1

7.7 Programmation du FC1 : MOTOR_HAND

® Sous la déclaration de l'interface, vous verrez dans la fenêtre de programmation une barre d'outils contenant différentes fonctions logiques et, en dessous, une zone avec des réseaux. Nous y avons déjà défini le titre du bloc et un titre pour le premier réseau. Dans les réseaux, la programmation s'effectue en utilisant différents blocs logiques. Pour des raisons de lisibilité, le programme est subdivisé en plusieurs réseaux. Vous allez maintenant vous familiariser avec les différentes méthodes qu'il existe pour insérer des blocs logiques.



® Dans votre fenêtre de programmation, à droite, se trouve une liste des instructions que vous pouvez utiliser dans le programme. Recherchez sous ® Instructions de base ® Fonctions logiques, la fonction –[=] (Affectation) et placez-la dans votre Réseau 1 grâce à un glisser-déposer (la ligne verte apparaît, pointeur de la souris avec le +).

(
 Instructions (Instructions)
 Basic instructions (Instructions de base)
 Bit logic operations (Fonctions logiques)
 –[=])

C_	_Pro	grammir	ng 🕨	CPU_	_1214C [C	PU 12	14C DC/DC	IDC] •	Progra	m blocks	MOTOR	_MANUA	IL [FC1]			Instr	uction	S		1		
																Optio	ons					
й.	X 3	i <⊵ ف	L, 1	= 🚍		1 ± - 3	2 ± 12 ± [= 😰	CO 60	a 🕫 🕹	ς ≡ 1 _≡ 1	= 61 0	0,00						inil	•		1
N	юто	OR_MAN	UAL													▼ F	avorite	s	_	_		
	Na	ame				1	Data type		Defau	Comment											1	
Y	•	Input													~	8	>=1	??	-	-01	-→	
*		Man	ual_m	ode_a	ctive		Bool			Manualmo	de activate	1										
4		Push	buttor	n_man	nual_mode		Bool			Pushbutto	n manual mo	de conve	yor on		-	-(=)						
Y		Enab	le_OK				Bool			All enable	conditions C	к										
4		Safe	ty_shu	toff_a	ctive		Bool]	Safety shu	offactive e.	. emerge	ncy stop op	erated								
		<add< td=""><td>new></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>▼ Bi</td><td>asic in</td><td>struct</td><td>ions</td><td></td><td></td><td></td></add<>	new>													▼ Bi	asic in	struct	ions			
Y	•	Output														Name					Des	
Y		Conv	eyor_r	motor	_manual_m	ode	Bool			Control of	he conveyor	motor in	manual mo	de	~	• 🗖	Genera	al			1	1
1	٢							1	II						>		Bit logi	ic opera	ations			
	1			- 1				Tour Cont									E &			9	AND	
	>=	1 ??	-	-01	→ -{=]												E >=1			1	OR log ≣	-
R	ock	title: M	lator c	ontrol	in manual r	node											E ×			1	EXCLU	
0	UCK	or motor i		ualm	ode: If the n	uchhut	ton manual	modei	r operate	d the ensh							E -[=]]			Assigi	
01	ivey	ons are gr	anted	and th	he safety shi	toffis	not activated	the out	tput	cu, the endu	ic.						E -[/=	=]			Negat	
0	nvey	or_motor_	manu	al_mo	ode is activa	ted											E -[R]	1		8	Reset	et
																	-[S]]			Setou	
	Net	twork 1:	Con	trol of	the conveyo	or moto	or in manual	mode									SET SET	BF			Set bi	
	Com	nment															E RES	ET_BF		9	Reset	
																	SR SR			2	Setire	
														-			E RS			100	Reset	
																	- P	-		3	Scan	
																	1 - N	-		1	Scan	
																	E -[P]	-		1	Setor	ý
																<		III			>	

- ® Maintenant, faites glisser le paramètre Output #Convoyeur_moteur_marche par impulsion et déposez-le sur <??.?> au-dessus du bloc que vous venez de créer. La meilleure méthode pour sélectionner un paramètre dans la description de l'interface consiste à le prendre par l'icône bleue
 - (
 Conveyor_motor_manual_mode (Convoyeur_moteur_marche par impulsion))

