

SIMATIC

STEP 7 Pour une transition facile de S5 à S7...

Manuel

Avant-propos, Sommaire

Première partie : Préparer la transition

Introduction

1

Matériel

2

Logiciel

3

Deuxième partie : Conversion du programme

Démarche

4

Préparation de la conversion

5

Conversion

6

Retouche du programme
converti

7

Compilation

8

Exemple d'application

9

Annexes

Listes d'opérandes et
d'opérations

A

Bibliographie

B

Glossaire, Index

Informations relatives à la sécurité

Ce manuel donne des consignes que vous devez respecter pour votre propre sécurité ainsi que pour éviter des dommages matériels. Elles sont mises en évidence par un triangle d'avertissement et sont présentées, selon le risque encouru, de la façon suivante :



Danger

signifie que la non-application des mesures de sécurité appropriées **conduit** à la mort, à des lésions corporelles graves ou à un dommage matériel important.



Attention

signifie que la non-application des mesures de sécurité appropriées peut conduire à la mort, à des lésions corporelles graves ou à un dommage matériel important.



Avertissement

signifie que la non-application des mesures de sécurité appropriées peut conduire à des lésions corporelles légères ou à un dommage matériel.

Nota

doit vous rendre tout particulièrement attentif à des informations importantes sur le produit, aux manipulations à effectuer avec le produit ou à la partie de la documentation correspondante.

Personnel qualifié

La mise en service et l'utilisation de l'équipement ne doivent être effectuées que conformément au manuel.

Seules des **personnes** sont autorisées à effectuer des interventions sur l'équipement. Il s'agit de personnes qui ont l'autorisation de mettre en service, de mettre à la terre et de repérer des appareils, systèmes et circuits électriques conformément aux règles de sécurité en vigueur.

Utilisation conforme aux dispositions



Attention

Le produit ne doit être utilisé que pour les applications spécifiées dans le catalogue ou dans la description technique, et exclusivement avec des périphériques et composants recommandés par Siemens.

Le transport, le stockage, le montage, la mise en service ainsi que l'utilisation et la maintenance adéquats de la console sont les conditions indispensables pour garantir un fonctionnement correct et sûr du produit.

Marque de fabrique

SIMATIC® , SIMATIC NET® et SIMATIC HMI® sont des marques déposées par SIEMENS AG. Les autres désignations figurant dans ce document peuvent être des marques dont l'utilisation par des tiers à leurs propres fins peut enfreindre les droits des propriétaires desdites marques.

Copyright © Siemens AG 1997-2000. Tous droits réservés. Exclure toute responsabilité

Toute communication ou reproduction de ce support d'information, toute exploitation ou communication de son contenu sont interdites, sauf autorisation expresse. Tout manquement à cette règle est illicite et expose son auteur au versement de dommages et intérêts. Tous nos droits sont réservés, notamment pour le cas de la délivrance d'un brevet ou celui de l'enregistrement d'un modèle d'utilité.

Nous avons vérifié la conformité du contenu du présent manuel avec le matériel et le logiciel qui y sont décrits. Or des divergences n'étant pas exclues, nous ne pouvons pas nous porter garants pour la conformité intégrale. Si l'usage de ce manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition. Veuillez nous faire part de vos suggestions.

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D-90327 Nuernberg

© Siemens AG 1997-2000
Sous réserve de modifications

Siemens Aktiengesellschaft

A5E00069888

Avant-propos

Objet du manuel	<p>Ce manuel vous aidera à convertir des programme S5 en programmes S7.</p> <p>Les informations qu'il contient vous permettront :</p> <ul style="list-style-type: none">• de convertir des programmes S5 existants en programmes S7 à l'aide du convertisseur et, le cas échéant, d'apporter des corrections manuellement,• d'utiliser des fonctions de S7 déjà converties (anciens blocs fonctionnels standard de S5) pour les intégrer à vos programmes S7.
Groupe cible	<p>Ce manuel s'adresse aux programmeurs qui souhaitent utiliser les programmes S5 existants dans S7.</p>
Champ d'application	<p>Ce manuel s'applique à la version 4.0 du logiciel de programmation STEP 7.</p>

Vue d'ensemble de la documentation utilisateur S7-300/400

Il existe une importante documentation utilisateur destinée à vous aider pour la configuration et la programmation d'un automate programmable S7 et dont vous vous servirez de manière sélective. Les explications ci-après doivent faciliter l'utilisation de cette documentation.

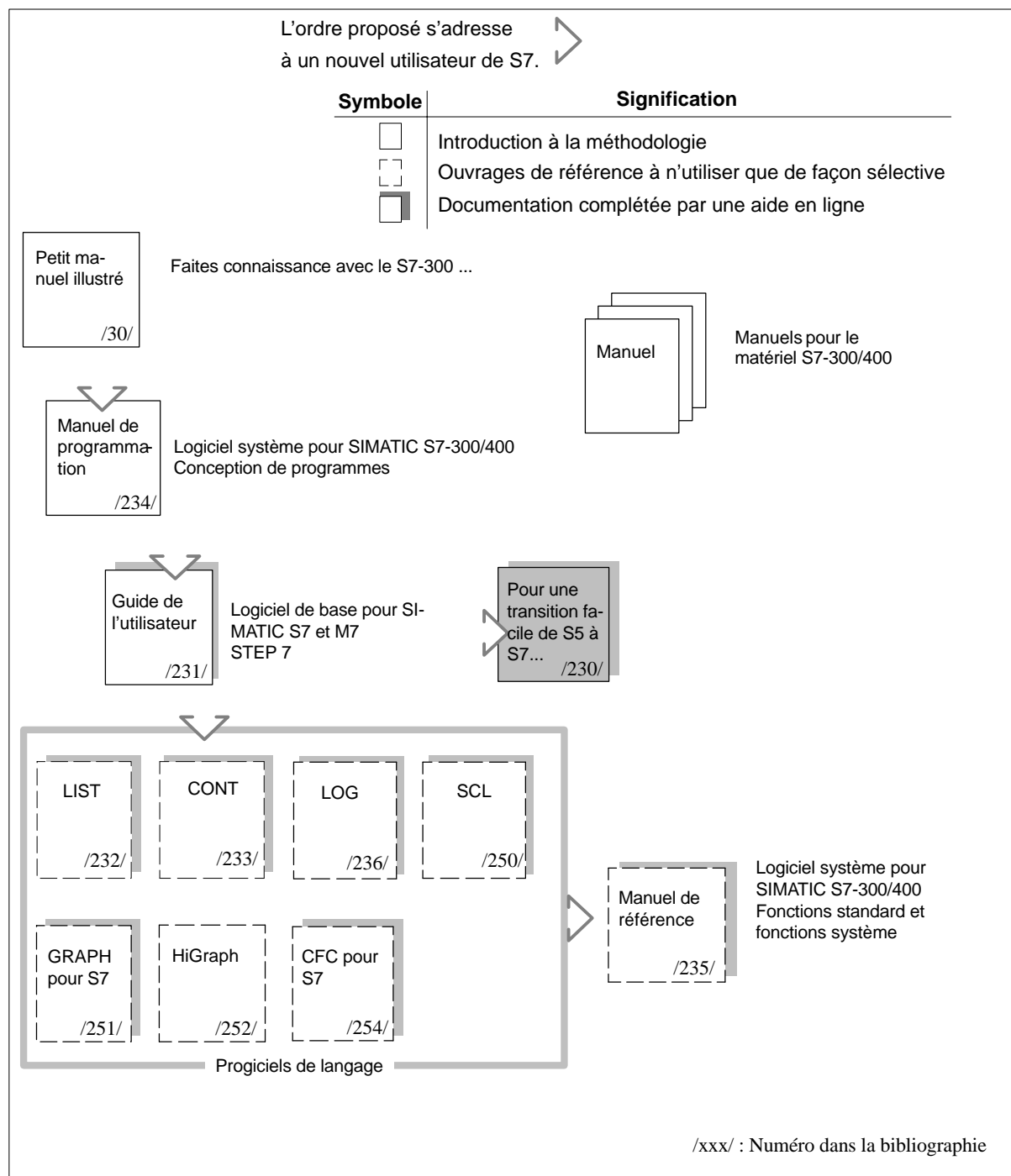


Figure 1-1 Vue d'ensemble de la documentation

Tableau 1-1 Contenu des manuels

Titre	Contenu
Petit manuel illustré Faites connaissance avec le S7-300...	Ce manuel constitue une introduction très simple à la méthodologie de configuration et de programmation d'un automate S7-300/S7-400. Il s'adresse tout particulièrement aux utilisateurs ne connaissant pas les automates programmables S7.
Manuel de programmation Conception de programmes S7-300/400	Ce manuel de programmation présente les connaissances de base sur l'organisation du système d'exploitation et d'un programme utilisateur d'une CPU S7. Il est conseillé aux nouveaux utilisateurs des S7-300/400 de l'utiliser pour avoir une vue d'ensemble de la méthodologie de programmation et pour concevoir, ensuite, leur programme utilisateur.
Manuel de référence Fonctions standard et fonctions système S7-300/400	Les CPU S7 disposent de blocs d'organisation et de fonctions système intégrés au système d'exploitation dont vous pouvez vous servir lors de la programmation. Ce manuel présente une vue d'ensemble des fonctions système, blocs d'organisation et fonctions standard chargeables disponibles dans S7, ainsi que – comme informations de référence – des descriptions d'interface détaillées pour leur utilisation dans le programme utilisateur.
Guide de l'utilisateur STEP 7	Le guide de l'utilisateur STEP 7 explique le principe d'utilisation et les fonctions du logiciel d'automatisation STEP 7. Que vous soyez un utilisateur débutant de STEP 7 ou que vous connaissiez bien STEP 5, il vous donne une vue d'ensemble sur la marche à suivre pour la configuration, la programmation et la mise en œuvre d'un automate S7-300/S7-400. Vous pouvez, lors de l'utilisation du logiciel, accéder de manière sélective à l'aide en ligne qui répondra à vos questions précises sur le logiciel.
Manuels LIST, LOG, CONT, SCL¹	<p>Les manuels concernant les progiciels de langage LIST, LOG, CONT et SCL contiennent aussi bien des instructions pour l'utilisateur que la description du langage. Vous n'avez besoin, pour la programmation d'un S7-300/400, que de l'un de ces langages, mais pouvez les mélanger à l'intérieur d'un projet si besoin est. Il est conseillé, lors de la première utilisation des langages, de se familiariser avec la méthodologie de la création de programmes à l'aide du manuel.</p> <p>Dans le logiciel, vous pouvez appeler l'aide en ligne qui répondra à vos questions détaillées sur l'utilisation des éditeurs et compilateurs associés.</p>
Manuels GRAPH¹, HiGraph¹, CFC¹	<p>Les langages GRAPH, HiGraph et CFC offrent des possibilités supplémentaires pour la réalisation de commandes séquentielles, de graphes d'état ou d'interconnexions graphiques de blocs. Ces manuels contiennent aussi bien des instructions pour l'utilisateur que la description du langage. Il est conseillé, lors de la première utilisation de ces langages, de se familiariser avec la méthodologie de la création de programmes à l'aide du manuel.</p> <p>Dans le logiciel, vous pouvez appeler l'aide en ligne (excepté pour HiGraph) qui répondra à vos questions détaillées sur l'utilisation des éditeurs et compilateurs associés.</p>

¹ Logiciels optionnels pour le logiciel système des S7-300/400

Structure du manuel

L'utilisation du présent manuel requiert des connaissances sur les programmes S7 décrits dans le manuel de programmation /234/. Vous devriez également être familiarisé avec le logiciel de base expliqué dans le guide de l'utilisateur /231/.

Le présent manuel est structuré selon les thèmes suivants :

- La première partie (chapitres 1-3) décrit comment préparer la transition.
- La deuxième partie (chapitres 4-9) explique la conversion du programme à l'aide du convertisseur.
- Le chapitre 9 contient un exemple d'application.
- Vous trouvez à l'annexe A la liste des opérations LIST (abréviations allemandes et internationales) à laquelle vous pourrez toujours vous référer en cas de besoin.
- Vous trouvez les principaux termes utilisés dans le manuel expliqués dans le glossaire.
- L'index vous aide à retrouver des passages du texte à partir de mots-clés importants.

Conventions

Les renvois à d'autres publications sont indiqués à l'aide de numéros entre barres obliques /.../. Vous trouverez, à l'aide de ces numéros, le titre exact de ces publications dans la bibliographie à la fin du manuel.

Aide supplémentaire

Adressez-vous à votre agence Siemens pour toute question sur le logiciel décrit à laquelle vous ne trouveriez pas de réponse dans la documentation papier ou dans l'aide en ligne. Vous trouvez les adresses des agences Siemens dans l'annexe des /70/ et /100/, dans les catalogues ou bien Compuserve (go autforum). Notre support technique est également à votre disposition :

Tel. +49(911) 895-7000 (Fax 7001)

Si vous avez des questions ou des remarques sur le présent manuel, nous vous prions de compléter le formulaire à la fin du manuel et de l'envoyer à l'adresse indiquée. N'hésitez pas à également indiquer votre appréciation personnelle du manuel.

Nous proposons des cours pour faciliter l'apprentissage des automates programmables SIMATIC S7. Adressez-vous pour tous renseignements à notre centre de formation en France au : 01 49 22 28 39.

Remarque

Ce manuel remplace le manuel "Conversion de programmes S5". Il y est souvent fait référence dans les autres manuels sous le nom de manuel de conversion.

Sommaire

Première partie : Préparer la transition

1	Introduction	1-1
2	Matériel	2-1
2.1	Systèmes d'automatisation	2-2
2.2	Modules S7	2-4
2.2.1	Unités centrales (CPU)	2-6
2.2.2	Modules d'alimentation (PS)	2-8
2.2.3	Coupleurs (IM)	2-9
2.2.4	Modules de communication (CP)	2-10
2.2.5	Modules de fonction (FM)	2-13
2.2.6	Modules de signaux (SM)	2-15
2.2.7	Modules de simulation (S7-300)	2-16
2.3	Périphérie décentralisée	2-17
2.4	Communication	2-18
2.4.1	Interface du programme utilisateur	2-20
2.5	Contrôle-commande	2-21
3	Logiciel	3-1
3.1	Principe d'utilisation	3-1
3.1.1	Environnement requis pour l'installation	3-1
3.1.2	Installation du logiciel STEP 7	3-2
3.1.3	Démarrage du logiciel STEP 7	3-3
3.2	Organisation du projet S7	3-4
3.3	Edition du projet dans SIMATIC Manager	3-7
3.3.1	Création du projet	3-7
3.3.2	Sauvegarde des projets	3-8
3.4	Configuration matérielle dans STEP 7	3-9
3.5	Configuration des liaisons dans la table des liaisons	3-11
3.6	Insertion et édition du programme	3-13
3.6.1	Principe de la conception	3-13
3.6.2	Insertion de composants	3-15
3.7	Blocs	3-17
3.7.1	Mise en parallèle	3-17
3.7.2	Fonctions et blocs fonctionnels	3-18
3.7.3	Blocs de données	3-18
3.7.4	Blocs système	3-19
3.7.5	Blocs d'organisation	3-20
3.7.6	Transposition des blocs à la conversion	3-24
3.8	Paramètres système	3-26

3.9	Fonctions standard	3-28
3.9.1	Arithmétique en virgule flottante	3-28
3.9.2	Fonctions de signalisation	3-28
3.9.3	Fonctions intégrées	3-28
3.9.4	Fonctions de base	3-29
3.9.5	Fonctions analogiques	3-29
3.9.6	Fonctions mathématiques	3-29
3.10	Types de données	3-30
3.11	Zones d'opérande	3-32
3.11.1	Vue d'ensemble	3-32
3.11.2	Nouveaux opérandes dans S7 : les données locales	3-33
3.12	Opérations	3-35
3.13	Adressage	3-38
3.13.1	Adressage absolu	3-38
3.13.2	Adressage symbolique	3-38
3.13.3	Nouveauté : accès aux opérandes au format données avec l'adresse complète	3-40
3.13.4	Adressage indirect	3-42
Deuxième partie : Conversion du programme		
4	Démarche	4-1
4.1	Analyse du système S5	4-2
4.2	Création du projet S7	4-3
4.3	Configuration du matériel	4-3
5	Préparation de la conversion	5-1
5.1	Fichiers requis	5-2
5.2	Vérification des opérandes	5-3
5.3	Préparation du programme S5	5-4
5.4	Création de macro-instructions	5-5
5.4.1	Macro-instructions pour opérations	5-6
5.4.2	Macro-instructions pour OB	5-7
5.4.3	Edition de macro-instructions	5-8
6	Conversion	6-1
6.1	Lancement de la conversion	6-1
6.2	Fichiers générés	6-5
6.3	Interprétation des messages	6-8
7	Retouche du programme converti	7-1
7.1	Modifications d'adresses	7-2
7.1.1	Possibilités de modification des adresses	7-2
7.2	Fonctions non convertibles	7-3
7.3	Conversion en cas d'adressage indirect	7-4
7.4	Accès directs à la mémoire	7-5
7.5	Transmission de paramètres	7-5
7.6	Fonctions standard	7-6

8	Compilation	8-1
9	Exemple d'application	9-1
9.1	Traitement des valeurs analogiques	9-2
9.2	Données locales temporaires	9-5
9.3	Evaluation des informations de déclenchement de l'OB82 (alarme de diagnostic)	9-9
9.4	Transfert par blocs	9-12
9.5	Appel des exemples	9-15
Annexes		
A	Listes d'opérandes/opérations	A-1
A.1	Opérandes	A-1
A.2	Opérations	A-3
B	Bibliographie	B-1
	Glossaire	Glossaire-1
	Index	Index-1

Première partie :
Préparer la transition

Introduction	1
Matériel	2
Logiciel	3

Introduction

Le nom de SIMATIC était hier encore associé aux automates et plus précisément aux automates SIMATIC S5. Aujourd'hui SIMATIC est devenu synonyme de l'intégration totale.

L'intégration totale est un concept révolutionnaire visant à réunir l'univers de la fabrication manufacturière et l'univers des procédés. Toutes les briques matérielles et logicielles nécessaires à la réalisation d'un projet portent désormais un seul nom : SIMATIC.

L'intégration totale est rendue possible par l'homogénéité parfaite des données, tant

- au niveau de la base de données,

Les données ne sont plus saisies qu'une seule fois mais sont disponibles dans toute l'usine. Les erreurs dues à la transposition des données et les incohérences appartiennent désormais au passé.

- au niveau de la conception et de la programmation,

Toutes les briques servant à la réalisation du projet sont conçues, configurées, programmées, mises en service, testées et surveillées sous une seule interface utilisateur avec l'outil qui leur est dédié.

- qu'au niveau de la communication.

Vous pouvez voir dans la table des liaisons qui communique avec qui, les liaisons pouvant à tout moment et en tout lieu être modifiées. Il est désormais possible de configurer différents réseaux avec un seul outil simple à utiliser.

Des fonctions résolues différemment dans SIMATIC S5 ont dû être repensées pour céder la place à une approche entièrement nouvelle dans SIMATIC S7.

Le logiciel de programmation STEP 7 mise sur les technologies et les idées nouvelles. Ainsi, la surface utilisateur exécutable sous Windows 95 ou Windows NT a été développée en tenant compte des connaissances ergonomiques les plus modernes. Nous avons tenu à respecter dans nos langages de programmation la norme CEI 1131 tout en restant compatible avec STEP 5.

Nous espérons avoir répondu dans les trois points suivants aux exigences posées à notre logiciel STEP 7.

- Plate-forme logicielle pour une automatisation intégrée
- Programmation conforme à la norme CEI 1131
- Compatibilité avec STEP 5

La décision de passer à un nouveau logiciel s'accompagne toutefois toujours de questions telles que la question que vous vous posez sûrement – "Pourrais-je continuer à utiliser mes programmes écrits en STEP 5 ?" –, questions auxquelles nous nous efforcerons de répondre ici tout en montrant que la transition de STEP 5 à STEP 7 est facile...

Matériel

2

Ce chapitre décrit le matériel mis en œuvre dans S7 et offre un tableau de comparaison des deux matériels utilisés en vue de vous faciliter la transition.

Programme de conversion S5 > S7 offert par le catalogue Siemens sur CD-ROM

Le catalogue du matériel sur CD-ROM “Constituants pour l’automatisation” CA01 (04/97) vous offre une application qui vous aide à convertir votre matériel S5 en matériel S7. Elle peut être lancée en sélectionnant la commande **Auswahlhilfen > Simatic** dans le catalogue produits. Vous pouvez y spécifier une installation S5 quelconque. Cette application génère à partir des données saisies la configuration du châssis avec sa liste d’entrées et sorties. Cette configuration peut ensuite être convertie en une configuration de châssis S7.

2.1 Systèmes d'automatisation

La gamme des nouveaux SIMATIC S7 comprend les systèmes d'automatisation suivants :

SIMATIC S7-200

Le SIMATIC S7-200 est un micro-automate compact d'entrée de gamme. Il possède un progiciel qui lui est propre et n'a pu être pris en compte dans la suite de ce manuel, les propriétés du système S7-200 ne supportant pas une conversion logicielle des programmes S5 en programmes S7.

SIMATIC S7-300

Le mini-automate modulaire SIMATIC S7-300 fait encore partie de l'entrée de gamme.

SIMATIC S7-400

Le SIMATIC S7-400 couvre le milieu, le haut et le très haut de gamme.

Pour faciliter l'orientation, les modules du S7-300 commencent par un 3 et les modules du S7-400 commencent par un 4.

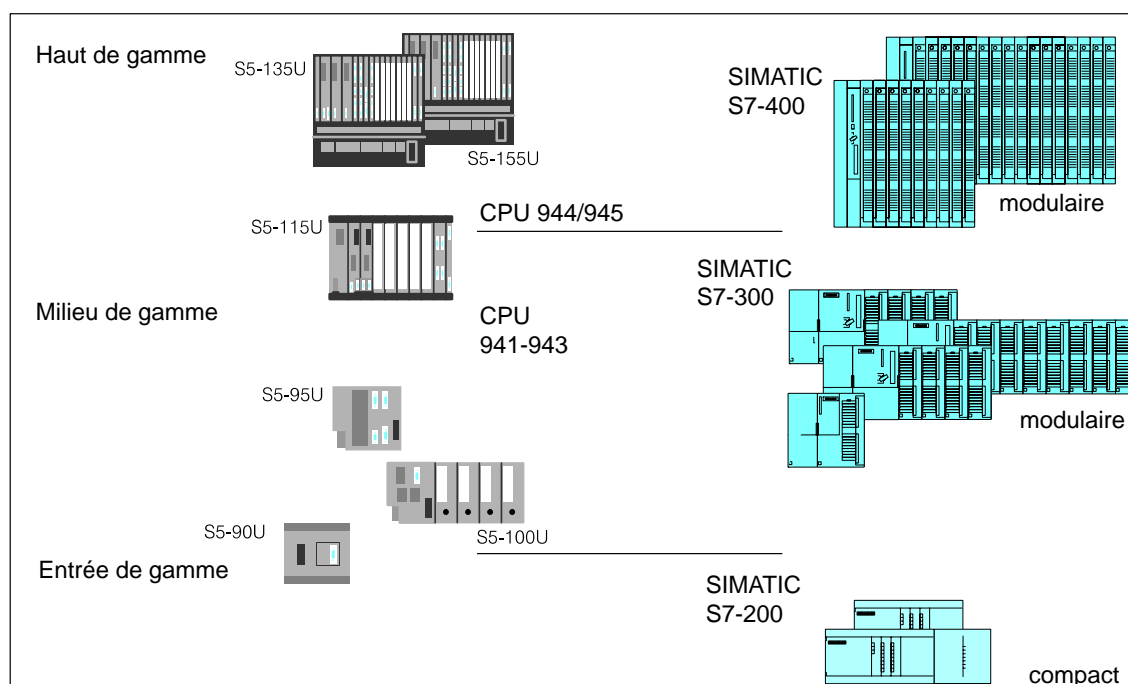


Figure 2-1 Systèmes d'automatisation SIMATIC

Raccordement de PG et OP à SIMATIC S7

Interface MPI (Multi Point Interface) pour PG et OP

L'interface AS511 pour PG du SIMATIC S5 a été remplacée par l'interface multipoint MPI pour S7-300 et S7-400. L'interface multipoint permet de raccorder directement par câble les appareils HMI (Human Machine Interface, auparavant COROS) et les consoles de programmation à l'interface PG des SIMATIC S7. Les interfaces sont intégrées aux appareils.

Leurs caractéristiques techniques sont décrites dans le tableau récapitulatif ci-après.

AS511	MPI
Interface TTY 25 points (20 mA)	Interface Sub-D 9 points, technique RS485
Vitesse de transmission : 9,6 Kbauds	Vitesse de transmission : 187,5 Kbauds
Protocole : 3964R	Protocole : Fonctions S7
	Etendue du réseau : 50 m avec répéteurs de bus ou câbles spéciaux jusqu'à 1000 m Tous les modules programmables communiquent entre eux à l'intérieur d'un réseau <i>via</i> le bus MPI
Un appareil raccordable	Possibilité de raccorder jusqu'à 31 appareils

Interface bus pour OP

Il est possible grâce au système de bus **PROFIBUS** (nouvelle dénomination de l'ancien SINEC L2) de raccorder des automates programmables S5 ou systèmes d'automatisation S7 de la famille SIMATIC. Le raccordement dépend comme auparavant du bus utilisé.

2.2 Modules S7

Peu de changements par rapport à S5

Tout comme la gamme des SIMATIC S5 qu'elle vient compléter, la gamme SIMATIC S7 mise sur la modularité bien éprouvée.

Vous trouvez les types de modules suivants :

- Unités centrales (CPU),
- Modules d'alimentation (PS),
- Coupleurs (IM),
- Processeurs de communication (CP) ; (par exemple pour le raccordement à PROFIBUS),
- Modules de fonction FM ; (par exemple de comptage, régulation, positionnement etc.),
- Les modules d'entrées/sorties TOR ou analogiques sont maintenant appelés des modules de signaux (SM).

Nous présenterons tout au long de ce chapitre les ressemblances et différences entre les modules S5 et les modules S7.

Nouvelles performances

Les modules de STEP 7 se signalent par de nouvelles performances techniques.

- Il n'y a plus de cavaliers ni de commutateurs sur les modules.
- Les modules peuvent fonctionner sans ventilateur. Ils disposent comme les S5 du degré de protection IP 20.
- Vous avez un choix de modules paramétrables et dotés de fonctions de diagnostic !
- Les règles d'enfichage des modules ne sont plus aussi rigides.
- Les appareils d'extension et la périphérie décentralisée ET 200 peuvent déclencher des alarmes.

Paramétrage des modules S5/S7

Le paramétrage des modules SIMATIC S5 et SIMATIC S7 est expliqué dans le tableau suivant.

SIMATIC S5	SIMATIC S7
	Disposition des modules dans l'application de configuration matérielle (HW Config) de STEP 7
Réglage des adresses à l'aide des commutateurs DIL	STEP 7 vous assiste dans l'enfichage des modules et choisit automatiquement les adresses
Réglage du mode de fonctionnement à l'aide de commutateurs	L'application de configuration matérielle de STEP 7 se charge du paramétrage des modules
Paramétrage du mode de fonctionnement des unités centrales par les zones de données système par exemple DB 1/DX 0	La CPU est paramétrée dans HW Config.
	Chargement après compilation des données de configuration dans la CPU et transmission automatique des paramètres aux modules à la mise en route

2.2.1 Unités centrales (CPU)

CPU des S7-300

Le tableau 2-1 présente les caractéristiques des diverses CPU existant pour le S7-300. Vous pouvez y rechercher la CPU de puissance équivalente à la CPU S5 que vous voulez remplacer.

Tableau 2-1 Caractéristiques techniques des CPU S7-300

Caractéristique	312 IFM	313	314	314 IFM	315	315-2 DP
Mémoire de travail (intégrée)	6 kilo-octets	12 kilo-octets	24 kilo-octets	24 kilo-octets	48 kilo-octets	
Mémoire de chargement <ul style="list-style-type: none">intégréeextension par cartes mémoire possible	20 kilo-octets RAM; 20 kilo-octets EEPROM –	20 kilo-octets RAM jusqu’à 512 kilo-octets	40 kilo-octets RAM jusqu’à 512 kilo-octets	40 kilo-octets RAM; 40 kilo-octets EEPROM –	80 kilo-octets RAM jusqu’à 512 kilo-octets (dont 256 kilo-octets pouvant être programmés dans la CPU)	
Taille de la mémoire image entrées/sorties	32 octets + 4 onboard	128 octets	128 octets	124 octets + 4 onboard	128 octets	
Zone d’adresses périphériques <ul style="list-style-type: none">Entrées/sorties TOREntrées/sorties analogiques	Entrées : 128 + 10 onboard Sorties : 128 + 6 onboard	128	512	Entrées : 496 + 20 onboard Sorties : 496 + 16 onboard	1024	
	32		64	Entrées : 64 + 4 onboard Sorties : 64 + 1 onboard	128	
Mémentos	1024	2048				
Compteurs	32	64				
Temporisations	64	128				
Somme des données rémanentes totales	72 octets		4736 octets	144 octets	4736 octets	
Données locales	512 octets au total; 256 octets par classe de priorité	1536 octets au total; 256 octets par classe de priorité				
Blocs :						
OB	3	13	13	13	13	14
FB	32	128	128	128	128	128
FC	32	128	128	128	128	128
DB	63	127	127	127	127	127
SFC	25	44	48	48	48	53
SFB	2	7	7	14	7	7

CPU S7-400

Les CPU S7-400 peuvent avoir des puissances différentes. Le tableau 2-2 affiche leurs principales caractéristiques techniques.

Tableau 2-2 Caractéristiques techniques des CPU S7-400

Caractéristique	CPU 412-1	CPU 413-1	CPU 413-2 DP	CPU 414-1	CPU 414-2 DP	CPU 416-1	CPU 416-2 DP
Mémoire de travail intégrée	48 kilo-octets	72 kilo-octets		128 kilo-octets	128/384 kilo-octets	512 kilo-octets	0,8/1,6 Moctets
Mémoire de chargement <ul style="list-style-type: none">intégréeextension par cartes mémoire possible	8 kilo-octets à 15 Mégaoctets			8 kilo-octets à 15 Mégaoctets		16 kilo-octets à 15 Mégaoctets	
Taille de la mémoire image entrées/sorties	128 octets			256 octets		512 octets	
Zone d'adresses périphé- riques <ul style="list-style-type: none">Entrées/sorties TOR max.Entrées/sorties analo- giques max.	2 kilo-octets 16384 1024			8 kilo-octets 65536 4096		16 kilo-octets 131072 8192	
Mémentos	4096 M 0.0 à M 511.7			8192 M 0.0 à M 1023.7		16384 M 0.0 à M 2047.7	
Compteurs	256 Z 0 à Z 255			256 Z 0 à Z 255		512 Z 0 à Z 511	
Temporisations	256 T 0 à T 255			256 T 0 à T 255		512 T 0 à T 511	
Données locales	4 kilo-octets au total			8 kilo-octets au total		16 kilo-octets au total	
Blocs :							
OB	23			31		44	
FB	256			512		2048	
FC	256			1024		2048	
DB	511			1023		4095	
SFB	24			24		24	
SDB	512			512		512	
SFC	55	55	58	55	58	55	58

Rémanence du S7-400

Les unités centrales du SIMATIC S7-400 ont besoin d'une pile de sauvegarde pour la sauvegarde des temporisations, compteurs et mémentos.

Rémanence sans pile du S7-300

Vous n'avez pas besoin de pile pour sauvegarder les temporisations, compteurs et mémentos du S7-300. Le contenu des blocs de données y est également sauvegardé en cas de coupure de la tension. Les unités centrales des SIMATIC S7-300 ont une mémoire tampon de secours ne nécessitant aucune maintenance qui sauvegarde les opérandes et les données qui ont été déclarés rémanents.

Le nombre et la taille des zones rémanentes autorisées sont fonction de la CPU.

Paramétrer la rémanence

Vous définissez les zones de mémoire pour lesquelles vous voulez la rémanence dans l'application de configuration matérielle de STEP 7.

2.2.2 Modules d'alimentation (PS)

Vous disposez pour chaque système d'automatisation de différents modules d'alimentation.

Module d'alimentation pour S7-300

Tout réseau 24 V industriel peut être utilisé pour alimenter la CPU du S7-300.

Les modules d'alimentation suivants de la gamme S7 sont prévus pour être utilisés avec le S7-300 :

Désignation	Courant de sortie	Tension à la sortie	Tension à l'entrée
PS 307	2A	DC 24V	AC 120V / 230V
PS 307	5A	DC 24V	AC 120V / 230V
PS 307	10A	DC 24V	AC 120V / 230V

Module d'alimentation pour S7-400

Désignation	Courant de sortie	Tension à la sortie	Tension à l'entrée
PS 407 4A	4A 0,5A	DC 5V DC 24V	AC 120V / 230V
PS 407 10A	10A 1A	DC 5V DC 24V	AC 120V / 230V
PS 407 20A	20A 1A	DC 5V DC 24V	AC 120V / 230V
PS 405 4A	4A 0,5A	DC 5V DC 24V	DC 24V
PS 405 10A	10A 1A	DC 5V DC 24V	DC 24V
PS 405 20A	20A 1A	DC 5V DC 24V	DC 24V

Pour plus d'informations sur les modules d'alimentation, veuillez vous référer aux manuels **/71/** et **/101/**.

2.2.3 Coupleurs (IM)

Certains coupleurs disponibles dans S5 ont leur équivalent dans S7. Ils peuvent être utilisés pour un couplage sur de courtes distances. Pour un couplage sur de longues distances, il est recommandé d'émettre les signaux via le bus PROFIBUS.

Comparaison des modules IM

Module S5	Module S7-300	Module S7-400	Description
IM 305 IM 306 IM 300 / IM 312	IM 365 IM 360 / IM 361	IM 460-0 / IM 461-0 IM 460-1 / IM 461-1	Configuration centralisée
-	-	IM 460-3 / IM 461-3	Longue distance (jusqu'à 100 m)
IM 301 / IM 310	Couplage via PROFIBUS	Couplage via PROFIBUS	Couplage de modules de périphérie et de prétraitement de signaux (jusqu'à 200 m)
IM 304 / IM 314	Couplage via PROFIBUS	Couplage via PROFIBUS	Pour la mise en œuvre de périphérie décentralisée sur longue distance (jusqu'à 600 m)
	–	IM 463-2	Couplage décentralisé des appareils d'extension S5 sur longue distance (jusqu'à 600 m)
IM 307 / IM 317	Couplage via PROFIBUS	Couplage via PROFIBUS	Couplage via câble à fibres optiques (jusqu'à 1500 m)
IM 308 / IM 318	Couplage via PROFIBUS	Couplage via PROFIBUS	Jusqu'à une distance de 3000 m

Il est possible d'utiliser dans S7 le coupleur IM 467 à la place du coupleur IM 308C.

Les modules TOR et analogiques S5 peuvent être couplés au châssis S7 à l'aide du coupleur IM 463-2 via un châssis d'extension S5 et un coupleur IM 314.

Châssis d'extension S5 compatibles

Les châssis d'extension S5 suivants peuvent être utilisés avec les modules S7 :

- EG 183
- EG 185
- ER 701-2
- ER 701-3

2.2.4 Modules de communication (CP)

Les modules de communication S5 et S7 pouvant être employés dans les différents réseaux sont énumérés ci-dessous avec leurs fonctions.

Sous-réseaux de SIMATIC

Pour répondre aux différents besoins des niveaux de l'automatisation (conduite, cellule, terrain, actionneurs/capteurs) SIMATIC met à votre disposition les sous-réseaux suivants :

- **Interface AS-i**

L'interface AS-i (*Actuator Sensor Interface*) est un système de connexion employé pour le premier niveau du processus dans les installations d'automatisation. Il permet notamment de relier des capteurs et actionneurs échangeant des données binaires. La quantité des données ne doit pas dépasser 4 bits par esclave.

- **MPI**

Le sous-réseau MPI est adapté au niveau du terrain et de la cellule lorsqu'ils restent de dimensions modestes. L'interface MPI est une interface multipoint pour SIMATIC S7/M7 et C7. Elle est également conçue comme interface PG et peut servir à la mise en réseau de plusieurs CPU pour l'échange de petites quantités de données (jusqu'à 70 octets).

- **PROFIBUS**

PROFIBUS est un système de communication ouvert acceptant en plus des SIMATIC les appareils d'autres constructeurs. C'est le réseau idéal à l'échelle de la cellule et du terrain et permet la transmission rapide de moyennes quantités de données (environ 200 octets).

- **Industrial Ethernet**

Industrial Ethernet est un système de communication ouvert acceptant en plus des SIMATIC les appareils d'autres constructeurs. C'est le réseau idéal pour le niveau de la conduite et de la cellule et permet la transmission rapide de grandes quantités de données.

- **Couplage point-à-point**

Un couplage point-à-point n'est pas un sous-réseau à proprement parler. Il est réalisé dans SIMATIC par des processeurs de communication (CP) point-à-point et relie deux partenaires de communication (automates programmables, scanner, PC, etc.).

**Interface AS-i
(SINEC S1)**

Vous disposez des modules énumérés dans le tableau ci-dessous pour la communication via l'interface AS-i.

Module S5	Module S7-300	Module S7-400
CP 2433 (fonctions AS-i) CP 2430 (fonctions AS-i)	CP 342-2 (fonctions AS-i)	-

MPI (SINEC L1)

La communication par le réseau SINEC L1 de S5 a été remplacée dans S7 par la communication par données globales et l'interface multipoint.

Toutes les CPU des S7-300 et S7-400, toutes les consoles de programmation (PG) et tous les pupitres opérateurs (OP) possèdent une interface multipoint (MPI).

**PROFIBUS
(SINEC L2)**

Vous voyez dans le tableau ci-dessous les modules de communication pouvant être utilisés avec PROFIBUS avec leurs fonctions.

Module S5	Module S7-300	Module S7-400
CP5431 (FMS, FDL, DP) CPU 95U (FDL, DP *)	CP 342-5 (fonctions S7, FDL, DP) CP 343-5 (fonctions S7, FDL, FMS)	CP 443-5 Ext. (fonctions S7, FDL, DP) CP 443-5 Basic (fonctions S7, FDL, FMS)
IM 308-B/C (DP)	CPU 315-2 DP (DP)	CPU 413-2 DP (DP) CPU 414-2 DP (DP) CPU 416-2 DP (DP) IM 467 (DP)

*) selon la version

**Industrial Ethernet
(SINEC H1)**

Le tableau ci-dessous vous donne un aperçu des modules disponibles pour la communication via le réseau Industrial Ethernet et des fonctions supportées.

Module S5	Module S7-300	Module S7-400
CP1430 TF (transport ISO)	CP 343-1 (fonctions S7, transport ISO)	CP 443-1 (fonctions S7, transport ISO)
CP 1430 TCP (ISO on TCP)	CP 343-1 TCP (fonctions S7, ISO on TCP)	CP 443-1 TCP (fonctions S7, ISO on TCP)

**Couplage
point-à-point**

Le tableau ci-dessous vous donne un aperçu des modules disponibles pour la communication multipoint et des fonctions supportées.

Module S5	Module S7-300	Module S7-400
CP 521 (3964 (R), ASCII) CP 523 (3964 (R), ASCII)	CP 340-RS 232C (3964 (R), ASCII) CP 340-20 mA (3964 (R), ASCII) CP 340-RS 422/485 (3964 (R), ASCII)	CP 441-1 (3964 (R), RK512, ASCII)
CP 544 (3964 (R), RK 512, ASCII)	-	
CP 524/525 (3964 (R), RK 512, ASCII, pilotes spéciaux pouvant être chargés) CP 544 B (3964 (R), RK 512, ASCII, pilotes spéciaux pouvant être chargés)	-	CP 441-2 (3964 (R), RK512, ASCII, pilotes spéciaux pouvant être chargés)

2.2.5 Modules de fonction (FM)

L'utilisation d'un boîtier d'adaptation permet de monter des cartes d'axes (IP) et de positionnement (WF) de SIMATIC S5 dans des châssis S7-400. Choisissez autrement le module de fonction remplissant la fonctionnalité voulue dans la gamme disponible des modules de fonction S7.