0		lic	Data type	Defau	Comment			
-0	•	Input						
		Manual_mode_active	Bool		Manual mode activated			
-	•	Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on			
-		Enable_OK	Bool		All enable conditions OK			
	•	Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated			
	•	<add new=""></add>						
	•	Output						
	•	Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode			
<	-			III				
Conv cond Conv	eyor ition eyor	motor in manual mode: If the pushb is are granted and the safety shutoff i _motor_manual_mode is activated	utton_manual_mode is not activated the ou	is operate tput	d, the enable			
S A	Vetv	work 1: Control of the conveyor mo	tor in manual mode					
•								

® Cela permet de définir que c'est ce bloc qui écrit le paramètre #Convoyeur_moteur_marche par impulsion. Toutefois, il manque encore les conditions d'entrée nécessaires pour le réaliser effectivement. Pour cela, placez le paramètre Input #Manuel_activé grâce à un glisser-déposer sur "…" à gauche du bloc d'affectation.

(
 Manual_mode_active (Manuel_activé))

MO	TO	R_MANUAL				
	Nar	ne	Data type	Defau	Comment	
-	•	Input				
-		Manual_mode_active	Bool		Manual mode activated	
		Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on	
-		Enable_OK	Bool		All enable conditions OK	
-		Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated	
		<add new=""></add>				
-	-	Output				Constantion of the local division of the loc
-		Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode	
1				1		1
Bloc	=1	Image: The second s	utton manual mode i	s operate	ed, the enable	
Bloc	= 1 eyoi tior	Image: The second s	utton_manual_mode is s not activated the out	s operate	ed, the enable	
> Bloc onve ondi onve	= 1 Eyon tion Eyon	Image: The second s	utton_manual_mode i is not activated the out stor in manual mode	s operate	ed, the enable	
Bloc onve ondi onve	e son tior eyon tior eyon letr	Image: The second s	utton_manual_mode i is not activated the out itor in manual mode	s operate	d, the enable	
> Bloc onve ondi onve N Co	eyon tior eyon letv	Image: The second s	utton_manual_mode i is not activated the out stor in manual mode	s operate	ed, the enable	
> Bloc onve ondi onve	eyon tior eyon	Image: The second s	utton_manual_mode i is not activated the out	s operate	ed, the enable	

® Sur l'entrée du bloc d'affectation, nous voulons combiner des paramètres supplémentaires par opération ET. Pour cela, cliquez d'abord sur l'entrée du bloc où #Manuel_activé est déjà interconnecté pour que la ligne d'entrée soit représentée sur fond bleu.



R Cliquez sur l'icône dans la barre d'outils pour insérer une opération ET entre la variable
 #Manuel_activé et votre bloc d'affectation.

å	>=1 [??] 0
- 6	AND logic operation [Shift+F2]
Co Co Co	nveyor motor in manual mode: If the pushbutton_manual_mode is operated, the enable nditions are granted and the safety shutoff is not activated the output nveyor_motor_manual_mode is activated
•	Network 1: Control of the conveyor motor in manual mode
_	Comment
	#Conveyor_ motor_manual_ mode
	#Manual_mode_ active

® Double-cliquez sur la deuxième entrée de l'opération logique l'entrée & <??.?>, un champ s'ouvre, saisissez la lettre "T" dans ce champ pour voir la liste des variables qui commencent par "T". Cliquez sur la variable #Bouton_mode_manuel à vue et validez avec ® Enter (touche Entrée).

(
 Bloc &
 e <??.?>
 e T
 e #Pushbutton_manual_mode (Bouton_mode_manuel à vue)
 e Enter)

	Nan	ne	Data type	Defau	Comment	
-	•	Input		1		
-	•	Manual_mode_active	Bool		Manual mode activated	
	•	Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on	
-	•	Enable_OK	Bool		All enable conditions OK	
-	•	Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated	
	•	<add new=""></add>				
	•	Output				
	•	Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode	ļ
<			1000			l
ondi	tion	is are granted and the safety shutoff i	s not activated the ou	tput		
ondi	tion	ns are granted and the satety shutoff in r_motor_manual_mode is activated	s not activated the ou	tput		
ondi onve	tior eyor	ns are granted and the satety shutom r_motor_manual_mode is activated work 1: Control of the conveyor mo ment	s not activated the ou tor in manual mode	tput		

Remarque : cette méthode d'affectation des variables comporte un risque de confusion avec les variables globales de la table des variables. C'est pourquoi il convient de privilégier la méthode d'affectation par glisser-déposer de la déclaration de l'interface décrite ci-dessus.

® Pour ne forcer la sortie que lorsque la validation a été attribuée et le dispositif d'arrêt automatique de sécurité n'est pas activé, vous devez en plus combiner les variables d'entrée #Enable_OK (Validation) et #Safety_shutoff_active (Arrêt automatique de sécurité_activé) par opération ET. Pour ce faire, double-cliquez sur l'étoile jaune se de votre élément logique ET pour insérer deux nouvelles entrées.

8	> = 1	??	۲	-01	↦	-[=]											
• 6	Block t	itle:															
▼ Co co Co	onveyor ondition onveyor	motor s are g _motor	in ma ranteo _man	nual m d and t ual_m	node: I the saf ode is	f the p fety sh activa	oushi nutoff ated	is not	n_manu tactiva	ual_ ted	the	de i ou	s op tput	erat	ed, ti	he en	able
•	Netv	vork 1	: Co	ntrol of	f the co	onvey	or mo	otor in	manu	aln	nod	e					
	Comn	nent															
	#Mar #F ma	nual_m a Pushbu nual_r	ode_ ctive tton_ node	-***	&		ı	#Co motor r	nveyor _manu node =	al_							

® Ajoutez sur les nouvelles entrées de l'élément logique ET les variables d'entrée #Enable_OK (Validation) et #Safety_shutoff_active (Arrêt automatique de sécurité_activé).

	&	
#Manual_mode_ active —		
#Pushbutton_ manual_mode —		#Conveyor_ motor_manual_
#Enable_OK —		mode
#Safety_shutoff_ active	*	

® Pour programmer une négation de l'entrée connectée au paramètre #Safety_shutoff_active



(Arrêt automatique de sécurité_activé), sélectionnez-la et cliquez sur

 N'oubliez pas de cliquer sur Save project à la fin de chaque étape. Ci-après, la fonction "MOTOR_HAND [FC1] terminée, représentée dans le logigramme (LOG).



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

 Language (Langage) : CONT)

MOTOR_MANUAL [FC1]		🔍 Properti	es 🚺	Info 追	Diagnostics	
General						
General Information	General					
Time stamps Compilation Protection		Name:	MOTOR_MA	NUAL		
Attributes		Language:	FBD LAD	N		•
		Number:	FBD manual	~		
		(automa 	tic		

® Ci-dessous, le programme représenté dans CONT.

	Nan	ne	Data type	Defau	Comment	
-	•	Input				
		Manual_mode_active	Bool		Manual mode activated	
		Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on	
-		Enable_OK	Bool		All enable conditions OK	
-		Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operate	d
		<add new=""></add>				
-	•	Output				
-		Conveyor_motor_manual_mode	Bool		Control of the conveyor motor in manual mode	
<				III		
Blo	t/⊢ ckt eyor	→ 1221 → - itle: Motor control in manual mode motor in manual mode: If the pushb	utton_manual_m	ode is operate	d, the enable	
Blow	k t eyor ition	→ 1221 → - itle: Motor control in manual mode motor in manual mode: If the pushb is are granted and the safety shutoff i _motor_manual_mode is activated	utton_manual_m s not activated th	ode is operate se output	d, the enable	
Blow ondi onv	k t eyor ition eyor	the state of the pushes are granted and the safety shutoff is motor_manual_mode is activated work 1: Control of the conveyor mo	utton_manual_m s not activated th tor in manual mo	ode is operate e output	d, the enable	
Bloc onvi ondi onvi onvi	k t eyor tion eyor	→ 1221 → - itle: Motor control in manual mode is are granted and the safety shutoff i motor_manual_mode is activated work 1: Control of the conveyor mo nent	utton_manual_m s not activated th tor in manual mo	ode is operate e output ode	d, the enable	

7.8 Programmation du bloc d'organisation OB1 – commande de l'avance du convoyeur en mode manuel

 R Avant la programmation des bloc d'organisation "Main [OB1]", nous commutons le langage de programmation sur LOG (logigramme). Pour cela, cliquez d'abord avec le bouton gauche de la souris sur "Main [OB1)" dans le dossier "Blocs de programme".

(® CPU_1214C[CPU 1214C DC/DC/DC ® Program blocks (Blocs de programme) ® Main [OB1] ® Switch programming language (Commuter le langage de programmation) ® FBD (LOG))



® Double-cliquez maintenant sur le bloc d'organisation "Main [OB1]" pour l'ouvrir.