Le tableau suivant met en parallèle les modules de prétraitement de signaux S5 et S7.

Tableau 2-3 Mise en parallèle des modules de prétraitement de signaux S5 et S7

Module S5	Boîtier d'adaptation	Module S7	Description
IP 240	oui	FM 451 (partiellement)	Module compteur, capteur de déplacement et de positionnement
IP 241	non	FM 451 / FM 452 (partiellement)	Capteur de déplacement TOR
IP 242A	non	non	Module de comptage
IP 242B	oui	non	Module de comptage
IP 244	oui	FM 455	Module de régulation
IP 246I/A	oui	FM 354 / FM 357 / FM 453	Module de commande d'axe pour servomoteurs
IP 247	oui	FM 353 / FM 357 / FM 453	Module de positionnement pour moteur pas-à-pas
IP 252	non	FM 455 (partiellement)	Module de régulation
IP 260	non	FM 355 (partiellement)	Module de régulation
IP 261	non	non	Module de dosage
IP 281	non	FM 350-1 / FM 450-1	Module de comptage
IP 288	non	FM 451 / FM 452	Module de positionnement pour entraînements à deux vitesses et came électronique
WF 705	oui	FM 451 (partiellement)	Came électronique
WF 706	non	FM 451 (partiellement)	Module de comptage et de positionnement
WF 707	non	FM 452 (partiellement)	Came électronique
WF 721	oui	FM 354 (selon la technique de montage)	Module de positionnement
WF 723A	oui	FM 453	Module de positionnement

Tableau 2-3 Mise en parallèle des modules de prétraitement de signaux S5 et S7, Fortsetzung

Module S5	Boîtier d'adaptation	Module S7	Description
WF 723 B	oui	FM 357 (selon la technique de montage)	Module de positionnement
WF 723 C	oui	non	Module de positionnement
-	-	FM 456-4	Module technologique personnalisable (M7-FM)
-	-	SINUMERIK FM-NC	Commande NC
-	-	FM STEPDRIVE	Commande de moteurs pas-à-pas
-	-	SIMOSTEP	Moteur pas-à-pas

2.2.6 Modules de signaux (SM)

Les modules de signaux dans SIMATIC S7 correspondent aux cartes d'entrées/sorties de S5. Des nouveautés par rapport à S5 sont dans S7 les modules de signaux paramétrables et les modules permettant le diagnostic.

SM paramétrables

Vous avez par exemple la possibilité pour les modules d'entrées TOR paramétrables d'indiquer dans l'application de configuration du matériel de STEP 7 quelles voies devront déclencher une alarme de processus lors d'un changement de front.

Les plages d'entrées des modules d'entrées analogiques se laissent facilement paramétrer dans STEP 7.

SM avec fonctions de diagnostic

Les modules permettant le diagnostic sont en mesure de détecter des erreurs provenant d'une source externe telles rupture de fil ou court-circuit et des erreurs internes telles erreur de RAM ou court-circuit interne.

Un événement de diagnostic donne lieu dans l'automate :

- au déclenchement d'une alarme de diagnostic (provoquant l'appel dans le programme utilisateur de l'OB correspondant) qui interrompt le programme cyclique
- à une entrée dans la mémoire tampon de diagnostic de la CPU (accessible depuis la PG ou l'appareil de contrôle-commande)

Les modules de signaux S7 sont énumérés dans les tableaux suivants :

Tableau 2-4 Modules de signaux pour SIMATIC S7-300

DI (SM 321)	DO (SM 322)	AI (SM 331)	AO (SM 332)
32 x DC 24V	32 x DC 24V/0,5A	8 x 12 Bit	2 x 12 Bit
16 x DC 24V	16 x DC 24V/0,5A	2 x 12 Bit	
16 x DC 24V avec alarme de processus et de diagnostic	8 x DC 24V/0,5A avec alarme de diagnos- tic	Ex: 4 x 15 Bit	Ex: 4 x 15 Bit
16 x DC 24V de type N	8 x DC 24V/2A	Ex: 12 x 15 Bit	
8 x AC 120V/230V	8 x AC 120V/230V/ 2A	AI 4/AO 2 X 8/8 Bit (SM 334)	
Ex: 4 x DC 24V	Ex: 4 x DC 15V/ 20mA		
	Ex: 4 x DC 24V/ 20mA		

Tableau 2-5 Modules de signaux pour SIMATIC S7-400

DI (SM 421)	DO (SM 422)	AI (SM 431)	AO (SM 432)
32 x DC 24V	32 x DC 24V/0,5A	8 x 13 Bit	8 x 13 Bit
16 x UC 24V/60V avec alarme de pro- cessus et de dia- gnostic	16 x DC 24V/2A	8 x 14 Bit (pour thermomé- trie)	
16 x UC 120V/230V	16 x AC 120V/230V /5A	8 x 14 Bit	
32 x UC 120V	16 x AC 120V/230V /2A	16 x 16 Bit	
	16 x UC 30V/230V/ Rel 5A		

2.2.7 Modules de simulation (S7-300)

Vous disposez pour tester votre programme du module de simulation SM 374 pour S7-300.

Il possède les caractéristiques suivantes :

- Simulation de
 - 16 entrées ou
 - 16 sorties ou
 - 8 entrées et 8 sorties
(ayant la même adresse de début !)
- Fonction pouvant être sélectionnée à l'aide d'un tournevis
- LED simulant l'état des entrées ou sorties

2.3 Périphérie décentralisée

Les modules pour périphérie décentralisée du système ET 200 que vous avez utilisés dans SIMATIC S5 peuvent être réutilisés dans SIMATIC S7.

Les nouveaux modules ET 200 viennent compléter la gamme déjà existante.

Maître DP

Les modules suivants peuvent assumer le rôle de **maître DP** dans le système de périphérie décentralisée :

- S7-300 avec CPU 315-2 DP ou CP 342-5 utilisé comme maître DP
- S7-400 avec CPU 413-2 DP / 414-2 DP / 416-2DP ou CP 443-5 Extended utilisé comme maître DP

Esclaves DP

Les appareils suivants peuvent par exemple servir d'**esclaves DP** dans un système de périphérie décentralisée :

- Appareils de périphérie décentralisée ET 200B, ET 200C, ET 200M, ET 200X (jusqu'à 12 Mbauds) et ET 200U, ET 200L (jusqu'à 1,5 Mbauds)
- Automates programmables/systèmes d'automatisation tels que des
 - S5-115U, S5-135U ou S5-155U avec un IM 308-C utilisé comme esclave DP
 - S5-95U avec interface esclave DP (jusqu'à 1,5 Mbauds)
 - S7-300 avec CPU 315-2 DP ou CP 342-5 utilisé comme esclave DP
 - S7-400 avec CP 443-5 utilisé comme esclave DP
- DP/AS-i Link assurant le lien à l'interface AS-I
- Afficheurs de texte et pupitres opérateurs pour un contrôle-commande à proximité des machines
- Système d'identification MOBY
- Appareils de coupure basse tension
- Appareils de terrain de Siemens ou d'autres constructeurs, par exemple entraînements, groupes de vannes etc.

Maître FMS

Vous pouvez avoir comme **maître FMS** un :

- S7-300 avec CP 343-5 utilisé comme maître FMS
- S7-400 avec CP 443-5 Basic utilisé comme maître FMS

Esclaves FMS

Vous pouvez trouver comme **esclaves FMS** une ET 200U ou l'appareil de commande ou de protection de moteur SIMOCODE.

Pour plus d'informations, veuillez consulter les manuels respectifs de ces appareils ou le catalogue de Siemens CA01.

2.4 Communication

Services et sous-réseaux

S7 a recours pour la communication à différents sous-réseaux. Les services offerts par ceux-ci sont énumérés ci-après.

Services	Fonctions de communication S7 (Fonctions S7)		
	Transport ISO ISO on TCP	FDL (SDA) FMS DP	GD
Sous-réseaux	Industrial Ethernet	PROFIBUS	MPI

Fonctions S7

Les services qu'offrent les fonctions S7 permettent la communication entre les CPU S7/M7, les OP/OS et les PC. Ces fonctions sont déjà intégrées dans tout appareil SIMATIC S7/M7. Ces fonctions correspondent à un service du système de référence pour les systèmes ouverts OSI et peuvent être donc utilisées quel que soit le réseau (MPI, PROFIBUS, Industrial Ethernet).

Transport ISO

Les fonctions Transport ISO qui s'appliquent dans la 4ème couche du modèle OSI permettent la transmission de quantités moyennes de données (jusqu'à 240 octets) et garantissent un transport sûr des données de SIMATIC S7 à SIMATIC S5.

ISO on TCP

Les fonctions ISO on TCP qui s'appliquent dans la 4ème couche du modèle OSI (protocole TCP/IP) permettent la transmission de quantités moyennes de données (jusqu'à 240 octets) et garantissent un transport sûr des données de SIMATIC S7 à SIMATIC S5.

Le service ISO on TCP requiert le standard RFC1006 étendu.

FDL (SDA)

Les fonctions FDL (Fieldbus Data Link) qui s'appliquent dans la 2ème couche du modèle OSI permettent la transmission de quantités moyennes de données (jusqu'à 240 octets) et garantissent un transport sûr des données de SIMATIC S7 à SIMATIC S5.

FMS

Le protocole FMS (Fieldbus Message Specification) PROFIBUS permet d'envoyer des données structurées (variables FMS) via des liaisons statiques FMS.

Le service FMS peut être intégré sur la 7ème couche du modèle de référence OSI et est conforme à la norme européenne EN 50170 Vol. 2 PROFIBUS.

DP

Grâce aux services PROFIBUS-DP, la communication avec la périphérie décentralisée devient transparente. La périphérie décentralisée réagit comme la périphérie centralisée directement au programme de commande.

GD

La communication par données globales est un système de communication simple intégré au système d'exploitation des CPU des S7-300/400.

La communication par données globales permet un échange de données entre CPU via l'interface MPI soit cyclique, soit déclenché par l'événement pour le S7-400.

2.4.1 Interface du programme utilisateur

Les blocs suivants constituent l'interface de communication du programme utilisateur :

- SFC (ne nécessitent pas de configurer de liaisons)
- SFB (pour lesquels des liaisons doivent être configurées) (uniquement pour le S7-400)
- FC/ FB pouvant être chargés

Ces blocs remplacent les blocs de dialogue S5. Leur fonctionnalité est identique, quoique réalisée avec les nouveaux moyens mis à disposition par les langages de STEP 7. Un programme S5 ayant des fonctions de dialogue doit être modifié en conséquence. Vous remplacez pour cela les anciens blocs de dialogue par les nouveaux blocs de communication.

Réseau	Service	Interface du programme utilisateur S5	Interface du programme utilisateur S7
Couplage point-à-point	-	Blocs de dialogue *	S7-300: FB chargeables S7-400: SFB chargeables
PROFIBUS	FDL (AP - AP) Couche 2 libre FMS	Blocs de dialogue * Blocs de dialogue * Blocs de dialogue *	FC chargeables - FB chargeables
Industrial Ethernet	ISO 4 ISO 4 + AP STF MAP	Blocs de dialogue * Blocs de dialogue * Blocs de dialogue * + FB chargeables Blocs de dialogue * + FB chargeables	FC chargeables - - FB chargeables

* sont selon la CPU intégrés ou chargeables.

2.5 Contrôle-commande

Introduction	Les paragraphes suivants vous informent au sujet des possibilités de mise en œuvre des pupitres opérateurs HMI SIMATIC (H uman M achine I nterface, anciennement COROS) avec SIMATIC S7.
Pupitres opérateur (OP)	Les pupitres opérateur SIMATIC HMI offrant des fonctionnalités C+C peuvent être utilisés conjointement à un SIMATIC S5, SIMATIC S7 et SIMATIC TI (et bien d'autres automates).
STEP 5	<p>La connexion d'un OP SIMATIC à un SIMATIC S5 requiert généralement un bloc fonctionnel standard dans l'API qui doit être appelé pour chaque OP connecté.</p> <p>Les OP suivants peuvent être utilisés avec un S5 :</p> <ul style="list-style-type: none"> • TD17, OP5/A1, OP7/PP, OP7/DP-12, OP15/x1, OP17/PP, OP17/DP-12 • OP25, OP35, OP37, TP37
STEP 7	<p>Il existe différentes possibilités pour coupler un OP SIMATIC à un S7/M7 SIMATIC. Nous considérerons successivement le couplage point-à-point (PPI), multipoint (MPI) et PROFIBUS (partenaire MPI).</p> <p>Le couplage PPI ou MPI passe par l'interface PG de la CPU. Les OP SIMATIC se servent pour cela des services de communication SIMATIC S7/M7 (fonctions S7); un bloc fonctionnel standard n'est donc pas nécessaire !</p> <p>Le couplage PROFIBUS se sert également des fonctions de communication S7 et ne requiert donc pas de FB standard ! (les OP SIMATIC sont des "partenaires actifs" et non des esclaves PROFIBUS-DP comme c'est le cas pour le couplage PROFIBUS à des S5 SIMATIC). Le nombre de stations pouvant être couplées est le même que pour le couplage multipoint.</p> <p>Les OP suivants peuvent être utilisés avec des appareils S7 :</p> <ul style="list-style-type: none"> • TD17, OP3, OP5/A2, OP7/DP, OP7/DP-12, OP15/x2, OP17/DP, OP17/DP-12 • OP25, OP35, OP37, TP37 <p>Du côté de l'OP valent les restrictions suivantes :</p> <ul style="list-style-type: none"> • OP3 : max. 2 liaisons • OP5/15/25 : max. 4 liaisons • TD17, OP7/17 : max. 4 liaisons • OP35 : max. 6 liaisons • OP37, TP37 : max. 8 liaisons
Configuration	SIMATIC ProTool et SIMATIC ProTool/Lite sont des outils modernes de configuration permettant de configurer les pupitres opérateur. SIMATIC ProTool permet de configurer tous les types d'appareil. La version allégée SIMATIC ProTool/Lite ne permet que la configuration des pupitres opérateur orientés lignes.

Intégration dans SIMATIC STEP 7

ProTool peut être intégré au logiciel de configuration STEP 7 de SIMATIC rendant ainsi possible l'accès direct aux données de configuration de l'automatisme telles que listes de mnémoniques, paramètres de communication etc., ce qui vous fait faire des économies de temps et d'argent et met fin aux saisies multiples, souvent sources d'erreurs.

Tableau 2-6 Outils de configuration pour appareils de contrôle-commande

Appareil	Outil de configuration
OP orienté lignes (TD17, OP 3, OP 5, OP7, OP 15, OP17)	ProTool/Lite ou ProTool
OP orienté graphique (OP 25, OP 35, OP37, TP37)	ProTool

WinCC

WinCC peut être utilisé en système monoposte ou multiposte (architecture client/serveur).

Le système de conduite et de surveillance WinCC peut être mis en œuvre dans tous les domaines et servir plusieurs technologies. Il est la solution pour la visualisation d'automatisation tant dans l'industrie manufacturière que dans l'industrie de process. Ses modules fonctionnels propres à répondre aux exigences industrielles permettent la signalisation, l'archivage et la journalisation des données du processus. Un couplage puissant au processus, un rafraîchissement rapide des images et l'archivage fiable des données en font un système de haute disponibilité.

Outre ces fonctions système, WinCC offre la possibilité d'interfaçage avec des automatismes utilisateur rendant possible l'intégration de WinCC dans des concepts d'automatisation complexes à l'échelle d'une usine par exemple. Y sont intégrés l'accès aux données archivées via ODBC et SQL, interfaces standard ainsi que la liaison d'objets et de documents via OLE 2.0 et OLE Custom Controls (OCX). Ces mécanismes font de WinCC un partenaire compétent et ouvert dans le monde de Windows.

WinCC a pour base les systèmes d'exploitation 32 bits MS Windows 95 ou MS Windows NT. Tous deux sont des systèmes multitâche préemptifs permettant de réagir rapidement aux événements du processus et garantissant une grande sécurité des données. Windows NT offre en outre des fonctions de sécurité et sert de base au fonctionnement serveur en système multiposte. WinCC est une application 32 bits orientée objet mise au point selon les techniques de conception de logiciel les plus modernes.

Logiciel

3.1 Principe d'utilisation

Présentation Le logiciel de configuration et de programmation des SIMATIC S7/M7/C7 a été conçu selon les connaissances ergonomiques modernes et son apprentissage est intuitif.

3.1.1 Environnement requis pour l'installation

Système d'exploitation Microsoft Windows 95.

Matériel de base Console de programmation ou PC avec

- Processeur 80486 (ou supérieur) et
- 16 Mo minimum de RAM, 32 Mo étant recommandés,
- moniteur VGA ou un autre moniteur pris en charge par Microsoft Windows 95,
- clavier, et en option mais fortement recommandé une souris pris en charge par Microsoft Windows 95.

Capacité mémoire Vous devez disposer de la capacité suivante sur votre disque dur

- 105 Mo pour l'installation du logiciel de base en une langue, les besoins en mémoire dépendent des options choisies à l'installation.
- STEP 7 a besoin de 64 Mo moins la capacité réservée à la mémoire centrale pour ses fichiers d'échange (c'est-à-dire 32 Mo pour une mémoire centrale de 32 Mo).
- Prévoyez 50 Mo pour vos données utilisateur.
- 1 Mo d'espace libre sur le lecteur C: pour le Setup (les fichiers Setup sont effacés, l'installation une fois achevée).

3.1.2 Installation du logiciel STEP 7

Présentation

STEP 7 contient un programme d'installation "Setup" qui se charge de l'installation pour vous. Les instructions affichées à l'écran vous guident tout au long de la procédure d'installation.

Autorisation

Une autorisation spécifique au produit (licence d'utilisation) est nécessaire pour pouvoir utiliser le logiciel de programmation STEP 7. Le logiciel ainsi protégé n'est utilisable que si l'autorisation nécessaire pour le programme ou le progiciel est détectée sur le disque dur de la PG ou du PC concerné.

Cette autorisation ainsi que le programme AUTHORS permettant d'afficher, installer et désinstaller les autorisations se trouvent sur la disquette protégée en écriture fournie avec le logiciel.

La procédure d'installation ou de désinstallation des autorisations est décrite dans le guide de l'utilisateur **/231/**.

Nota

Les consoles de programmation de Siemens, comme la PG 740, sont livrées avec, sur leur disque dur, le logiciel STEP 7 installable.

Pour plus d'informations sur l'installation, veuillez consulter le guide de l'utilisateur **/231/**.

3.1.3 Démarrage du logiciel STEP 7

Mise en route

Une fois Windows 95 ou Windows NT démarré, vous trouverez dans l'interface Windows une icône pour le gestionnaire de projets SIMATIC (SIMATIC Manager) qui permet d'accéder au logiciel STEP 7.

Vous démarrez rapidement STEP 7 en effectuant un double clic sur l'icône "SIMATIC Manager". La fenêtre du gestionnaire de projets SIMATIC s'affiche alors. De là, vous pouvez accéder à toutes les fonctions du logiciel de base ou des logiciels optionnels installés.

L'autre méthode consiste à lancer le gestionnaire de projets SIMATIC via le bouton "Démarrer" dans la barre des tâches de Windows 95/NT (sous "Simatic/STEP 7".)

SIMATIC Manager

Le gestionnaire de projets SIMATIC (SIMATIC Manager) constitue l'interface d'accès à la configuration et à la programmation. Vous pouvez :

- créer des projets,
- configurer et paramétrer le matériel,
- configurer les liaisons de communication,
- élaborer vos programmes,
- tester et mettre en œuvre vos programmes.

L'interface utilisateur est orientée objet et ses fonctions sont simples à utiliser.

Avec le gestionnaire de projets SIMATIC vous pouvez travailler :

- hors ligne, c'est-à-dire sans qu'un automate soit raccordé ou
- en ligne, c'est-à-dire avec un automate raccordé

Tenez compte, dans ce dernier cas, des remarques relatives à la sécurité.