(® Network 1 (Réseau 1):... ® Control conveyor motor forwards in manual mode (Commande de l'avance du convoyeur en mode manuel/marche par à-coups))

03	1-1	00	_FC_Pi	rogra	mmin	g 🕨	CPU_	1214C [CPU 1	214C DC/DC/DC] 🕨	Program blocks > Main [OB1]		×
ю	ы	٤ 🗉	¢² ≣?	Ū.,	E	3 🗃	1	🛿 ± 🖀 ± 😫 :	• 🖃 😰 🧐 😡 e	≣ 🥶 🍫 💶 '₌ '₌ 📢 📣 🖤 🔒		
	Ma	in										
		Na	me					Data type	Default value	Comment		
1	-	•	Input									~
2			Init	ial_Ca	all			Bool		Initial call of this OB		
3	-		Ren	nane	nce			Bool		=True, if remanent data are available		=
4		•	Temp									
5			<ac< td=""><td>ld nev</td><td>N></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ac<>	ld nev	N>							
6		•	Consta	nt								
7	1	•	<ac< td=""><td>ld nev</td><td>N></td><td></td><td></td><td></td><td></td><td></td><td></td><td>~</td></ac<>	ld nev	N>							~
	•										1	
8		> = 1	??	н	-01	↦	-[=]					
•	Blo	ck	title:	"Main	Progra	m Sw	eep (C	Cycle)*			[^
C	om	mei	nt									
•		Vet	work 1	· 10	ontrol	onvey	ormo	tor forwards in m	anual mode		_	
			MOIN I	• [ondore	.onrej	0				_	
	C	.om	ment									
												=

Placez maintenant votre fonction "MOTOR_HAND [FC1]" dans le réseau 1 sur la ligne verte grâce à un glisser-déposer.



In bloc avec l'interface que vous avez définie et les connexions EN et ENO est inséré dans le réseau 1.



® Pour insérer un ET devant le paramètre d'entrée "Enable_OK" (Validation_OK), sélectionnez

-[=] 2 > = 1 [??] -01 -AND logic operation [Shift+F2] ep (Cycle)* -Comment Network 1: Control conveyor motor forwards in manual mode Comment %FC1 "MOTOR_MANUAL" EN Manual_ <??.?>mode_active Pushbutton manual_ Conveyor_ <??.?> --- mode motor_ <??.?> 🚍 Enable_OK manual_ mode --<??.?> Safety_ <??.?> --- shutoff_active ENO -

cette entrée et ajoutez l'opération ET en cliquant sur l'icône de la barre d'outils. (®

® Cliquez sur l'étoile jaune ³ de l'élément logique ET pour ajourer une entrée supplémentaire.



- Pour connecter le bloc avec les variables globales de la "table_des_variables_installation_de_tri", il existe 2 méthodes :
- R Vous
 sélectionner dans navigateur du projet pouvez le la "table_des_variables_installation_de_tri" et faire glisser la variable globale souhaitée de la vue détaillée sur l'interface du FC1 R Tag table sorting station (-S0 ® Manual_mode_active (Manuel_activé))



® Ou saisir pour <???.> les caractères de début (par ex. : "-S") de la variable globale souhaitée et sélectionner dans la liste qui apparaît la variable d'entrée globale "-S0" (%E0.2). (® Manual_mode_active (Manuel_activé) ® -S ® -S0)

ł	>=1	??	٦	-01	↦	-[=]				
BI	ock ti	tle: "	Main F	rogram	n Swe	ep (Cycle)*				
Con	nment									
8	Netv	ork 1:	Con	trol co	nveyo	r motor forwards in m	anual mode			
	Comm	ent								
								9	ÆC1	
								MOIOR	MANUAL	
							— EN			
						-s	Mar	de active		
					&	-S0"	Bool	%10.2	mode selector manual(0) / auto	
		<	27.7>-	-		"-S1"	Bool	%10.3	pushbutton automatic start (no)	
		<	27.7>-	-		"-S2"	Bool	%10.4	pushbutton automatic stop (nc)	1
			27.7>-	- *		- 4 "-S3"	Bool	%11.4	pushbutton manual mode conv	L
						- 54 "	Bool	%11.5	pushbutton manual mode conv	

Insérez les autres variables d'entrée "-S3", "-K0", "-B1", "-S4" et "-A1", puis également la variable de sortie "-Q1" (%A0.0) sur la sortie "Conveyor_motor_manual_mode" (Convoyeur_moteur_marche par à-coups).





7.9 Enregistrer et compiler le programme

Pour enregistrer votre projet, sélectionnez le bouton Seve project dans le menu. Pour compiler tous les blocs, cliquez sur le dossier "Programm blocks" (Blocs de programme) et sélectionnez l'icône Compile (Compiler) dans le menu. (R Seve project R Programm blocks (Blocs de programme) R



	Q Proper	ties	L Info	🔒 🗓 Diag	gnostics		
Compile	Energy Suite	Synt	tax				
-							
0)							
Description				Go to	?	Errors	Wa
				7		0	0
				~		0	0
Block was succes	sfully compiled.			~			
Block was succes	sfully compiled.			~			
Compiling finishe	d (errors: 0; warnings	0)					
	Compile Compile O) Description Block was succes Block was succes Compiling finishe	Compile Energy Suite O) Description Block was successfully compiled. Block was successfully compiled. Compiling finished (errors: 0; warnings	Compile Energy Suite Sym • • 0) Description Block was successfully compiled. Block was successfully compiled. Compiling finished (errors: 0; warnings: 0)	Compile Energy Suite Syntax • • • 0) Description Block was successfully compiled. • Block was successfully compiled. • Compiling finished (errors: 0; warnings: 0) •	Compile Energy Suite Syntax • • • 0) Description Go to Block was successfully compiled. • Block was successfully compiled. • Compiling finished (errors: 0; warnings: 0) •	Compile Energy Suite Syntax • • 0) Description Go to P • • Block was successfully compiled. • Block was successfully compiled. • Compiling finished (errors: 0; warnings: 0) •	Compile Energy Suite Syntax • • • 0) Description Go to ? Block was successfully compiled. • • Block was successfully compiled. • • Compiling finished (errors: 0; warnings: 0) • •

7.10 Charger le programme

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

		X Instructions
roject tree		
Devices		Options
<u>*</u>	📴 🕸 🕸 👘 👘 📰 🔚 🚍 🗩 🗃 🛎 🔚 🔚 👘 🚱 🕪 🕼 👘 🚱 👘 👘 👘 👘 👘 👘 👘	i wit , 🗆 🗉
	Main	✓ Favorites
O31-100_FC_Programming	Name Data type Default value Comment	
Add new device	1 📲 🔻 Input	▲ >=1 1??? -1 -01 →
Devices & networks	2 🔄 = Initial_Call Bool Initial call of this OB	× 1.1
 CPU_1214C [CPU 1214C DC/DC/DC] 		1-1
Device configuration		
Conline & diagnostics	a >=1 127 → −o1 ↦ −[=]	
Gran blocks	Block title: "Main Program Sweep (Cycle)"	^
Add new block	Comment	
		✓ Basic instructions
Technology objects	 Network 1: Control conveyor motor forwards in manual mode 	Name Descri
External source files	Comment	General
PLC tags		Bit logic operations
Show all tags	SEC1	Imer operations
Add new tag table	"MOTOR_MANUAL"	Comparator operations
😤 Default tag table [29]	&EN	+ Math functions
Details view	%0.1 %0.2 Manual_	Move operations
Details view	-sumode_active	Conversion operations
	"B1" Pushbutton_	Program control operati
	141.4 manual_ Conveyor_	Word logic operations
Name Details	*54* -0 * motor	Shift and rotate
Add new block	\$0.0 c.t.t. manual_ x00	
Main OB1	*A1 •• shutoff active ENO ••	
MOTOR_MANUAL FC1		<
		> Extended instructions
	 Network 2: 	> Extended instructions
	Comment	V Flechhology
	< III > 100% •	Communication