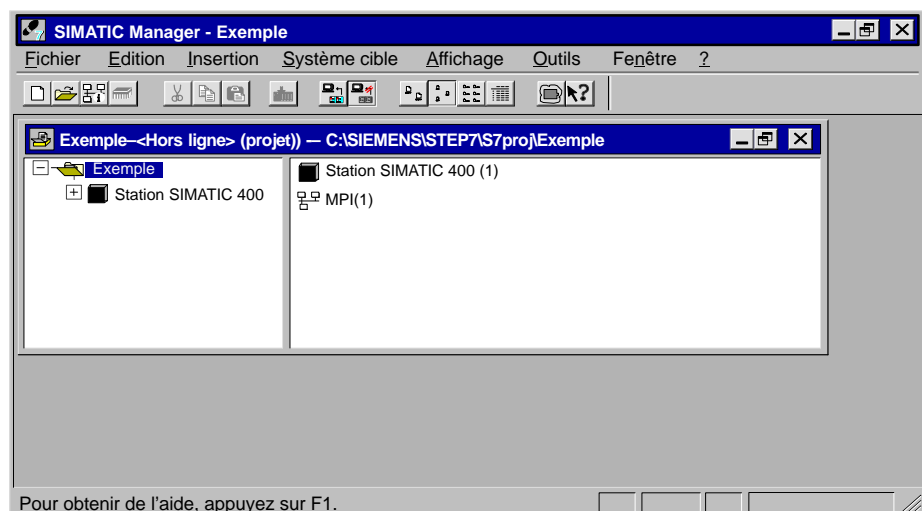


Figure 3-1 Projet ouvert dans SIMATIC Manager

3.2 Organisation du projet S7

Définition	On entend par "Projet" l'ensemble des données et programmes appartenant à une solution d'automatisation. Dans STEP 7, le terme désigne un classeur permettant de regrouper toutes ces données.
Projet de STEP 5	<p>Vous connaissiez déjà le concept de "Projet" dans STEP 5. Il regroupait dans STEP 5 tous les fichiers STEP 5 créés dans un programme utilisateur en un fichier de projet.</p> <p>Le fichier de projet servait dans STEP 5 à la sauvegarde d'informations en vue de faciliter l'édition et la maintenance du programme utilisateur, telles que choix de paramètres et noms de fichiers ou de répertoires.</p>
Projet de STEP 7	Dans STEP 7, le projet regroupe la totalité des données et des programmes d'une solution d'automatisation, quels que soient le nombre d'unités centrales et leur mise en réseau. Un projet n'est donc pas limité à un programme utilisateur d'un module programmable, mais pourra réunir sous un nom de projet commun plusieurs programmes utilisateur pour plusieurs modules programmables.
Remarque	Vous avez évidemment toujours la possibilité dans STEP 7 de créer, comme vous étiez habitué à le faire dans STEP 5, un programme utilisateur simple pour une seule unité centrale. Un projet se limitera dans ce cas à une CPU. (Voir ci-dessous la structure hiérarchique du projet STEP 7.)

Composants d'un projet

Les principaux objets du projet STEP 7 sont représentés sur la figure ci-dessous. Ils sont explicités ci-après.

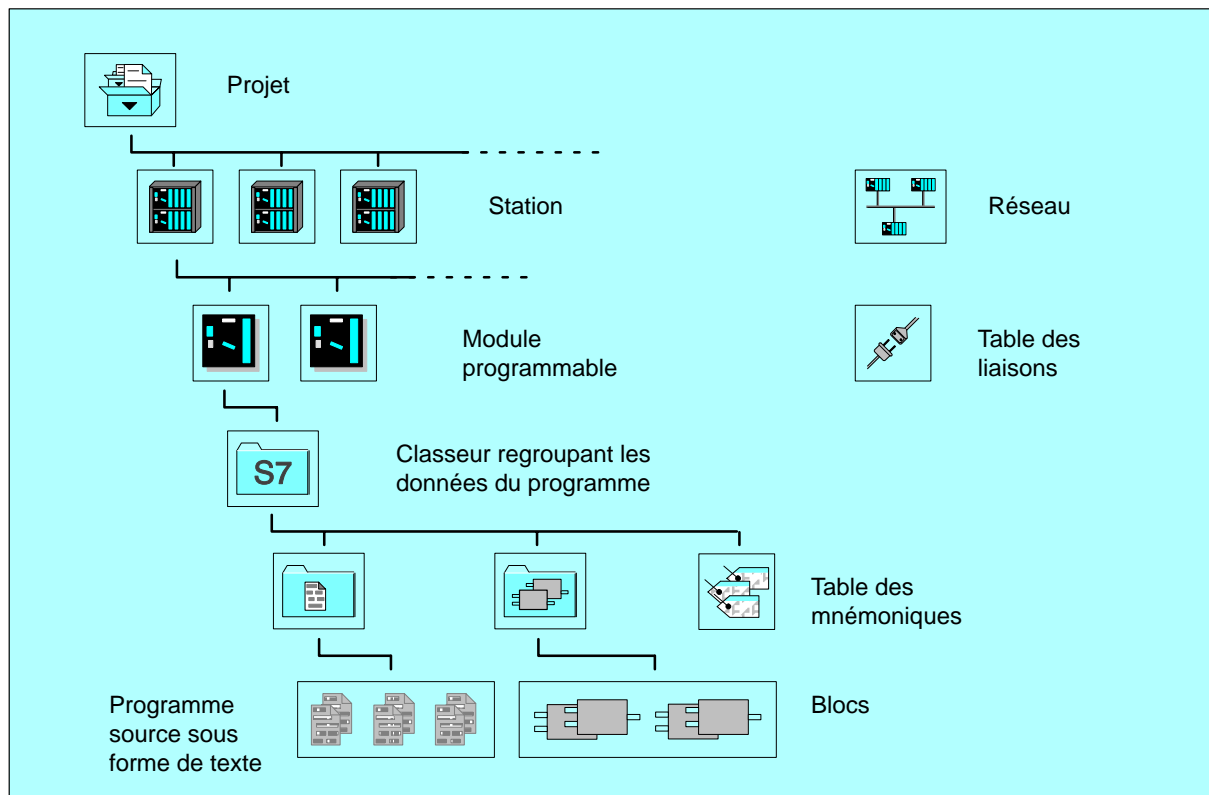


Figure 3-2 Structure hiérarchique des objets du projet STEP 7

Réseau

Objet renfermant les paramètres de réseau – MPI ou PROFIBUS –. Ceci permet la vérification des paramètres de communication par le programme dès qu’une station ou un module de celle-ci est connecté au réseau.

Station

La station représente la configuration d’un système d’automatisation avec ses châssis. Lorsque vous enfichez un module avec interface DP dans une station, le réseau maître DP qui part de la station fait partie avec tous ses esclaves de la station !

Une station peut comprendre un ou plusieurs modules programmables (CPU).

Matériel

Objet renfermant les données de configuration et les paramètres d’une station. Les données de configuration et les paramètres de la station sont sauvegardés dans les blocs de données système (SDB).

Module programmable

Les modules programmables sont à l’opposé des autres modules les modules porteurs des programmes utilisateur. Sous les modules programmables vous trouvez des dossiers – appelés dans STEP 7 des “classeurs” – contenant toutes les données du programme pour ce module :

- Programmes-sources sous forme de texte (créés à l’aide d’un éditeur de texte)

Des blocs exécutables sont générés à partir de ces programmes-sources et stockés en fin de compilation dans le classeur des blocs.

- Blocs (pouvant être chargés dans le module programmable)
- Table des mnémoniques

Table des liaisons

La table des liaisons représente la totalité des liaisons d'un module programmable (par exemple d'une CPU) au sein d'une station. Une liaison définit les propriétés de la communication entre deux partenaires de communication et est caractérisée par une ID de liaison. Celle-ci suffit pour programmer une communication déclenchée par événement à l'aide de blocs de communication standardisés qui peuvent être comparés aux blocs de dialogue de STEP 5.

Sources

Les sources servent en programmation S7 à la génération des blocs. Elles ne peuvent pas être chargées dans une CPU S7.

Blocs

Les blocs sont des parties du programme utilisateur délimités de par leur fonction, leur structure ou leur utilisation. Il est possible de charger les blocs dans des CPU S7.

Outre les blocs exécutables, vous trouvez dans le classeur des blocs les tables des variables.

Table des mnémoniques

Vous affectez dans la table des mnémoniques des noms (c'est-à-dire des mnémoniques) aux entrées, sorties, mémentos et blocs.

3.3 Edition du projet dans SIMATIC Manager

3.3.1 Création du projet

Nouveau projet

Pour créer un projet, procédez comme suit :

1. Choisissez la commande de menu **Fichier ► Nouveau** dans SIMATIC Manager.
2. Sélectionnez dans la boîte de dialogue "Nouveau" l'option "Nouveau projet".
3. Inscrivez le nom du projet et confirmez avec "OK".

Flexibilité pour la suite

Vous disposez de la plus grande flexibilité pour la suite de l'élaboration de votre projet. Vous pouvez ainsi commencer par

- configurer le matériel puis concevoir ensuite votre logiciel ou
- commencer à programmer le logiciel sans tenir compte de la configuration matérielle. La saisie de programmes est en effet indépendante de la configuration matérielle de la station.

Tableau 3-1 Alternatives

Alternative 1	Alternative 2
Si vous voulez configurer d'abord le matériel, (référez-vous au paragraphe 3.4)	Si vous voulez commencer par la conception du logiciel,
Configurez votre matériel comme décrit au paragraphe 3.4.	
La configuration une fois achevée, les classeurs nécessaires à la programmation sont déjà insérés dans le classeur "Programme S7".	Insérez un classeur de programme S7 (Programme S7) dans votre projet comme décrit au paragraphe 3.6.
Concevez ensuite le logiciel de vos modules programmables comme décrit au paragraphe 3.6.	Concevez ensuite le logiciel de vos modules programmables comme décrit au paragraphe 3.6.
	Configurez votre matériel comme décrit au paragraphe 3.4.
	Affectez ensuite le programme S7 à une CPU.

Pour en savoir plus sur la manière de charger et tester des programmes sans configuration matérielle, référez-vous au guide de l'utilisateur de STEP 7 **/231/** .

3.3.2 Sauvegarde des projets

Présentation

Pour sauvegarder un projet, vous pouvez le copier et l'enregistrer sous un autre nom ou archiver le projet.

Enregistrer sous...

Pour enregistrer un projet sous un autre nom, procédez comme suit :

1. Ouvrez le projet.
2. Choisissez la commande **Enregistrer sous** du menu **Fichier**. La boîte de dialogue "Enregistrer sous" apparaît.
3. Choisissez Enregistrer avec ou Enregistrer sans vérification de cohérence et fermez la boîte de dialogue avec "OK". La boîte de dialogue "Enregistrer le projet sous" s'affiche.
4. Dans la zone "Enregistrer dans", sélectionnez le répertoire dans lequel vous souhaitez enregistrer le projet.
5. Dans la zone de saisie "Nom de fichier", remplacez le caractère (*) par un nom de fichier. Ne modifiez pas l'extension du nom.
6. Quittez la boîte de dialogue par "OK".

Nota

Vous devez vous assurer que la capacité mémoire disponible sur le lecteur sélectionné est suffisante. Ainsi, il est par exemple inutile de sélectionner un lecteur de disquettes, puisque sa capacité est en principe insuffisante pour qu'un projet puisse y être enregistré. Pour pouvoir enregistrer des projets sur des disquettes, vous devez préalablement les archiver. Les archives peuvent ensuite être réparties sur plusieurs disquettes.

Archivage

Vous pouvez mémoriser un projet ou une bibliothèque particuliers sous forme comprimée dans un fichier d'archivage. Cette sauvegarde comprimée peut être réalisée sur un disque dur ou sur un support de données transportable (disquette).

Pour pouvoir accéder à des composantes d'un projet ou d'une bibliothèque archivés, vous devez préalablement désarchiver le projet. La fonction d'archivage de STEP 7 est décrite en détail dans le guide de l'utilisateur **/231/**.

3.4 Configuration matérielle dans STEP 7

Il n'était pas possible de configurer le matériel par logiciel dans SIMATIC S5. Une seule application de STEP 7 se charge maintenant de l'adressage et du paramétrage des modules et de la configuration de la communication. L'utilisateur n'a donc plus besoin de régler les adresses sur les modules et peut désormais s'acquitter confortablement de cette tâche autrefois pénible depuis son PC ou sa PG.

Condition préalable

Un projet doit avoir été créé.

Insérer une station

Pour insérer une nouvelle station dans un projet, ouvrez le projet pour afficher la fenêtre de projet si ce n'est déjà fait.

- Sélectionnez le projet.
- Générez l'objet correspondant au matériel souhaité au moyen de la commande **Insertion > Station**.

Sélectionnez ensuite votre station dans le sous-menu :

- Station SIMATIC 300
- Station SIMATIC 400
- PC/PG
- SIMATIC S5
- Autres stations, c'est-à-dire ni SIMATIC S7/M7, ni SIMATIC S5

Les stations PC/PG, SIMATIC S5 et autres stations ne peuvent être sélectionnées que pour la configuration des liaisons. La configuration ou programmation de SIMATIC S5 n'est pas possible dans STEP 7.

Cliquez sur le signe "+" précédant l'icône du projet dans la fenêtre de projet pour faire apparaître la station, si celle-ci n'est pas visible.

**Comment
procéder ?**

Procédez ensuite comme suit :

- Cliquez sur la station que vous venez d'insérer. Elle contient l'objet "Matériel".
- Cliquez sur celui-ci pour l'ouvrir. La fenêtre "HW Config" de l'application de configuration matérielle s'affiche.
- Définissez la configuration matérielle de votre station en sélectionnant par glisser-lâcher les modules dans le catalogue des modules. Pour l'afficher, choisissez la commande de menu **Affichage > Catalogue**.
- Commencez par insérer un châssis (Rack) dans la fenêtre vide. Sélectionnez ensuite vos modules que vous enfichez aux emplacements prévus dans le châssis. Vous devez configurer au moins une CPU par station. Vos saisies sont automatiquement vérifiées par le programme qui vous avertit toutes les fois qu'une action n'est pas autorisée.

Pour plus d'informations sur la configuration matérielle, veuillez vous référer au guide de l'utilisateur de STEP 7 **/231/**.

**Résultat de la
configuration**

La configuration matérielle terminée et sauvegardée, le programme génère automatiquement pour chaque CPU configurée une table des liaisons (objet "Liaisons") et un programme S7. Le programme S7 contient les objets "Sources" et "Blocs" (classeurs contenant le logiciel) et la table des mnémoniques.

Le classeur "Blocs" contient à son tour d'autres objets, à savoir l'OB1 et les "Données système" dans lesquelles sont enregistrées les données de la configuration après compilation.

3.5 Configuration des liaisons dans la table des liaisons

Les liaisons sont configurées dans S5 avec COM NCM. Il existe un progiciel COM pour chaque CP. Dans S7, vous configurez les liaisons dans la table des liaisons.

Présentation

Si vous voulez pouvoir utiliser les fonctions de communication par blocs fonctionnels dans votre programme utilisateur, vous devez au préalable avoir configuré les liaisons dans la table de configuration des liaisons.

Une liaison existe lorsque vous avez défini :

- les partenaires de communication à l'intérieur du projet S7,
- le type de la liaison (par exemple "homogène S7", FDL etc.),
- et des propriétés supplémentaires concernant l'initialisation de la liaison ou l'émission de messages d'état de fonctionnement.

Une ID locale univoque est attribuée lors de la configuration des liaisons à chaque liaison. Seule cette ID est requise lors du paramétrage de la fonction de communication.

Chaque CPU point terminal d'une liaison possède sa propre table des liaisons.

Particularité

Si les deux partenaires de communication sont des stations S7-400, une ID locale est affectée automatiquement aux deux points terminaux de la liaison. Si le partenaire de la liaison est une station S7-300, une seule ID locale sera générée sur la station S7-400.

Charger les données de configuration

Les données locales de configuration des points terminaux de liaison d'une station S7 doivent être chargés de manière explicite dans chaque station cible.

Une table des liaisons (vide) (objet "Liaisons") est automatiquement générée pour chaque CPU. Celle-ci sert à définir les liaisons de communication entre les CPU d'un réseau. Un double clic sur cet objet fait s'afficher une fenêtre avec une table dans laquelle vous définissez les liaisons entre modules programmables. Pour en savoir plus sur la configuration de ces liaisons, veuillez vous référer au guide de l'utilisateur de STEP 7 [\[231\]](#).

Exemple de liaison à un S5

L'exemple vous montre comment configurer une liaison à une station SIMATIC S5. Vous devez pour cela avoir inséré une station SIMATIC 400 dans votre projet.

- Insérez une station SIMATIC S5 dans votre projet et définissez les propriétés de la station.
- Ouvrez la table des liaisons de la station S7 et insérez une liaison avec la commande **Insertion > Liaison**. Ceci ouvre une boîte de dialogue dans laquelle vous pouvez entrer le partenaire de communication – dans ce cas la station SIMATIC S5 – et le type de liaison.
- Une fois que vous avez entré ces informations, la nouvelle liaison apparaît aussitôt dans la table des liaisons. Les propriétés de la liaison doivent encore être inscrites dans S5 pour la station S5 dans le COM NCM correspondant.

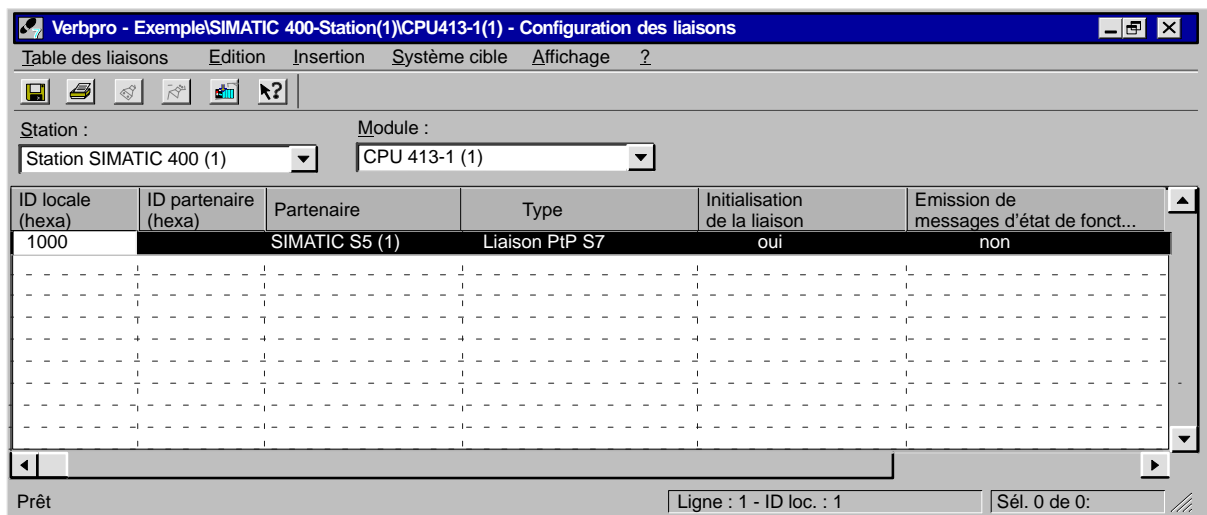


Figure 3-3 Table des liaisons

3.6 Insertion et édition du programme

Nous décrivons dans ce paragraphe suivant comment créer un nouveau programme.

3.6.1 Principe de la conception

Présentation

Le logiciel destiné aux CPU est stocké dans des classeurs de programme. Quand il est destiné à des modules SIMATIC S7, ce classeur s'appelle un "Programme S7".

Vous pouvez voir sur la figure ci-dessous un programme S7 dans la CPU d'une station SIMATIC 300.