7.11 Visualiser des blocs de programme

⊩X 学 学 🔍 🖿 🗄 Main	≝ <u>-</u> 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		• ₩ ₩ ₩ • •	'∃ 41 Cr €°	> wa :
Name	Data type	Default value	Comment		
Input	Rool		Initial call of this OR		
<	5001	10	initial can of dis Ob		11
>=1 1?? -1 -01	·→ -{=]				
lock title: "Main Program	n Sweep (Cycle)*				
mment					
Network 1: Control co	nveyor motor forwards in m	anual mode			
Comment					
			%FC1		
_	-	MOT	DR_MANUAL"		
910 1	&	— EN			
-ко —		*-S0" mode_active			
%10.5		Pushbutton_			
"-B1" —		%1.4 manual_	Conveyor_		
-S4•• *		Enable OK	motor_	%00.0	
		-	mondor_	ind a real	
		%10.0 Safety_	mode -	- "-Q1"	
		"-A1" - o shutoff_active	e ENO		
I-100_FC_Programming	▶ CPU_1214C [CPU 1]	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] +	mode ENO Program blocks	 Main [OB1]	- 01
I-100_FC_Programming KX 글라 글라 🔍 🖿 🗮	▶ CPU_1214C [CPU 1 플 및 월호 월호 월	40.0 Safety_ *A1* → shutoff_active 214C DC/DC/DC] + ± □ 0 (* \$0 (mode e ENO Program blocks →	 Main [081] L≝ 📢 🕼 Р	
I-100_FC_Programming Kă ∰ ∰ ■ E E Main Name	▶ CPU_1214C [CPU 1 :=	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] → ± □ □ □ C ² \$ ₀ \$ Default value	mode e ENO Program blocks ► E 👀 🕸 🗲 I = 1	 Main [OB1] ⊑ ⊊ € € ₽ ₽	
I-100_FC_Programming kX 学 学 🔩 🖿 🖻 Main Name ⓓ ▼ Input ⓓ ■ Initial Call	CPU_1214C [CPU 1 CPU_1214C [CPU 1 Data type Bool	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] + * * ::::::::::::::::::::::::::::::::::::	Program blocks >		
-100_FC_Programming KX 🚔 🚔 🔍 🖿 🖻 Main Name Initial_Call K	 > CPU_1214C [CPU 1 □ 21 호 월 호 월 1 □ Data type Bool 	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] + * * •	Program blocks > Program blocks > Comment Initial call of this OB		
-100_FC_Programming KX = ==================================	CPU_1214C [CPU 1 CPU_1214C [CPU 1 Data type Bool	%40.0 Safety	Program blocks > Program blocks > Program blocks > Comment Initial call of this OB		- • • •
-100_FC_Programming KĂ 🚔 🚔 🔍 🖿 🗮 Vain Name I ritial_Call < >=1 [77]ol	CPU_1214C [CPU 1 Data type Bool H f=1	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] → ± □ □ □ C* ♀₀ ♦ □ Default value	mode e ENO Program blocks → © 🐨 😳 🗲 I = 1 Comment Initial call of this OB	 Main [081] ⊑ ⊊ € € ₽ ₽	
-100_FC_Programming KX ∰ ∰ ∰ ■ E E Vain Name Initial_Call C Network 1: Control co		%40.0 Safety_ *-A1* - • shutoff_active 214C DC/DC/DC] + *	Program blocks >		
-100_FC_Programming	CPU_1214C [CPU 1 Data type Bool det = 1	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] + * * •	Program blocks > Program blocks > Comment Initial call of this OB		
I-100_FC_Programming Main Name Input Initial_Call Network 1: Control cc Comment	CPU_1214C [CPU 1 Data type Bool det =	%40.0 Safety	Program blocks + Program blocks + Program blocks + Comment Initial call of this OB		
I-100_FC_Programming K ﷺ ﷺ ■ E Main Name Initial_Call Comment RUE	CPU_1214C [CPU 1 Deta type Bool deta type deta ty	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] → ± □ □ □ □ C □ □ ± □ □ □ C □ • □ □ C □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ • □ <	Program blocks > Program bloc		
I-100_FC_Programming IA IA IA IA IA IA IA IA IA IA	CPU_1214C [CPU 1 Deta type Bool deta deta	%40.0 Safety	Program blocks > Program blocks > ©		
1-100_FC_Programming KĂ 👾 👾 🔍 📰 🗮 Main Name Initial_Call Comment TRUE %00.1 *-K0° TRUE	CPU_1214C [CPU 1 Data type Bool H H	%40.0 Safety	Program blocks > Program blocks > ©	 Main [081]	
H-100_FC_Programming KX ∰ ∰ ∰ ■ E Main Name	CPU_1214C [CPU 1 Data type Bool H f=1 nveyor motor forwards in m &	%40.0 Safety	Program blocks Program blocks Comment Initial call of this OB		
1-100_FC_Programming KX ﷺ ∰ ♣ E Main Name I intial_Call X >=1 1 1 + -ol Network 1: Control co Comment TRUE %00.5 *-81*	CPU_1214C [CPU 1 Data type Bool H f=1 nveyor motor forwards in m A	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] + * * •<	Program blocks Program blocks Comment Initial call of this OB		
I-100_FC_Programming KX ﷺ ∰ ∰ ■ E Main Name I v Input I v In	CPU_1214C [CPU 1 Data type Bool f=1 nveyor motor forwards in m &	%40.0 Safety_ *-A1* → shutoff_active 214C DC/DC/DC] + * * •<	Program blocks Program blocks Comment Initial call of this OB		
1-100_FC_Programming KX ﷺ ♣ ♣ ♣ Main Name I > Input I initial_Call X >=1 177ol Network 1: Control cc Comment TRUE %0.1 *K0° TRUE %0.5 *B1° FALSE %1.5 *54° - € ↔	CPU_1214C [CPU 1 Data type Bool detain type detaintype detain type detaintype detaintype	%40.0 Safety_ *-A1* - • shutoff_active 214C DC/DC/DC] + * * Default value *	FC1 Conveyor_		
1-100_FC_Programming KX ﷺ ∰ ♣ ♣ Main Name Initial_Call >=1 177 + -ol Network 1: Control cc Comment TRUE %00.1 *K0° TRUE %00.5 *B1° FALSE %1.5 *54° - o ¥	CPU_1214C [CPU 1 Data type Bool definition f=] nveyor motor forwards in m &	%40.0 Safety_ *-A1* - • shutoff_active 214C DC/DC/DC] + * * Default value Default value *	FC1 Conveyor Conveyor Conveyor Manual		
1-100_FC_Programming Main Name I v Input I v Input I v Input I v Input I v Input I v Input V - ol Network 1: Control cc Comment TRUE %0.1 *K0 FALSE %0.5 *B1 *S4* - o	CPU_1214C [CPU 1 Data type Bool deta type deta ty	\$40.0 Safety_ *A1* - shutoff_active 214C DC/DC/DC] * 314 * *S3* mode_active Enable_OK Safety_	Conveyor. Manual Manual Manual Manual Manual Manual	FALSE %Q0.0 *-Q1*	