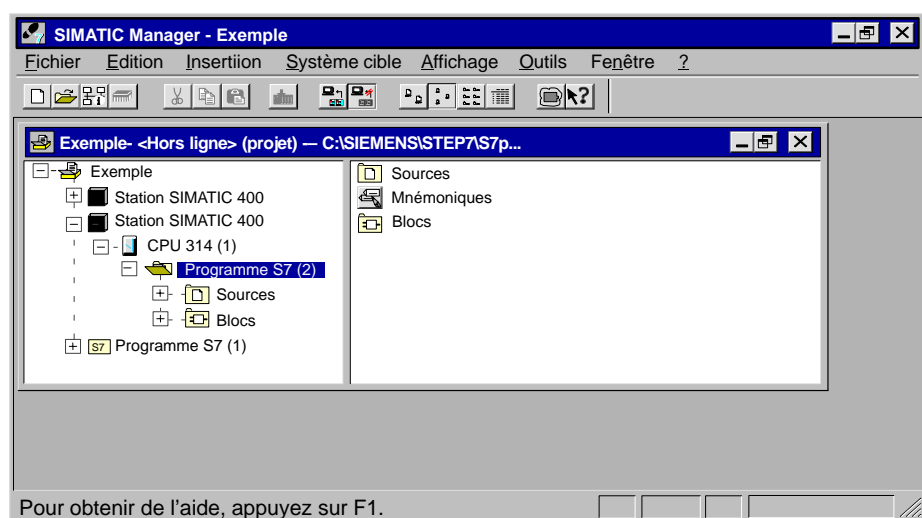


Figure 3-4 Programme ouvert dans SIMATIC Manager

Marche à suivre

Pour concevoir le logiciel de votre projet, procédez comme suit :

- Ouvrez le programme S7.
- Opérez dans celui-ci un double clic sur l'objet "Mnémoniques" et définissez vos mnémoniques pour le projet dans la table qui s'y ouvre. Vous pouvez aussi le faire à tout moment ultérieur. Pour en savoir plus sur la table des mnémoniques, reportez-vous au paragraphe 3.13.2.
- Ouvrez le classeur des blocs, si vous choisissez d'abord de créer vos blocs ou le classeur des sources si vous choisissez d'écrire votre programme sous la forme d'un document-texte source.
- Insérez un bloc ou une source (se reporter pour plus de détails au paragraphe 3.6.2). Les commandes de menu correspondantes sont les suivantes :
 - **Insertion > Logiciel S7 > Bloc...**, ou
 - **Insertion > Logiciel S7 > Source**
- Ouvrez le bloc ou la source et saisissez-y un programme. Cette procédure est décrite en détails dans les manuels de programmation /232/-/236/.

Selon l'application, il n'est pas nécessaire d'exécuter toutes ces étapes.

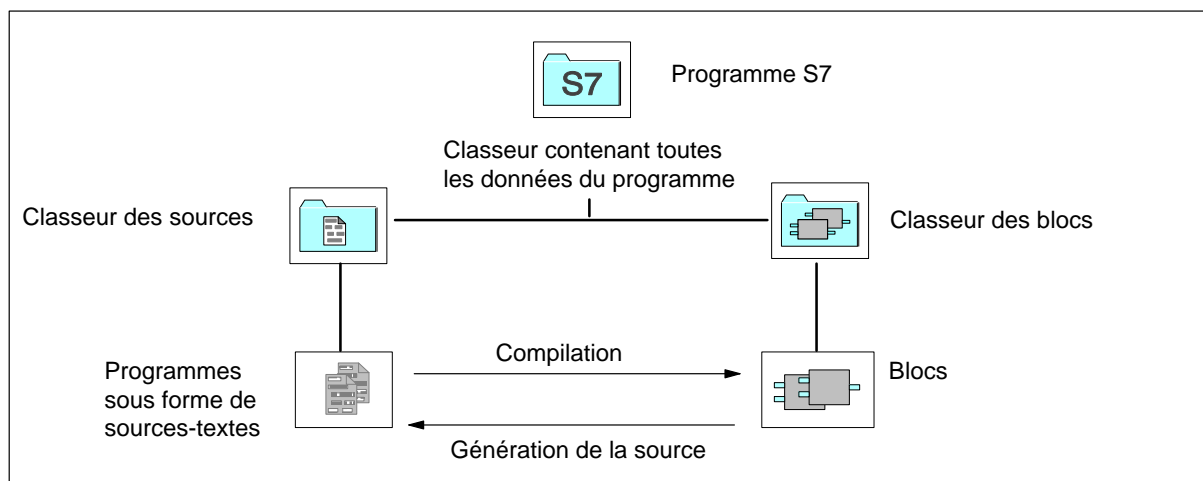


Figure 3-5 Structure hiérarchique des objets du projet STEP 7

3.6.2 Insertion de composants

Composants standard

Un programme S7/M7 est automatiquement généré comme classeur du logiciel pour chaque module programmable.

Dans un programme S7 figurent déjà :

- une table des mnémoniques (objet "Mnémoniques"),
- un classeur "Blocs" pour les blocs, contenant le premier bloc, l'OB1,
- un classeur "Sources" pour des programmes source.

Créer des blocs S7

Pour écrire des programmes LIST, CONT ou LOG, vous sélectionnez l'objet "Blocs" déjà créé et choisissez ensuite la commande **Insertion > Logiciel S7 > Bloc**. Le menu suivant vous permet de choisir le type de bloc : par exemple, bloc de données, type de données utilisateur (UDT), fonction, bloc fonctionnel, bloc d'organisation, table des variables (VAT).

Vous saisissez votre programme LIST, CONT ou LOG dans le bloc (vide) qui s'ouvre alors. Vous trouverez plus d'informations à ce sujet dans les manuels relatifs à LIST /232/, CONT /233/ et LOG /236/.

L'objet Données système (SDB), que vous trouverez éventuellement dans des programmes utilisateur, est généré par le système. Vous pouvez l'ouvrir, mais vous ne pouvez pas en modifier le contenu pour des raisons de cohérence. Il sert à modifier la configuration après le chargement d'un programme et à charger ces modifications dans le système cible.

Prélever des blocs dans des bibliothèques

Vous pouvez également créer vos programmes utilisateur en prélevant des blocs pré-programmés dans des bibliothèques de blocs standard faisant partie du logiciel. Vous accédez aux bibliothèques par la commande **Fichier > Ouvrir**. Vous trouverez des renseignements complémentaires sur l'utilisation des bibliothèques standard ainsi que sur la création de vos propres bibliothèques dans l'aide en ligne.

Créer des sources

Pour créer une source en langage LIST, sélectionnez dans le programme S7 l'objet "Sources" ou "Diagrammes" si cette source doit servir à créer un diagramme dans CFC et activez ensuite la commande **Insertion > Logiciel S7 > Source**. Le menu suivant vous permet de choisir la source correspondant au langage de programmation. Vous pouvez saisir le programme une fois la source vide ouverte.

Créer une table des mnémoniques

Une table de mnémoniques vide (objet "Mnémoniques") est automatiquement générée lors de la création d'un programme S7. Son ouverture entraîne également celle de la fenêtre de l'éditeur de mnémoniques et l'affichage de la table de mnémoniques qu'elle contient (voir paragraphe 3.13.2 *Adressage symbolique*).

Insérer une source externe

Vous pouvez créer et éditer des fichiers source avec des éditeurs ASCII quelconques. Il est ensuite possible d'importer ces fichiers dans un projet et de les compiler en blocs exécutables. Procédez pour cela de la manière suivante :

1. Sélectionnez le classeur des sources dans lequel vous désirez importer le fichier source.
2. Choisissez ensuite la commande de menu **Insertion > Source externe**.
3. Précisez le fichier source dans la boîte de dialogue qui s'ouvre alors.

Les blocs générés lors de la compilation d'une source importée sont stockés dans le classeur des blocs.

3.7 Blocs

3.7.1 Mise en parallèle

Le tableau suivant énumère les blocs STEP 5 et les blocs STEP 7 en parallèle montrant à chaque fois la correspondance entre ceux-ci et indique quand un bloc S5 doit être remplacé par un bloc ou une fonction STEP 7 équivalente.

Liberté d'association

Les possibilités qu'offre le nouvel univers de blocs ne pouvant toutes être traitées ici, ce tableau a uniquement valeur de suggestion et est destiné à vous aider dans votre choix de blocs STEP 7.

Tableau 3-2 Mise en parallèle des blocs STEP 5 et blocs STEP 7

Bloc STEP 5	Bloc STEP 7	Explication
Bloc d'organisation (OB)	Bloc d'organisation (OB)	Interface au système d'exploitation
OB spéciaux intégrés	Fonctions système (SFC) Blocs fonctionnels système (SFB)	Les fonctions système de STEP 7 remplacent les blocs d'organisation spéciaux de STEP 5 pouvant être appelés dans le programme utilisateur.
Bloc fonctionnel (FB, FX)	Fonction (FC)	Les fonctions (FC) de STEP 7 ont les mêmes propriétés que les blocs fonctionnels de STEP 5.
Bloc de programme (PB)	Bloc fonctionnel (FB)	Les blocs de programme ont leur équivalent dans STEP 7 : on les appelle des blocs fonctionnels. A la différence de leurs homologues dans STEP 5, ils possèdent des propriétés nouvelles et ouvrent de nouvelles perspectives en matière de programmation. Attention : les blocs de programme sont convertis en fonctions STEP 7 (FC).
Bloc séquentiel (SB)	-	Les blocs séquentiels n'existent plus dans STEP 7.
Bloc de données (DB, DX)	Bloc de données (DB)	Les blocs de données sont plus longs dans STEP 7 que leurs équivalents dans STEP 5 (jusqu'à 8 kilo-octets pour le S7-300, et jusqu'à 64 kilo-octets pour le S7-400)
Blocs de données DX0, DB1 dans leur fonction spéciale	Blocs de données système (SDB) (paramétrage CPU)	Les nouveaux blocs de données système renferment toutes les données de configuration matérielle ainsi que les paramétrages de CPU nécessaires à l'exécution du programme.
Blocs de commentaire DK, DKX, FK, FKX, PK	-	Les blocs de commentaire n'existent plus dans STEP 7. Le commentaire est contenu dans les blocs correspondants dans la base de données hors ligne.

3.7.2 Fonctions et blocs fonctionnels

Fonctions (FC)

Une fonction (FC) est un bloc de code "sans mémoire" dont les paramètres de sortie affichent en fin d'exécution les valeurs qu'elle a calculées. Le traitement ultérieur et la sauvegarde de ces résultats doivent donc être considérés par l'utilisateur lorsqu'il programme l'appel de la fonction.

Ne confondez pas "fonctions" et "blocs fonctionnels". En STEP 7, il s'agit de deux types de blocs distincts.

Blocs fonctionnels (FB)

Un bloc fonctionnel (FB) est un bloc de code "avec mémoire". Lui sert de mémoire un bloc de données d'instance qui lui est associé et dans lequel les paramètres effectifs et les données statiques du bloc fonctionnel sont stockés.

Les blocs fonctionnels sont par exemple utilisés lorsqu'il s'agit de programmer des régulateurs.

3.7.3 Blocs de données

Les blocs de données servent à stocker les données du programme utilisateur. On fait la distinction entre les blocs de données globaux et les blocs de données d'instance :

- Les blocs de données globaux ne sont pas affectés à un bloc précis (comme dans STEP 5).
- Les blocs de données d'instance sont associés à un bloc fonctionnel et peuvent contenir en plus des données de ce FB les données de multi-instances que l'on aura éventuellement définies.

Un bloc de données est soit un bloc de données global, soit un bloc de données d'instance.

3.7.4 Blocs système

Fonctions et blocs fonctionnels système (SFC et SFB)

Il n'est pas nécessaire de programmer chaque fonction. Vous pouvez recourir à des blocs préprogrammés intégrés au système d'exploitation des unités centrales, par exemple pour programmer les fonctions de communication. Il s'agit des blocs suivants :

- **Fonctions système (SFC)**, possédant les mêmes propriétés que les fonctions (FC),
- **Blocs fonctionnels système (SFB)**, possédant les mêmes propriétés que les blocs fonctionnels (FB).

Blocs de données système (SDB)

Nous avons jusqu'ici parlé de blocs renfermant le code ou les données du programme utilisateur. Il existe à côté de ces blocs des blocs qui servent à stocker les adresses ou les paramètres des modules. On les appelle des **blocs de données système (SDB)**. Les blocs de données système sont générés par des applications spécifiques de STEP 7, par exemple lors de la configuration matérielle ou à la création des tables de liaisons.

3.7.5 Blocs d'organisation

Les blocs d'organisation (OB) servent d'interface entre le système d'exploitation et le programme utilisateur. Ces blocs d'organisation remplissent des tâches différentes bien précises.

Classification des OB

Vous écrivez votre programme utilisateur LIST pour la CPU S7 en sélectionnant les blocs d'organisation (OB) dont vous avez besoin pour votre automatisme.

Tableau 3-3 Blocs d'organisation dans S5 et dans S7

Fonction		S5	S7
Programme principal	Cycle libre	OB1	OB1
Alarmes	Alarme temporisée (alarme de retardement)	OB6	OB20 à OB23
	Alarme horaire (alarme d'horloge à heure fixe)	OB9	OB10 à OB17
	Interruptions	OB2 à OB5	OB40 à OB47
	Alarmes de processus	OB2 à OB9 (EB 0)	remplacées par des interruptions
	Alarmes cycliques (alarmes d'horloge)	OB10 à OB18	OB30 à OB38
	Alarme multiprocesseur	-	OB60
Mise en route	Démarrage manuel	OB21 (S5-115U) OB20 (à partir de S5-135U)	OB100
	Redémarrage manuel	OB21 (à partir de S5-135U)	OB101
	Redémarrage automatique	OB22	OB101
Erreurs	Erreurs	OB19 à OB35	OB121, OB122, OB80 à OB87
Autres	Traitement à l'état de STOP	OB39	n'est plus possible
	Traitement en arrière-plan	-	OB90

OB d'erreur

Les OB d'erreur sont appelés quand une erreur survient dans le déroulement du programme. Ils vous permettent de programmer des réactions à l'erreur. Lorsqu'il n'existe pas d'OB pour un type d'erreur, la CPU passe à l'arrêt.

Tableau 3-4 Traitement des erreurs dans S5 et dans S7

Fonction	S5	S7
Appel d'un bloc non chargé	OB19	OB121
Retard d'acquiescement pour accès direct à un module périphérique	OB23	OB122
Retard d'acquiescement pour actualisation de la mémoire image et des mémentos de couplage	OB24	OB122
Erreur d'adressage	OB25	OB122
Dépassement du temps de cycle	OB26	OB80
Erreur de substitution	OB27	–
Arrêt par manipulation	OB28 (S5-135U)	–
Retard d'acquiescement pour octet d'entrée EB 0	OB28 (S5-155U)	OB85
Code d'opération illicite	OB29 (S5-135U)	STOP
Retard d'acquiescement pour accès direct à la périphérie dans la plage d'adressage étendue	OB29 (S5-155U)	OB122
Paramètre illicite	OB30 (S5-135U)	–
Erreur de parité ou retard d'acquiescement pour accès à la mémoire utilisateur	OB30 (S5-155U)	OB122
Erreur groupée de fonction spéciale	OB31	–
Erreur de transfert pour bloc de données	OB32	OB121
Erreur d'horloge pour traitement déclenché par horloge	OB33	OB80
Erreur de régulateur	OB34 (S5-135U)	–
Erreur à la création d'un bloc de données	OB34 (S5-155U)	Message en retour de la SFC
Erreur d'interface	OB35	OB84

Débordement de zone

Il est possible dans STEP 7 comme dans STEP 5 d'évaluer les bits d'état DB et DM signalant un débordement de zone. La fonction de ces bits d'état n'a pas changé si ce n'est à quelques différences près.

Vous les trouvez décrites dans le manuel LIST /232/ avec leurs instructions correspondantes.

Fonctions spéciales intégrées

Dans S5, le programme utilisateur et le programme système de la CPU communiquaient via des accès à la zone de mémoire système et le recours à des OB spéciaux servant d'interface.

Dans S7, de nouveaux types de blocs – les **fonctions et blocs fonctionnels système** – remplissent ces fonctions à côté des blocs d'organisation.

Fonctions et blocs fonctionnels système

Les fonctions système (**SFC**) et les blocs fonctionnels système (**SFB**) sont des blocs intégrés au système d'exploitation de la CPU qui peuvent être appelés, lorsque cela est nécessaire, dans le programme utilisateur de STEP 7. Lorsqu'une erreur survient à l'exécution d'une fonction système (SFC), une valeur en retour RET_VAL permet d'analyser l'erreur dans le programme utilisateur.

Tableau 3-5 Fonctions spéciales dans S5 et dans S7

Fonction	Bloc de S5	Equivalent dans S7
Déclenchement du temps de cycle	OB31	SFC43 RE_TRIGR
Pile défaillante	OB34	OB81 (réaction à l'erreur programmable par l'utilisateur)
Accès à l'octet indicateur	OB110	Commande STEP 7 : L STW/T STW
Effacement des accumulateurs 1 à 4	OB111	Séquence d'instructions STEP 7 : L 0; PUSH; PUSH; PUSH
Décalage vers le haut du contenu des accumulateurs	OB112	Fonction modifiée : Commande STEP 7 : PUSH
Décalage vers le bas du contenu des accumulateurs	OB113	Fonction modifiée : Commande STEP 7 : POP
Activation/désactivation de l'inhibition collective des alarmes	OB120	SFC41 DIS_AIRT SFC42 EN_AIRT
Activation/désactivation de l'inhibition sélective des alarmes cycliques	OB121	SFC39 DIS_IRT SFC40 EN_IRT
Activation/désactivation de l'ajournement collectif des alarmes	OB122	SFC41 DIS_AIRT SFC42 EN_AIRT
Activation/désactivation de l'ajournement sélectif des alarmes cycliques	OB123	SFC39 DIS_IRT SFC40 EN_IRT
Réglage et lecture de l'horodateur interne	OB150	SFC0 SET_CLK SFC1 READ_CLK
Réglage et lecture de l'heure de déclenchement d'une alarme à heure fixe	OB151	SFC28 SET_TINT SFC30 ACT_TINT SFC31 QRY_TINT

Tableau 3-5 Fonctions spéciales dans S5 et dans S7 (suite)

Fonction	Bloc de S5	Equivalent dans S7
Statistiques de cycle	OB152	Données locales dans OB1
Compteur de boucles	OB160 - 163 (S5-135U)	Instruction de S7 : LOOP
Boucle de temps variable	OB160 (S5-115U)	SFC47 WAIT
Lecture de la pile des blocs	OB170	–
Accès variable aux blocs de données	OB180	–
Test de bloc de données	OB181	SFC24 TEST_DB
Duplication d'une zone de données	OB182	SFC20 BLKMOV
Transfert de mémentos dans des blocs de données	OB190, 192	SFC20 BLKMOV
Transfert de paquets de données dans des zones de mémentos	OB191, 193	SFC20 BLKMOV
Fonctions pour la communication multiprocesseur	OB200 - 205	–
Accès aux pages	OB216 - 218	Pas d'adressage de page dans S7.
Extension de signe	OB220	Instruction de S7 : ITD
Définition d'un nouveau temps de cycle maximal	OB221	Paramétrage à l'aide de S7
Redémarrage de la surveillance du temps de cycle	OB222	SFC43 RE_TRIGR
Comparaison des modes de mise en route	OB223	Mise en route multiprocesseur uniquement en même mode
Transfert en bloc des mémentos de couplage	OB224	–
Lecture du contenu d'une cellule mémoire du programme système	OB226	–
Lecture du total de contrôle du programme système	OB227	–
Lecture de l'information d'état concernant un niveau d'exécution du programme	OB228	SFC51 RDSYSST SFC6 RD_SINFO
Fonctions pour blocs de dialogue	OB230 à 237	Communication par blocs SFB
Initialisation d'un registre à décalage	OB240	–
Traitement d'un registre à décalage	OB241	–
Effacement d'un registre à décalage (suite à la page suivante)	OB242	–

Tableau 3-5 Fonctions spéciales dans S5 et dans S7 (suite)

Fonction	Bloc de S5	Equivalent dans S7
Régulation : initialisation de l'algorithme PID Régulation : traitement de l'algorithme PID	OB250 OB251	FB de régulation : FB41 à FB43 ou SFB41 à SFB43
Transfert de blocs de données (DB/DX) dans la mémoire vive de DB	OB254, 255	—

3.7.6 Transposition des blocs à la conversion

Correspondance des blocs

La structure des blocs a été modifiée dans S7. Vous voyez sur la figure ci-dessous la correspondance des blocs de STEP 5 et STEP 7. A la conversion, les blocs de STEP 5 sont remplacés de façon analogue par les blocs de STEP 7.

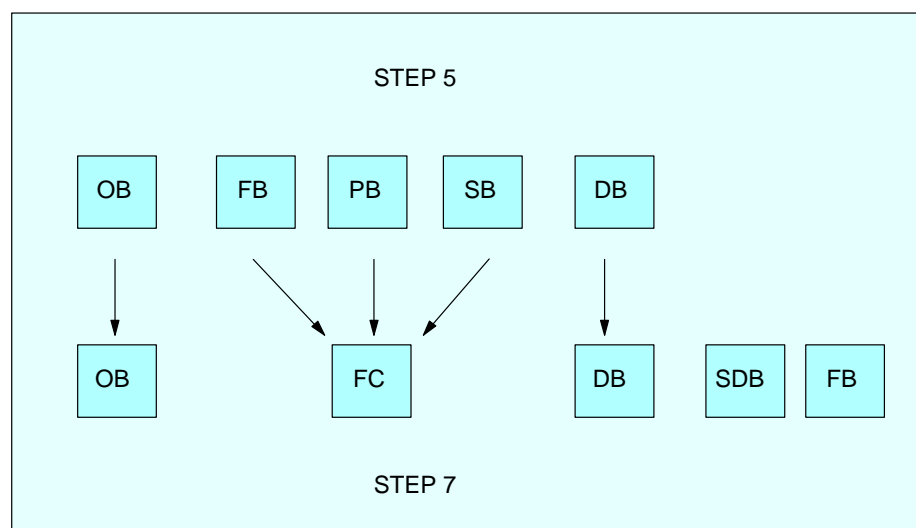


Figure 3-6 Blocs S5 et S7 remplissant des fonctions similaires

Le tableau 3-6 de la page 3-25 montre les appels de blocs après leur conversion.

Tableau 3-6 Types de blocs dans S5 et S7

S5			S7	
OB	Numéros fixes	Programme utilisateur	OB lui correspondant	Numéros fixes
OB	Numéros fixes	Fonctions spéciales	Non convertible. Doivent être reprogrammées dans S7.	
PB	0 à 255	Programme utilisateur	FC sans paramètres	Un numéro vous est proposé.
FB/FX	0 à 255	Programme utilisateur	FC avec paramètres dont le nom est conservé	Un numéro vous est proposé.
FB	Numéros fixes	Blocs fonctionnels intégrés	FC chargeables contenues dans la bibliothèque FBLib1 et devant être chargées dans le fichier converti avant la compilation	Numéros fixes
FB/FX	Noms fixes	Blocs fonctionnels standard	FC chargeables contenues dans la bibliothèque FBLib1 et devant être chargées dans le fichier converti avant la compilation	Numéros fixes
SB	0 à 255	Programme utilisateur	FC sans paramètres Les graphes séquentiels ne sont pas convertibles et doivent être programmés à l'aide de l'application GRAPH pour S7	Un numéro vous est proposé.
DB	2 à 255	Données utilisateur	Blocs de données DB globaux	Le numéro de S5 est repris.
DX	1 à 255	Données utilisateur	Blocs de données DB globaux	Un numéro vous est proposé à partir de 256.
DB1/ DX0		Blocs de données avec paramètres système	Si ces blocs renferment des informations spécifiques à la CPU, il faut effectuer les paramétrages correspondants dans STEP 7. Le contenu des blocs DB1 et DX0 après conversion est sans intérêt et peut être effacé.	

3.8 Paramètres système

Transposition de DB1 et de DX0

Les tableaux 3-7 et 3-8 montrent comment les fonctions des paramètres contenus dans DB1 et dans DX0 (paramètres système) sont réalisées dans S7.

Tableau 3-7 Transposition des paramètres système de DB1

Bloc de paramètres de S5	Réalisé comme suit dans S7
Ajournement de la mise en route	Appel de SFC47 WAIT
Mémentos de couplage	Remplacés par la communication par données globales Appel des fonctions système : SFC60 GD_SND SFC61 GD_RCV
Position du code d'erreur	Le système dépose les messages d'erreur dans la mémoire tampon de diagnostic. L'indication "Position du code d'erreur" n'existe plus.
Remplacement du numéro des blocs fonctionnels intégrés	Supprimé
Entrées analogiques internes	Paramétrés via les propriétés de la CPU (HW Config)
Interruptions internes	Paramétrés via les propriétés de la CPU (HW Config)
Compteurs internes	Paramétrés via les propriétés de la CPU (HW Config)
Modification de la priorité des blocs d'organisation	Paramétrés via les propriétés de la CPU (HW Config)
Exportation et inhibition de la mémoire image	Appel de SFC27 UPDAT_PO
Importation et inhibition de la mémoire image	Appel de SFC26 UPDAT_PI
Mémentos rémanents	Paramétrés via les propriétés de la CPU (HW Config)
Temporisations rémanentes	Paramétrés via les propriétés de la CPU (HW Config)
Compteurs rémanents	Paramétrés via les propriétés de la CPU (HW Config)
SINEC L1	Est remplacé par le bus MPI (communication par données globales)
SINEC L2	Paramétré dans HW Config
Protection du logiciel	En préparation
Paramètres d'horloge	Paramétrés dans HW Config via les propriétés de la CPU ou l'appel de SFC28 SET_TINT
Paramétrage des blocs d'organisation d'horloge	Paramétrés via les propriétés de la CPU (HW Config)
Surveillance du temps de cycle	Paramétrée via les propriétés de la CPU (HW Config)

Tableau 3-8 Transposition des paramètres système de DX0

Bloc de paramètres de S5	Réalisé comme suit dans S7
Surveillance des erreurs d'adressage	Appel de l'OB121
Actualisation des mémentos de couplage	Communication par données globales
Mode de mise en route après mise sous tension	Paramétré via les propriétés de la CPU (HW Config)
Synchronisation de mise en route en mode multiprocesseur	Paramétrée via les propriétés de la CPU (HW Config)
Nombre de cellules de temporisation	Valeur fixe propre à la CPU (pour S7-300) ou paramétrable via les propriétés de la CPU (pour S7-400)
Traitement des erreurs	Appel de : SFC36 MSK_FLT SFC37 DMSK_FLT
Arithmétique en virgule flottante	Réalisée
Déclenchement des alarmes de processus	Paramétré via les propriétés de la CPU (HW Config)
Mode de traitement des alarmes cycliques	Appel de SFC28 SET_TINT
Surveillance du temps de cycle	Paramétrée via les propriétés de la CPU (HW Config)

3.9 Fonctions standard

Les fonctions standard de S5 sont automatiquement remplacées à la conversion par des fonctions de même fonctionnalité. Ces fonctions se laissent facilement remplacer dans S7 par des séquences de commandes simples, permettant ainsi d'optimiser l'utilisation de la mémoire et le temps de cycle.

Vous trouvez ces fonctions standard dans le classeur de programmes FBLib1 de la bibliothèque S7 "StdLib30".

Vous trouverez plus d'informations sur l'utilisation des bibliothèques dans l'aide en ligne.

3.9.1 Arithmétique en virgule flottante

STEP 5	STEP 7		STEP 5	STEP 7	
Nom de FB	Numéro	Nom	Nom de FB	Numéro	Nom
GP:FPGP	FC61	GP_FPGP	GP:MUL	FC65	GP_MUL
GP:GPFP	FC62	GP_GPFP	GP:DIV	FC66	GP_DIV
GP:ADD	FC63	GP_ADD	GP:VGL	FC67	GP_VGL
GP:SUB	FC64	GP_SUB	RAD:GP	FC68	RAD_GP

3.9.2 Fonctions de signalisation

STEP 5	STEP 7		STEP 5	STEP 7	
Nom de FB	Numéro	Nom	Nom de FB	Numéro	Nom
MLD:TG	FC69	MLD_TG	MLD:EZ	FC75	MLD_EZ
MELD:TGZ	FC70	MELD_TGZ	MLD:ED	FC76	MLD_ED
MLD:EZW	FC71	MLD_EZW	MLD:EZWK	FC77	MLD_EZWK
MLD:EDW	FC72	MLD_EDW	MLD:EDWK	FC78	MLD_EDWK
MLD:SAMW	FC73	MLD_SAMW	MLD:EZK	FC79	MLD_EZK
MLD:SAM	FC74	MLD_SAM	MLD:EDK	FC80	MLD_EDK

3.9.3 Fonctions intégrées

STEP 5	STEP 7	
Nom de FB	Numéro	Nom
COD:B4	FC81	COD_B4
COD:16	FC82	COD_16
MUL:16	FC83	MUL_16
DIV:16	FC84	DIV_16

3.9.4 Fonctions de base

STEP 5	STEP 7		STEP 5	STEP 7	
Nom de FB	Numéro	Nom	Nom de FB	Numéro	Nom
ADD:32	FC85	ADD_32	REG:LIFO	FC93	REG_LIFO
SUB:32	FC86	SUB_32	DB:COPY	FC94	DB_COPY
MUL:32	FC87	MUL_32	DB:COPY	FC95	DB_COPY
DIV:32	FC88	DIV_32	RETTEN	FC96	RETTEN
RAD:16	FC89	RAD_16	LADEN	FC97	LADEN
REG:SCHB	FC90	REG_SCHB	COD:B8	FC98	COD_B8
REG:SCHW	FC91	REG_SCHW	COD:32	FC99	COD_32
REG:FIFO	FC92	REG_FIFO			

3.9.5 Fonctions analogiques

STEP 5	STEP 7		STEP 5	STEP 7	
Nom de FB	Numéro	Nom	Nom de FB	Numéro	Nom
AE:460	FC100	AE_460_1	AE:466	FC106	AE_466_1
AE:460	FC101	AE_460_2	AE:466	FC107	AE_466_2
AE:463	FC102	AE_463_1	RLG:AA	FC108	RLG_AA1
AE:463	FC103	AE_463_2	RLG:AA	FC109	RLG_AA2
AE:464	FC104	AE_464_1	PER:ET	FC110	PER_ET1
AE:464	FC105	AE_464_2	PER:ET	FC111	PER_ET2

3.9.6 Fonctions mathématiques

STEP 5	STEP 7		STEP 5	STEP 7	
Nom de FB	Numéro	Nom	Nom de FB	Numéro	Nom
SINUS	FC112	SINUS	ARCCOT	FC119	ARCCOT
COSINUS	FC113	COSINUS	LN X	FC120	LN_X
TANGENS	FC114	TANGENS	LG X	FC121	LG_X
COTANG	FC115	COTANG	B LOG X	FC122	B_LOG_X
ARCSIN	FC116	ARCSIN	E^X	FC123	E_H_N
ARCCOS	FC117	ARCCOS	ZEHN^X	FC124	ZEHN_H_N
ARCTAN	FC118	ARCTAN	A2^A1	FC125	A2_H_A1

3.10 Types de données

STEP 7 utilise de nouveaux types de données. Vous voyez les types de données S5 et S7 en parallèle dans le tableau ci-après.

Tableau 3-9 Types de données dans S5 et S7

Types de données dans S5	Types de données dans S7	Classe
BOOL, BYTE, WORD, DWORD, nombre entier 16 bits, nombre entier 32 bits, virgule flottante, valeur temporelle, - (Caractères ASCII)	BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, S5TIME, TIME, DATE; TIME_OF_DAY, CHAR	Types de données simples
-	DATE_AND_TIME, STRING, ARRAY, STRUCT	Types de données complexes
Temporisations, compteurs, blocs - -	TIMER, COUNTER, BLOCK_FC, BLOCK_FB, BLOCK_DB, BLOCK_SDB, POINTER, ANY	Types de paramètre

Tableau 3-10 Formats de constante dans S5 et dans S7

Formats dans S5	Exemple	Formats dans S7	Exemple
KB	L KB 10	3#16#	L B#16# A
KF	L KF 10	-	L 10
KH	L KH FFFF	W#16#	L W#16# FFFF
KM	L KM 1111111111111111	2#	L 2# 11111111_11111111
KY	L KY 10,12	B#	L B# (10,12)
KT	L KT 10.0	S5TIME# (S5T#)	L S5TIME# 100ms
KZ	L KZ 30	C#	L C#30
DH	L DH FFFF FFFF	DW#16#	L DW#16# FFFF_FFFF
KC	L KC WW	' xx '	L ' WW '
KG	L KG +234 +09	REAL	L +2.34 E+08
Représentation : format S5		Représentation : format simple selon ANSI/IEEE	
← exposant → ← mantisse →		V ← exposant → ← mantisse →	
31 30 24 23 22 0 <div>SE 2⁶.. ... 2⁰ SM 2⁻¹..... 2⁻²³</div>		31 30 23 22 0 <div>S 2⁷.. ... 2⁰ 2⁻¹.. 2⁻²³</div>	
Exposant = valeur absolue de l'exposant		Exposant = exposant réel + biais* (+127)	
SE = signe de l'exposant		S = signe de la mantisse	
SM = signe de la mantisse			
Valeurs admises : 1,5 x 10 ⁻³⁹ à 1,7 x 10 ³⁸		Valeurs admises : 1,18 environ x 10 ⁻³⁸ à 3,40 x 10 ⁺³⁸	

* Biais : plage de décalage séparant l'exposant dans la plage des nombres positifs et négatifs. La valeur 127 dans la plage des exposants correspond à la valeur absolue 0.

Pour plus d'informations au sujet des types de données, référez-vous au manuel
LIST/232/.

3.11 Zones d'opérande

3.11.1 Vue d'ensemble

Tableau 3-11 Opérandes dans S5 et S7

Zone d'opérande	Opérandes dans S5	Opérandes équivalents dans S7	Remarque
Entrées	E	E	
Sorties	A	A	
Périphérie	P, Q, G	PE → pour commandes de chargement	Pas de conversion de la périphérie globale
		PA → pour commandes de transfert	
Zone de mementos	M	M	
	S	M	à partir du M 256.0 (converti)
	“Mémentos banalisés”	L	Conversion identique à celle des mementos
Temporisations	T	T	
Compteurs	Z	Z	
Zone de données	D...	DB...	sont transformées en opérandes de DB globaux
Données système	BS, BT, BA, BB	-	ne sont pas convertis
Pages	C	-	

Remarque sur les opérandes au format données

Il existe deux registres de blocs de données dans S7 : le registre DB principalement utilisé pour les blocs de données globaux et le registre DI utilisé de préférence pour les DB d'instance. Il existe donc deux types d'opérandes au format donnée. Les opérandes DBX, DBB, DBW, DBD sont des opérandes provenant des blocs de données globaux, les opérandes DIX, DIB, DIW, DID sont des opérandes de DB d'instance. A la conversion, les opérandes de blocs de données D, DB, DW, DD sont remplacés par des opérandes de DB globaux.

Tenez également compte des informations relatives à la conversion des blocs de données au paragraphe 3.7.6.

Nota

Il est important de savoir que la taille et la plage de numéros des zones d'opérandes, tout comme le nombre et la longueur des blocs sont fonction de la CPU utilisée. Reportez-vous pour ceci au paragraphe 2.2.1 sur les caractéristiques des CPU.

3.11.2 Nouveaux opérandes dans S7 : les données locales

Données locales dans STEP 7

Les données locales sont dans STEP 7 les données affectées à un bloc de code et qui sont déclarées dans sa partie déclarative ou dans la déclaration des variables. Il peut s'agir selon le bloc des paramètres du blocs, de données statiques, de données temporaires etc. Les données locales sont en général adressées de manière symbolique.

Paramètres de blocs

Les paramètres des fonctions (FC) sont traitées comme des paramètres de blocs dans S5 : les paramètres de blocs sont des pointeurs renvoyant aux paramètres effectifs.

Les paramètres des blocs fonctionnels (FB) sont comme les données locales statiques sauvegardés dans le DB d'instance.

Données locales statiques

Des données locales statiques peuvent être utilisées dans chaque bloc fonctionnel. Elles sont définies dans une partie déclarative et sauvegardées dans le DB d'instance.

Comme les opérandes des DB globaux, les données locales statiques conservent leur valeur jusqu'à ce qu'elles soient écrasées dans le programme.

D'ordinaire, les données locales statiques ne sont traitées que dans le bloc fonctionnel. Comme elles sont toutefois sauvegardées dans un bloc de données, vous pouvez y accéder à tout moment dans le programme comme vous pouvez le faire avec les variables d'un bloc de données global.

Données locales temporaires

Mémentos banalisés dans STEP 5

Cinq zones de mémentos étaient utilisées dans STEP 5 pour stocker des résultats intermédiaires. Les mémentos 200 à 255 servaient en standard de mémoire intermédiaire. La tâche de gérer ces mémentos incombait entièrement à l'utilisateur.

Données locales temporaires dans STEP 7

Les données locales temporaires servent à stocker temporairement les données valables uniquement pour le temps que dure le traitement du bloc. La mémoire utilisée est libérée aussitôt après l'exécution de ce bloc. Chaque tâche ou OB possède sa pile de données locales. Ainsi, le risque d'écraser des résultats intermédiaires par des programmes d'alarme est exclu.

Utilisation de données locales temporaires dans STEP 7

Les trois champs d'application des variables temporaires de STEP 7 sont les suivants :

- Mémoire tampon pour les données du programme utilisateur.

Ce cas d'application est expliqué plus haut et est valable pour les blocs FC, FB et OB.

- Mémoire servant à la transmission d'informations du système d'exploitation au programme utilisateur.

Les informations qu'envoie le système d'exploitation au programme utilisateur portent un nom. On les appelle les "informations de déclenchement". La transmission des informations de déclenchement est assurée par les blocs d'organisation assumant le rôle d'interface.

- Transmission des paramètres pour les FC.

Où déclare-t-on les données locales ?

Vous déclarez les données locales dans le bloc. Lorsque vous créez un nouveau bloc, vous commencez par définir des mnémoniques pour vos variables temporaires, puis n'utilisez plus que les mnémoniques dans le programme. Vous disposez sur un S7-300 de 256 octets pour chaque tâche ou OB et sur un S7-400 jusqu'à 16 kilo-octets pouvant être répartis sur les différentes tâches ou OB.

3.12 Opérations

Le tableau suivant affiche en regard les opérations S5 et S7. Il indique quelles sont les opérations convertibles et lorsque la convertibilité n'existe pas, les moyens d'y remédier.