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

 Après avoir fait un clic droit de la souris, il est possible de sélectionner directement la fonction "MOTOR_HAND" [FC1] appelée dans le bloc d'organisation "Main [OB1]" à "Open and monitor" (Ouvrir et surveiller). (
 "MOTOR_HAND" [FC1]
 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.

Call	environment			
	Dependency structure	1 /	Address	Details
1	💶 Main	1	OB1	@Main > NW1 (Control conveyor motor forwards in manual mot
2				
_	1			
				Transfer to "adjusted manually"

7.12 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 ® 031-100_Programmation de FC.... ® Save (Enregistrer)

IIA Siemens - C:\Users\mde\Documents\Au	utomati	n\031-100_FC_Programming\031-100_FC_Programm	ing	_ ¤ ×
Project Edit View Insert Online Opt	tions 1	ols Window Help		Totally Integrated Automation
3 New) ±	🍽 🛨 🐻 🛄 🕼 🖳 🎇 💋 Go online 🖉 Go offline	🔹 🌆 🌆 🗶 🖃 🛄 -Search in project> 📲	PORTAL
Open Ctrl-	+0	031-100 EC Programming > CPU 1214C [CPU	1214C DC/DC/DCI → Program blocks → Main [OB1] 🖬 🖬 🔪	Instructions
Migrate project				
Close Cth-	+ + + + + + + + + + + + + + + + + + + +			Options
ave Ctrl	+S 📑	ත් ත් 🖻 👻 🛼 📰 🚍 🚍 💬 🗃 ± 🚇 ± 😫	1= 🖂 🕫 🖓 🕼 🖢 🖕 🐂 🐂 🖬 🖓 📽 🐘 📑	init, 🗖 🔜 😫
Save as Ctrl+Shift	(+S	Main		✓ Favorites
Delete project Ctrl	+E ^	Name Data type	Default value Comment	
Archive		1 🕣 🕶 Input		∧ 8 >=1 12?1 → −01 → 0
Retrieve		2 🕣 = Initial_Call Bool	Initial call of this OB	Y 14
Manage multiuser server projects	=	<	ш >	
T Card Reader/USB memory	•			es
The Memory card file	•	a >=1 1??' → -ol ↦ -[=]		lin lin
Start basic integrity check		 Network 1: Control conveyor motor forwards in 	manual mode	~
Upgrade		Comment		
E Print Ctrl	I+P	comment		✓ Basic instructions
Print preview			%FC1	Name Descri
C:\Users\mde\D_\031-100_FC_Programming			"MOTOR_MANUAL"	General General
C:\Users\mde\Docume\011-101_CPU12140	c	&	EN	Timer operations
C:\Users\mde\Documen\011-100_CPU1200	0	%0.1	%a0.2 Manual_	Counter operations
Exit Alt+	-F4	"-ко" —	*-S0" mode_active	Comparator operations
 опше овскорз 	~	%0.5	Pushbutton	Math functions
✓ Details view		"-B1" —	%1.4 manual	Move operations
		%11.5	-53 - mode motor_	Conversion operations
		-54 -0 🛠	Enable_OK manual_ %Q0.0	Program control operati
			%40.0 Safety	Word logic operations
Name			In Shuton_active ENO	Shift and rotate
Add new device				
Devices & networks		 Network 2: 		
Ungrouped devices		Comment		< III >
Common data				> Extended instructions
Documentation settings		e	> 100%	> Technology
Languages & resources		* I		Communication
			Properties Info Diagnostics	
	_	General Cross-references Compile	Energy Suite Syntax	Optional packages
Portal view	-	Main (OB1)	🔜 😪 Co	nnection to CPU_1214C terminated.

7.13 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 MANUEL (-S0 = 0) Activation de l'avance du convoyeur en marche par à-coups (-S3 = 1) Le moteur du convoyeur déclenche l'avance à vitesse fixe (-Q1 = 1)	
4	comme au point 3 mais activation ARRET D'URGENCE (-A1 = 0) B -Q1 = 0	
5	comme au point 3 mais mode AUTO (-S0 = 1) \textcircled{B} -Q1 = 0	
6	comme au point 3 mais mise hors circuit de l'installation (-K0 = 0) \bigcirc -Q1 = 0	
7	comme au point 3 mais tige du vérin pas rentrée (-B1 = 0) ® -Q1 = 0	
8	comme au point 8 mais activer l'inversion du sens de marche du convoyeur en marche par à-coups (-S4 = 1) \circledast -Q1 = 0	
9	Projet archivé correctement	

8 Exercice

8.1 Énoncé du problème – Exercice

Dans cet exercice, nous voulons planifier, programmer et tester les fonctions de processus de l'installation de tri suivante:

 Mode manuel – commande d'inversion du sens de marche du convoyeur en mode manuel/marche par à-coups

8.2 Schéma technologique

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



Figure 10 : Schéma technologique

Schalter der Sortieranlage Switches of sorting station	Automatikbetrieb Automatic mode	Handbetrieb / Manual mode -S3 Tippbetrieb -M1 vorwärts/ Manual -M1 forwards
-P1 einion -P4 aktiviet/active -P4 aktiviet/active -P4 aktiviet/active -P4 aktiviet/active -P2 Handimanual -P3 Auto/auto -P2 Handimanual -P3 Auto/auto	Si start/start	-S4 Tippbetrieb -M1 rückwärts/ Manual -M1 backwards -P7 ausgefahren/extended -S6 Zylinder -M4 ausfahren/ cylinder -M4 extend -S5 Zylinder -M4 einfahren/ cylinder -M4 retract

Figure 11 : 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 0.5	BOOL	-B1	Capteur tige du vérin -M4 rentrée	NO
E 1.4	BOOL	-S3	Bouton-poussoir Avance en marche par à- coups du convoyeur -M1	NO
E 1.5	BOOL	-S4	Bouton-poussoir Recule en marche par à- coups du convoyeur -M1	NO

Cette application requiert les signaux suivants comme opérande.

DA	Туре	Code	Fonction	
A 0.1	BOOL	-Q2	Moteur du convoyeur -M1 en sens inverse à vitesse fixe	

Legende zur Belegungsliste

DE	Entrée TOR	DA	Sortie TOR	

- AE Entrée analogique AA Sortie analogique
 - Entrée A Sortie
- NC Normally Closed (contact à ouverture)
- NO Normally Open (contact à fermeture)

8.4 Planification

Е

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

N⁰	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 MANUEL (-S0 = 0) Activation de l'inversion du sens de marche du convoyeur en marche par impulsion (-S4 = 1) Moteur du convoyeur en sens inverse à vitesse fixe (-Q2 = 1)	
4	comme au point 8 mais activation ARRET D'URGENCE (-A1 = 0) 0 -Q2 = 0	
5	comme au point 8 mais mode AUTO (-S0 = 1) \textcircled{B} -Q2 = 0	
6	comme au point 8 mais mise hors circuit de l'installation (-K0 = 0) \bigcirc -Q2 = 0	
7	comme au point 8 mais tige du vérin pas rentrée (-B1 = 0) (B) -Q2 = 0	
8	comme au point 8 mais activation de l'avance du convoyeur en marche par impulsion (-S3 = 1) \circledast -Q1 = 0 et également -Q2 = 0	
9	Projet archivé correctement	

8.5 Check-list – Exercice

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