Tableau 3-12 Opérations dans S5 et dans S7

Type d'opération	Opérations dans S5	Opérations dans S7	Convertibilité	Conversion recommandée
Opérations sur accumulateurs	TAK, ENT, I, D, ADDBF, ADDKF, ADDDH	TAK, ENT, INC, DEC, +, Nouveauté dans S7: TAW, TAD, PUSH, POP, LEAVE	oui	-
Opérations sur registres/registres d'adresses	MA1, MBR, ABR, MAS, MAB, MSB, MSA, MBA, MBS; TSG, LRB, LRW, LRD, TRB, TRW, TRD	Nouveauté dans S7 : LAR1, LAR2, TAR1, TAR2, +AR1, +AR2, TAR	non	Utiliser les registres d'adresse AR1, AR2
Opérations sur bits	U, UN, O, ON, U(, O(,), O, S, R, RB, RD, = P, PN, SU, RU	U, UN, O, ON, U(, O(,), O, S, R, = SET; U, SET; UN, SET; S, SET; R Nouveauté dans S7: X, XN, X(, XN(FP, FN, NOT, SET, CLR, SAVE	oui	-
Temporisations	SI, SV, SE, SS/SSV, SA/SAR, FR, SVZ	SI, SV, SE, SS, SA, FR, S T	oui	-
Opérations de comptage	ZV/SSV, ZR/SAR, FR, SVZ	ZV, ZR, FR, S Z	oui	-
Opérations de chargement et de transfert	L, LC, LW, LD, T L PB, L QB, L PW, L QW, T PB, T QB, T PW, T QW	L, LC, T L PEB, L PEW, T PAB, T PAW	oui	-
	LB GB / GW / GD / CB / CW / CD, LW GW / GD / CW / CD, TB GB / GW / GD / CB / CW / CD, TW GW / GD / CW / CD		non	Remplacer par des accès à la périphérie

(suite à la page suivante)

Tableau 3-12 Opérations dans S5 et dans S7, suite

Type d'opération	Opérations dans S5	Opérations dans S7	Conver- tabilité	Conversion recommandée
Arithmétique sur nombres entiers	+F, -F, XF, :F, +D, -D	+I, -I, *I, /I, +D, -D, *D, /D Nouveauté dans S7: MOD	oui	-
Arithmétique virgule flottante	+G, -G, XG, :G	+R, -R, *R, /R	oui	-
Opérations de comparaison	!=F, ><F, >F, <F, >=F, <=F, !=D, ><D, D, <D, >=D, <=D, !=G, ><G, >G, <G, >=G, <=G	==I, <>I, >I, <I, >=I, <=I, ==D, <>D, >D, <D, >=D, <=D, ==R, <>R, >R, <R, >=R, <=R	oui	-
Opérations de conversion	KEW, KZW, KZD, DEF, DED, DUF, DUD, GFD, FDG	INVI, NEGI, NEGD, BTI, BTD, DTB, ITB, RND, DTR Nouveauté dans S7: ITD, RND+, RND-, TRUNC, INVD, NEGR	oui	-
Opérations sur mots	UW, OW, XOW	UW, OW, XOW Nouveauté dans S7: UD, OD, XOD	oui	-
Opérations de décalage	SLW, SLD, SRW, SRD, SVW, SVD, RLD, RRD	SLW, SLD, SRW, SRD, SSI, SSD, RLD, RRD Nouveauté dans S7: RLDA, RRDA	oui	-
Opérations sur blocs de données (suite à la page suivante)	A, AX	AUF	oui	
	E, EX	SFC22	non	Remplacer par l'appel de SFC22 CREATE_DB
		Nouveauté dans S7: TDB L DBLG, L DBNO, L DILG, L DINO		

Tableau 3-12 Opérations dans S5 et dans S7, suite

Type d'opération	Opérations dans S5	Opérations dans S7	Convertibilité	Conversion recommandée
Opérations de saut	SPA, SPB, SPN, SPZ, SPP, SPM, SPO, SPS, SPR	SPA, SPB, SPN, SPZ, SPP, SPM, SPO, SPS Nouveauté dans S7: SPBN, SPBB, SPBNB, SPBI, SPBIN, SPMZ, SPPZ, SPU, LOOP, SPL	oui	-
Opérations sur blocs	SPA, SPB, BA, BAB, BE, BEA, BEB	CALL, BE, BEA, BEB	oui	-
Opérations de validation / inhibition de sorties Opérations MCR	BAS, BAF	Nouveauté dans S7: MCRA, MCRD, MCR(,)MCR	non	Remplacer par un appel de SFC26, SFC27 ou par des opérations MCR
Opérations d'arrêt	STP, STS, STW	SFC46	non	Remplacer par un appel de SFC46 STP
Opérations de substitution	B <paramètre formel>	-	non	Appel de DB / le bloc de code doit être reprogrammé
	B MW, B DW	Adressage indirect en mémoire	oui	Recommandation : Remplacer par l'adressage indirect par registre
	B BS	Adressage indirect par registre interzone	non	A remplacer par l'adressage indirect (voir 3.13.4)
Adressage absolu en mémoire	LIR, TIR, LDI, TDI	-	non	A remplacer par l'adressage indirect (voir 3.13.4)
Transfert de blocs	TNB, TNW, TXB, TXW	SFC20	non	Remplacer par un appel de SFC20 BLKMOV
Commandes d'interruption	LIM, SIM, AFS, AFF, AS, AF	SFC39 -42	non	Remplacer par un appel de SFC39 - 42
Adressage par page	ACR, TSC, TSG	-	non	Plus d'adressage par page dans S7 !
Fonctions mathématiques	-	ABS, COS, SIN, TAN , ACOS, ASIN, ATAN, EXP, LN	-	-
Opérations ineffectives NOP	BLD xxx NOP 0, NOP 1	BLD xxx NOP 0, NOP 1	oui	-

3.13 Adressage

3.13.1 Adressage absolu

L'adressage absolu est identique dans S5 et S7 à une exception près :

L'adressage des données dans les blocs de données s'effectue dans STEP 7 **par octets**. Les adresses au format mot de S5 sont donc converties (multiplication par 2) en des adresses au format octet.

Le tableau suivant montre comment s'effectue la conversion de l'adressage par zones d'adresses :

S5	S7
DL 0, 1, 2, 3, ...255	DBB 0, 2, 4, 6, ...510
DR 0, 1, 2, 3, ...255	DBB 1, 3, 5, 7, ...511
DW 0, 1, 2, 3, ...255	DBW 0, 2, 4, 6, ...510
DD 0, 1, 2, 3, ...254	DBD 0, 2, 4, 6, ...508
D x.y	DBX 2 x.y für $8 \leq y \leq 15$ DBX (2 x+1).y für $0 \leq y \leq 7$

3.13.2 Adressage symbolique

L'adressage symbolique de S5 a été repris dans S7, si ce n'est que les possibilités offertes par STEP 7 sont plus grandes que celles existant dans STEP 5. La programmation avec adressage symbolique reste en revanche inchangée.

Mnémoniques de STEP 5

Les mnémoniques étaient définis pour les programmes de STEP 5 dans un éditeur de mnémoniques. Celui-ci affichait une liste d'assignation dans laquelle il était possible de définir des mnémoniques remplaçant les adresses absolues dans le programme.

Mnémoniques de STEP 7

STEP 7 autorise des mnémoniques de 24 caractères.

Mnémoniques globaux

STEP 7 possède également un éditeur de mnémoniques. La liste d'assignation de STEP 5 s'appelle dans STEP 7 la table des mnémoniques. Vous définissez dans celle-ci tous vos mnémoniques globaux (les sorties, les mémentos, les blocs etc.).

Lorsque l'affectation a eu lieu dans l'éditeur de mnémoniques, ces mnémoniques valent pour l'ensemble du programme de la CPU.

Mnémoniques locaux

Vous avez en outre la possibilité dans STEP 7 de définir lors de la programmation de blocs des mnémoniques locaux pour les opérands au format données d'une zone de données précise.

Si un mnémonique a été déclaré dans un bloc et non dans la table des mnémoniques, il ne sera valable que pour ce bloc. Nous parlons dans ce cas d'un mnémonique local.

Quand définit-on les mnémoniques ?

STEP 7 vous laisse libre de définir vos mnémoniques quand vous le voulez. Vous pouvez le faire soit :

- avant de commencer à programmer,
(nécessaire en mode de saisie incrémentale dans lequel la syntaxe est vérifiée après chaque ligne du programme),
- une fois que vous avez écrit votre programme, c'est-à-dire avant la compilation
(nécessaire en mode de saisie source du programme, lorsque celui-ci est créé sous la forme d'un fichier ASCII).

Importer une table des mnémoniques

Vous pouvez créer et éditer dans S7 des tables de mnémoniques avec l'éditeur de votre choix.

Vous pouvez également importer dans votre table des mnémoniques des tables créées avec d'autres outils en vue de les réutiliser. Vous avez par exemple recours à la fonction d'importation pour réutiliser des listes d'assignation de STEP5/ST après leur conversion dans la nouvelle table des mnémoniques de STEP 7.

Vous avez au choix les formats de fichier *.SDF, *.ASC, *.DIF et *.SEQ.

Pour importer une table des mnémoniques, procédez de la manière suivante.

1. Ouvrez la fenêtre du projet du programme S7 dans lequel se trouve la table des mnémoniques que vous désirez importer.
2. Ouvrez-la en double-cliquant sur l'icône "Mnémoniques".
3. Choisissez dans la fenêtre de la table des mnémoniques la commande de menu **Table > Importer**. Une boîte de dialogue s'affiche.
4. Sélectionnez-y la table des mnémoniques que vous désirez importer et cliquez sur le bouton "Ouvrir".
5. Vérifiez vos saisies et corrigez-les le cas échéant.
6. Sauvegardez et fermez la table des mnémoniques.

Nota

Une table de mnémoniques au format de fichier *.SEQ qui a été convertie de S5 à S7 ne peut plus être importée dans S5. Il est recommandé d'utiliser le format *.DIF pour l'échange de tables de mnémoniques entre S5 et S7.

Pour plus d'informations sur la table des mnémoniques, veuillez consulter le guide de l'utilisateur **/231/**.

3.13.3 Nouveauté : accès aux opérandes au format données avec l'adresse complète

L'adressage complet d'opérandes au format données signifie que l'opérande est indiqué avec son bloc de données. Cela n'était pas possible dans S5.

L'adressage complet doit être soit absolu, soit symbolique. Il n'est pas possible de mélanger des opérandes symboliques et absolus dans une même instruction.

Exemple

L DB100.DBW6
L DB_MOTEUR.VITESSE

DB_MOTEUR est le mnémonique défini dans la table des mnémoniques pour le DB100. VITESSE est un opérande au format données qui a été déclaré dans le bloc de données. L'adressage symbolique de l'opérande au format données (DB_MOTEUR.VITESSE) est tout aussi univoque que son adressage absolu (DB100.DBW6).

L'adressage complet n'est possible qu'avec un registre de DB globaux (registre DB). L'éditeur LIST a recours pour l'adressage complet d'opérandes au format données à deux instructions consécutives :

1. Ouverture préalable du bloc de données par l'intermédiaire du registre DB (exemple : AUF DB100)
2. Opération proprement dite sur l'opérande au format données (exemple L DBW 6)

Accès aux données avec l'adresse complète

Un adressage avec indication de l'adresse complète est possible pour toutes les opérations autorisées par le type de données de l'opérande adressé.

Ces opérandes peuvent également servir de paramètres aux blocs. Ce type d'adressage est fortement recommandé, car le bloc de données a pu changer à l'appel du bloc et en indiquant l'adresse complète, vous êtes certain que le bon opérande au format données provenant du bon bloc de données sera appelé dans le programme.

Dangers de l'adressage partiel

Il est en principe possible dans STEP 7 comme dans STEP 5 d'accéder à des opérandes au format données en ayant recours à l'adressage partiel.

Exemple :

L DBW 6
L VITESSE

L'adressage partiel n'est toutefois dans STEP 7 pas sans certains dangers, le programme changeant lors de certaines opérations de registre de CPU. Dans certains cas, le numéro de DB est écrasé dans le registre DB.

Le registre DB peut être écrasé dans les cas suivants réclamant une attention particulière :

- Le registre DB est écrasé lors d'un adressage complet de données.
- En cas d'appel d'un FB, le registre DB du bloc appelant est écrasé.
- Après l'appel d'une fonction (FC) fournissant un paramètre au type de données complexe comme STRING, DATE_AND_TIME, ARRAY, STRUCT ou UDT, le contenu du registre DB du bloc appelant est écrasé.
- Pour affecter à une FC un paramètre effectif sauvegardé dans un DB (par exemple, DB100.DBX0.1), STEP 7 ouvre le DB (DB100) et écrase en même temps le contenu du registre DB.
- Après l'adressage par un FB d'un paramètre d'entrée/sortie au type de données complexe comme STRING, DATE_AND_TIME, ARRAY, STRUCT ou UDT, STEP 7 utilise le registre DB pour accéder aux données et écrase le contenu du registre DB.
- Après l'adressage par une FC d'un paramètre (entrée, sortie ou entrée/sortie) au type de données complexe comme STRING, DATE_AND_TIME, ARRAY, STRUCT ou UDT, STEP 7 utilise le registre DB pour accéder aux données et écrase le contenu du registre DB.

3.13.4 Adressage indirect

La fonction de substitution de S5 est remplacée dans S7 par les nouvelles instructions utilisant l'adressage indirect en mémoire et par registre.

Format de pointeur dans STEP 5

Le pointeur occupe dans S5 un mot pour l'opération de substitution indexée. Il est représenté sur la figure 3-7.

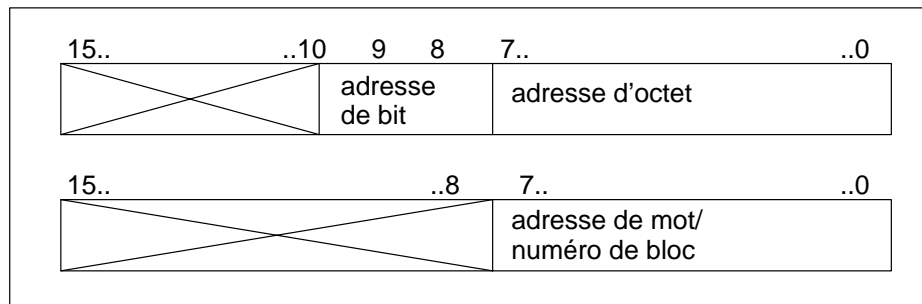


Figure 3-7 Constitution d'un pointeur S5

Formats de pointeur dans STEP 7

On rencontre deux formats de pointeur dans S7 : le mot et le double mot.

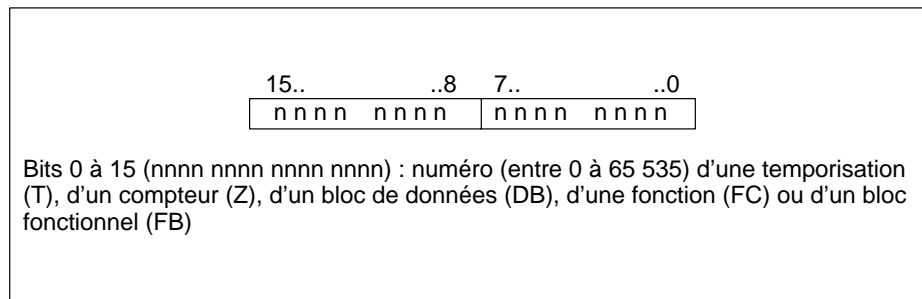


Figure 3-8 Pointeur au format mot pour l'adressage indirect en mémoire

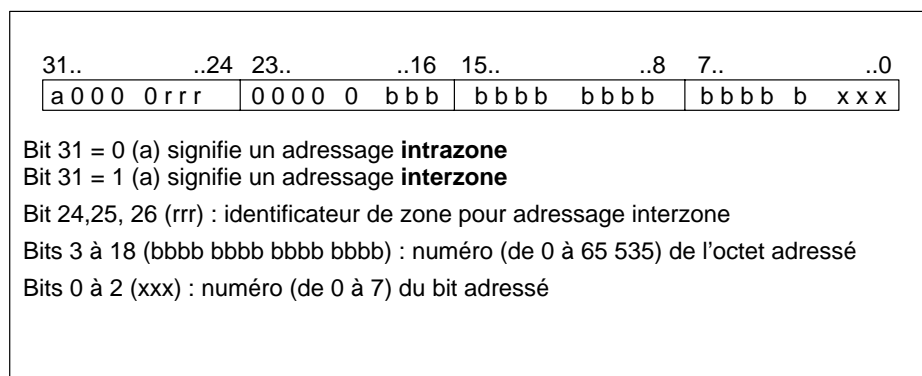


Figure 3-9 Pointeur au format double mot pour l'adressage indirect en mémoire et l'adressage indirect par registre

Adressage indirect en mémoire

L'adressage indirect en mémoire n'a pas changé par rapport à S5. Dans l'adressage indirect en mémoire, l'opérande indique l'adresse de la valeur sur laquelle l'opération doit porter. L'opérande est constitué :

- d'un identificateur permettant de l'identifier (par exemple "EB" pour octet d'entrée) et
- d'un mot indiquant le numéro d'une temporisation (T), d'un compteur (Z), d'un bloc de données (DB), d'une fonction (FC) ou d'un bloc fonctionnel (FB) ou
- d'un double mot indiquant l'adresse exacte à l'intérieur de la zone de mémoire (indiquée par l'identificateur d'opérande)

L'opérande indique indirectement via le pointeur l'adresse de la valeur ou du numéro. Ce mot ou double mot peut se trouver dans les zones de mémoire suivantes.

- Mémento (M)
- Bloc de données (DB)
- Bloc de données d'instance (DI)
- Données locales (L)

L'avantage de l'adressage indirect en mémoire est que l'opérande peut être modifié de manière dynamique durant l'exécution du programme.

Exemples

L'exemple suivant vous montre comment utiliser un pointeur au format mot.

LIST S5	LIST S7	Commentaire
L KB 5 T MW 2 B MW 2 L T 0	L +5 T MW 2 L T [MW 2]	Charger la valeur 5 comme nombre entier dans l'accumulateur 1. Transférer le contenu de l'accumulateur 1 dans le mot de memento MW2. Charger la valeur temporelle de la temporisation T 5.

Les deux exemples suivants vous montrent comment utiliser un pointeur au format double mot.

LIST S5	LIST S7	Commentaire
L KB 8 T MB 3 L KB 7 T MB 2 B MW 2 U E 0.0 B MW 2 = A 0.0	L P#8.7 T MD 2 U E [MD 2] = A [MD 2]	Charger 2#0000 0000 0000 0000 0000 0000 0100 0111 (valeur binaire) dans l'accumulateur 1 (S7). Stocker l'adresse 8.7 dans le mot de memento MW 2 (S5) / double mot de memento MD 2 (S7). L'automate interroge le bit d'entrée E 8.7 et affecte son état de signal à la sortie A 8.7.

LIST S5	LIST S7	Commentaire
L KB 8 T MW 2 B MW 2 L EB 0 B MW 2 T MW 0	L P#8.0 T MD2 L EB [MD2] T MW [MD2]	Charger 2#0000 0000 0000 0000 0000 0000 0100 0000 (valeur binaire) dans l'accumulateur 1 (S7). Stocker l'adresse 8 dans le mot de memento MW 2 (S5) / double mot de memento MD 2 (S7). L'automate interroge l'état de l'octet d'entrée EB 8 et transfère son contenu dans le mot de memento MW 8.

Utilisation de la syntaxe correcte

Lorsque vous utilisez une adresse indirecte en mémoire qui est sauvegardée dans la zone de mémoire de bloc de données, ouvrez d'abord le bloc de données à l'aide de l'opération AUF. Vous pouvez alors utiliser le mot ou le double mot de données comme adresse indirecte comme illustré dans l'exemple suivant :

```
AUF          DB10
L            EB [DBD 20]
```

Si vous accédez à un octet, mot ou double mot, prenez garde à ce que le numéro de bit de votre pointeur soit égal à 0.

Adressage indirect par registre

Les registres d'adresse AR1 et AR2 sont utilisés pour l'adressage indirect par registre dans STEP 7.

Dans ce type d'adressage, l'opérande indique l'adresse de la valeur sur laquelle l'opération doit porter. L'opérande se compose :

- d'un identificateur d'opérande,
- d'un registre d'adresse et d'un pointeur marquant un déplacement. Ajouté au contenu du registre, ce déplacement détermine l'adresse exacte de la valeur que doit traiter l'opération. Le pointeur est indiqué par P#octet.bit.

L'opérande désigne donc l'adresse de la valeur indirectement par le registre d'adresse plus le déplacement.

Une opération utilisant l'adressage indirect intrazone par registre ne modifie pas la valeur figurant dans le registre d'adresse.

Pour plus d'informations sur ce type d'adressage, référez-vous au manuel LIST /232/.

Deuxième partie : Conversion du programme

Démarche

4

Préparation de la conversion

5

Conversion

6

Retouche du programme converti

7

Compilation

8

Exemple d'application

9

Démarche

Pour une grande part, la programmation S7 en LIST, CONT et LOG est compatible avec – respectivement – LIST, CONT et LOG de S5. Par conséquent, si vous êtes un utilisateur de S5 et que souhaitez mettre en œuvre dans S7 des programmes existants, la conversion en sera très facile. Vous pouvez rester fidèle à vos programmes S5 éprouvés ; il suffit de les convertir en programmes S7.

Marche à suivre

Ce chapitre indique la démarche à suivre pour convertir votre programme S5 et renvoie à chaque fois au chapitre traitant l'étape décrite.

Cette démarche est générale. Il est donc possible d'en sauter des étapes.

4.1 Analyse du système S5

Avant de convertir votre programme S5, nous vous conseillons de vérifier que les conditions requises pour la conversion sont bien remplies.

Fonctions des cartes (voir chapitre 2)

Comment pouvez-vous réaliser les fonctions des cartes S5 que vous utilisez ? Pouvez-vous employer vos cartes S5 dans S7 à l'aide de boîtiers d'adaptation ou de cartes de couplage ? Est-il possible de remplacer les cartes S5 par des modules S7 ?

Paramètres du système (voir paragraphe 3.8)

Comment peut-on réaliser les paramétrages système requis dans S7 ?

Jeu d'opérations (voir paragraphe 3.12)

Le jeu d'opérations utilisé par la CPU S5 est-il réalisable avec votre CPU S7 ?

Quand certaines opérations ne sont pas convertibles, vous recevez un message précisant les endroits correspondants dans le programme et devez reprogrammer ces opérations vous-même.

Logiciel standard (voir paragraphe 3.9)

Les blocs fonctionnels standard de S5 appelés dans le programme à convertir existent-ils aussi comme fonctions dans S7 ?

Le logiciel de base S7 fourni comprend les logiciels standard déjà convertis pour l'arithmétique à virgule flottante, les fonctions de signalisation, les fonctions intégrées, les fonctions de base et les fonctions mathématiques.

Fonctions spéciales (voir tableaux à partir de la page 3-22)

Est-il possible de remplacer les fonctions spéciales intégrées éventuellement utilisées dans le programme S5 ?

Parties du programme à reprogrammer dans S7

En règle générale, il n'est pas possible de convertir toutes les parties d'un programme. Les conseils suivants vous aideront à décider si vous effectuerez la conversion de votre programme S5 à l'aide du convertisseur ou si vous récrirez dans S7 votre ancien programme S5.

- Les programmes contenant uniquement des combinaisons sur mots et sur bits ne nécessitent pas de retouches.
- L'accès à des opérandes via des adresses absolues n'est pas possible dans S7. Les opérations correspondantes (par exemple, LIR, TIR, etc.) ne sont pas converties. Si un programme utilise surtout des adresses absolues, il est préférable d'en récrire les parties correspondantes ou même l'intégralité.
- Les fonctions de substitution (par exemple, B MW, B DW) sont certes partiellement converties, mais vous gagnerez de l'espace mémoire si vous les reprogrammez dans S7 en utilisant l'adressage indirect.
- En cas d'appels de blocs, il faut absolument vérifier et adapter l'affectation des paramètres, car les paramètres effectifs sont repris sans vérification lors de la conversion.

4.2 Création du projet S7

Il existe deux méthodes pour créer un projet dans STEP 7.

Création d'un projet avec l'assistant STEP 7

L'assistant STEP 7 vous permet de créer très rapidement un projet STEP 7 pour la CPU que vous comptez utiliser. Vous pouvez ensuite entamer la programmation.

Création d'un projet sans assistance

Vous pouvez bien sûr aussi créer vous-même le projet. La marche à suivre est décrite au paragraphe 3.3.1.

4.3 Configuration du matériel

A ce stade, il est recommandé de configurer le matériel, car l'application HW Config détermine des données pouvant déjà servir lors de la préparation de la conversion.

Mais, si vous ne désirez pas encore définir la configuration matérielle, vous pouvez bien sûr reporter cette configuration à plus tard.

Définition du matériel

Les informations du chapitre 2 (Matériel) vous aideront à sélectionner les cartes ou modules S7 et S5 nécessaires à votre configuration et à compléter la table de configuration du matériel (voir paragraphe 3.4).

Affectation des adresses

L'application HW Config affecte automatiquement des adresses aux cartes et modules. Vous pouvez tenir compte de ces adresses dès la conversion.

Paramétrages système

Lors du paramétrage de la CPU dans HW Config, vous pouvez procéder à des paramétrages du système correspondant à ceux réalisés dans S5 par l'intermédiaire du DB1, du DX0 ou des utilitaires système (voir paragraphe 3.4).

Rémanence

Vous pouvez également définir la rémanence dans les données de paramétrage de la CPU. Le comportement de rémanence dépend toutefois de la sauvegarde par pile (voir paragraphe 3.4).

Préparation de la conversion

Présentation

Fichiers requis (paragraphe 5.1)	<ul style="list-style-type: none">• Fichier programme <nom>ST.S5D• Liste des références croisées <nom>XR.INI• Liste d'assignation facultative <nom>Z0.SEQ
Vérification des opérandes (paragraphe 5.2)	<ul style="list-style-type: none">• Nombre d'opérandes• Nombre de blocs
Préparation du programme S5 (paragraphe 5.3)	<ul style="list-style-type: none">• Evaluation et suppression des blocs de données DB1 ou DX0• Suppression des appels de blocs intégrés• Suppression des accès à la zone de données système• Adaptation des zones d'opérandes• Définition de macro-instructions pour les parties de programme non convertibles• Suppression de blocs de données sans structure, à l'exception d'un mot de données
Création de macro-instructions (paragraphe 5.4)	<ul style="list-style-type: none">• Macro-instructions pour opérations• Macro-instructions pour OB

5.1 Fichiers requis

Les fichiers suivants sont nécessaires à la conversion de votre programme S5 :

- fichier programme <nom>ST.S5D
- et liste des références croisées <nom>XR.INI

La liste croisée est nécessaire afin de conserver la structure et la hiérarchie d'appel du programme S5.

Fichier facultatif

Si vous désirez employer des mnémoniques (noms symboliques) plutôt que des adresses absolues dans votre programme, vous aurez besoin de :

- la liste d'assignation S5 <nom>Z0.SEQ

pour créer la liste d'assignation convertie.

Marche à suivre

Procédez comme suit pour préparer la conversion :

1. Créez à l'aide du logiciel S5 une liste croisée actuelle pour votre programme S5.
2. Copiez dans un répertoire DOS votre fichier programme STEP 5, la liste croisée correspondante et, le cas échéant, la liste d'assignation.

5.2 Vérification des opérandes

Fonctions disponibles dans la CPU

Il s'avérera peut-être nécessaire d'adapter le programme à convertir à la CPU S7 que vous comptez utiliser.

Procédez comme suit pour vous faire une idée des fonctions disponibles dans votre CPU S7 :

1. Déterminez la CPU S7 que vous désirez utiliser.
2. Recherchez ses caractéristiques dans les tableaux du paragraphe 2.2.1 et comparez :
 - le nombre d'opérandes
 - et le nombre de blocs

aux opérandes et blocs utilisés.

Ou bien

1. Ouvrez le gestionnaire de projets SIMATIC (SIMATIC Manager).
2. Sélectionnez votre CPU S7 dans la vue en ligne de la structure du projet.
3. Sélectionnez la commande **Système cible > Etat du module** ; une boîte de dialogue avec, entre autres, les informations suivantes s'ouvre alors :
 - La page d'onglet **Fiche d'identité** vous renseigne sur le type de CPU, la configuration de la mémoire et la taille des zones d'opérandes disponibles.
 - La page d'onglet **Blocs** contient des informations sur les blocs disponibles. Elle précise en particulier le nombre et la longueur maximum des types de blocs et énumère tous les blocs OB, SFB et SFC existant dans la CPU.

Adaptation du programme à convertir

Pour que le programme LIST à convertir puisse plus tard s'exécuter dans la CPU S7 utilisée, vous devez vérifier le nombre de blocs et le nombre d'opérandes autorisés et si nécessaire les modifier.

5.3 Préparation du programme S5

Avant la conversion, vous pouvez préparer votre programme STEP 5 à son emploi ultérieur comme programme STEP 7. Cela n'est pas une obligation : vous pouvez aussi effectuer toutes les corrections dans le fichier source STEP 7 après la conversion. L'adaptation vous permet toutefois de réduire le nombre de messages d'erreur et d'avertissement.

Vous pouvez, par exemple, procéder aux adaptations suivantes avant la conversion :

- Évaluez les paramètres système figurant dans les blocs de données DB1 ou DX0, puis supprimez ces blocs.
- Supprimez tous les appels de blocs intégrés ou accès à la zone de données système BS dont vous pouvez réaliser les fonctions par paramétrage de la CPU S7.
- Adaptez les zones d'opérandes des entrées, des sorties et de la périphérie aux nouvelles adresses de modules à l'aide de la fonction "Réassignation" de STEP 5. Veillez, ce faisant, à ne pas dépasser la plage d'adresses de STEP 5, car cette erreur serait signalée pendant la première phase de la conversion et les instructions correspondantes ne seraient pas converties.
- Supprimez les parties de programme récurrentes non convertibles, à l'exception d'une instruction STEP 5 non équivoque par partie de programme. Vous affecterez à cette instruction non équivoque une macro-instruction remplaçant la partie de programme (voir paragraphe 5.4).
- Si votre programme contient beaucoup de blocs de données longs et sans structure (servant, par exemple, de mémoire tampon), supprimez les mots de données de ces blocs, à l'exception d'un mot de données. Après la conversion et avant la compilation, vous programmerez le contenu de ces blocs dans le fichier source via une déclaration de tableau. Exemple : tampon: ARRAY [1..256] of WORD.

Le convertisseur vous permet non seulement de convertir des programmes complets, mais également des blocs individuels.

5.4 Création de macro-instructions

Utilité

Le convertisseur S5/S7 vous permet de définir des macro-instructions pour :

- les opérations S5 qui ne sont pas converties automatiquement
- et les opérations S5 que vous souhaitez convertir autrement que selon la conversion standard.

Les macro-instructions s'avèrent utiles lorsque votre programme contient plusieurs opérations S5 présentant les caractéristiques énumérées ci-dessus.

Fonction des macro-instructions

Les macro-instructions peuvent remplacer :

- des opérations S5 (opérateurs)
- et des blocs d'organisation (OB) de S5.

Elles sont rangées dans le fichier S7S5CAPA.MAC pour le jeu d'opérations SIMATIC et dans le fichier S7S5CAPB.MAC pour le jeu d'opérations international. Si vous utilisez ces deux jeux d'opérations, vous devez faire figurer les macro-instructions dans chaque fichier séparément. On distingue les macro-instructions pour opérations et celles pour OB. Vous pouvez créer 256 macro-instructions de chaque sorte.

5.4.1 Macro-instructions pour opérations

Les macro-instructions pour opérations doivent être construites comme ceci :

```
$MAKRO: <opération S5>
```

```
suite d'opérations S7
```

```
$ENDMAKRO
```

Dans la définition de la macro-instruction , vous indiquez l'instruction complète – opérateur et opérande absolu – pour <opération S5>.

Le tableau suivant montre une macro-instruction pour l'instruction E DB 0 servant dans S5 à définir des blocs de données. La longueur en mots du bloc de données à définir figure dans l'accumulateur 1. Dans S7, c'est la fonction système SFC 22, CREAT_DB, qui permet de réaliser cette opération. La longueur du bloc de données doit être convertie en nombre d'octets.

Tableau 5-1 Exemple de macro-instruction pour opération

Macro-instruction	S5	S7
\$MAKRO: E DB 0 //Remplace l'opération //de définition de DB	L constante B MW 100	L constante
SLW 1 //Conversion du nombre de //mots en nombre d'octets T MW 102 CALL SFC 22(//Appel de la SFC CREAT_DB LOW_LIMIT := MW 100, UP_LIMIT := MW 100, COUNT := MW 102, RET_VAL := MW 106, DB_NUMBER := MW 104); \$ENDMAKRO	E DB 0	SLW 1; T MW 102; CALL SFC 22(LOW_LIMIT := MW 100, UP_LIMIT := MW 100, COUNT := MW 102, RET_VAL := MW 106, DB_NUMBER := MW 104);

5.4.2 Macro-instructions pour OB

En raison des différences entre les blocs d'organisation de S5 et ceux de S7, il peut être préférable de gérer vous-même la conversion des OB S5. Les macro-instructions pour OB doivent être construites comme ceci :

```
$OBCALL: <numéro de l'OB>
CALL <fonction système S7>;
$ENDMAKRO
```

Quand le logiciel trouve dans le fichier source S5 une instruction avec l'opérande OBx, il la remplace par la macro-instruction que vous avez définie. Les appels de FB qui utilisent des OB comme paramètres formels constituent une exception.

Tableau 5-2 Exemple de macro-instruction pour OB

Macro-instruction	S5	S7
\$OBCALL: 31 //Remplace les opérations avec OB31 CALL SFC 43; \$ENDMAKRO	SPA OB 31	CALL SFC 43;

Conseils pour la création

Dans S5, les blocs d'organisation n'ont pas les mêmes fonctions que dans S7. Lors de la retouche du programme converti, vous devez remplacer les OB qui ne sont pas convertis automatiquement par :

- des blocs d'organisation avec d'autres fonctions,
- de nouvelles opérations S7
- ou des paramètres système que vous définissez lors du paramétrage du matériel.

Vous trouverez, au paragraphe 3.7.5, des renseignements détaillés sur la façon de remplacer les OB de S5.

Nota

Le logiciel ne vérifie pas si une macro-instruction est définie en double. Si ce cas se produit, il utilisera la macro-instruction définie en premier.

Il ne vérifie pas non plus si la séquence d'opérations S7 précisée est correcte.

Veillez à écrire correctement les mots-clés et les caractères spéciaux (deux-points).

5.4.3 Edition de macro-instructions

Procédez comme suit pour créer des macro-instructions :

- Lancez le convertisseur S5/S7 en cliquant sur le bouton “Démarrer” dans la barre des tâches de Windows 95, puis sélectionnez “Simatic, STEP 7, Conversion de fichiers S5”.
- Choisissez la commande **Edition > Macro de remplacement** (aucun fichier programme n’étant ouvert).

Résultat : Le fichier S7S5CAPA.MAC est ouvert.

- Saisissez les macro-instructions comme décrit ci-avant, puis enregistrez le fichier avec la commande **Fichier > Enregistrer**.
- Fermez le fichier avec la commande **Fichier > Fermer**.

Résultat : Le fichier S7S5CAPA.MAC est fermé. Les macro-instructions définies entreront en vigueur dès la prochaine phase de conversion.

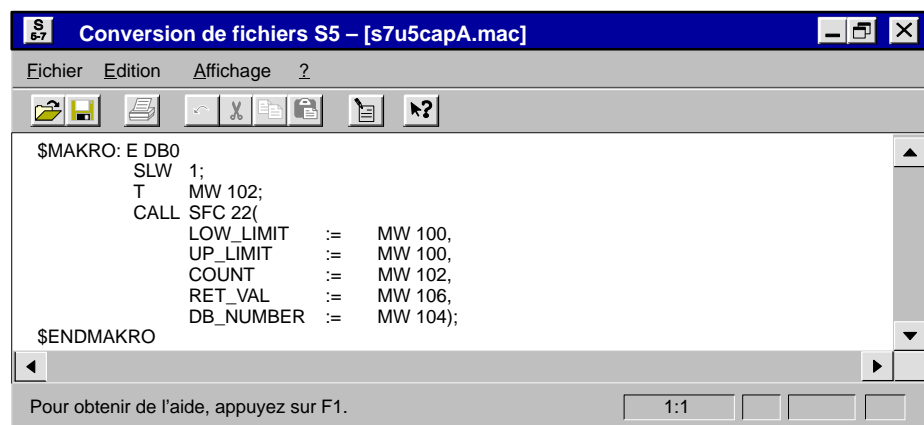


Figure 5-1 Macro-instruction dans la fenêtre « Conversion de fichiers S5 »

Conversion

6.1 Lancement de la conversion

Conditions préalables

Avant de lancer la conversion, assurez-vous que le fichier S5 à convertir, la liste des références croisées et, éventuellement, la liste d'assignation se trouvent bien dans le même répertoire (voir aussi le paragraphe 5.1).

Lancement du convertisseur S5/S7

Après avoir installé le logiciel STEP 7 dans votre console de programmation, lancez le convertisseur S5/S7 en cliquant sur le bouton "Démarrer" dans la barre des tâches de Windows 95.

- Sélectionnez l'option "Simatic, STEP 7, Conversion de fichiers S5".

Le convertisseur S5/S7 se présente avec l'image-écran ci-après.



Figure 6-1 Image-écran initiale du convertisseur S5/S7

Sélection d'un fichier programme

Procédez comme suit pour sélectionner un fichier programme :

1. Exécutez la commande **Fichier > Ouvrir**.
2. Sélectionnez le lecteur et le répertoire dans lesquels se trouvent les fichiers à convertir.
3. Sélectionnez le fichier à convertir et cliquez sur OK pour confirmer votre choix.

Résultat : Le convertisseur S5/S7 affiche les fichiers source et cible ainsi qu'une table de correspondance des anciens et nouveaux numéros de blocs.

La figure suivante présente la boîte de dialogue "Conversion de fichiers S5 – [<nom>ST.S5D]".

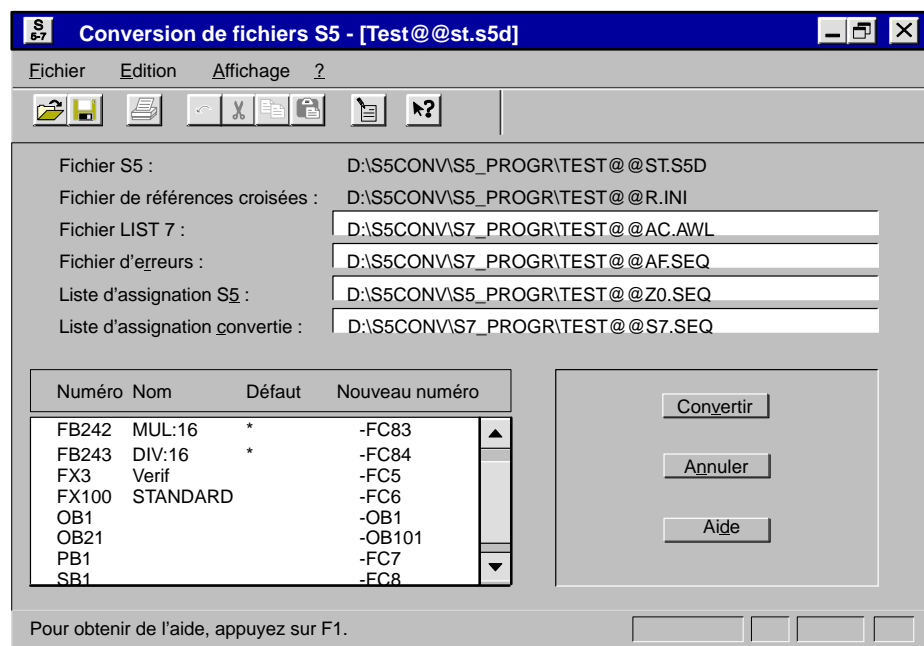


Figure 6-2 Boîte de dialogue « Conversion de fichiers S5 – [<nom>ST.S5D] »

Modification des noms des fichiers cible

Si besoin est, vous pouvez modifier les noms proposés par le logiciel pour les fichiers cible "Fichier LIST", "Fichier d'erreurs" et "Liste d'assignation convertie". Cela peut s'avérer nécessaire quand l'éditeur avec lequel vous voulez retoucher le fichier converti impose certaines conventions pour les noms (NOM.TXT, par exemple).

Procédez comme suit :

- Cliquez sur la zone de texte contenant le nom de fichier cible que vous voulez modifier.
- Apportez les modifications souhaitées.

Correspondance Numéro → Nouveau numéro

Le logiciel propose de nouveaux numéros pour les blocs à convertir et les inscrit dans la boîte de dialogue "Conversion de fichiers S5 – [<nom>ST.S5D]". Procédez comme suit pour attribuer d'autres numéros :

1. Cliquez deux fois sur le numéro de bloc que vous voulez changer.
2. Entrez le nouveau numéro dans la boîte de dialogue "Nouveau numéro de bloc" et cliquez sur le bouton "OK" pour valider l'entrée.

Blocs fonctionnels standard de S5

Si votre programme S5 contient des blocs fonctionnels standard, ils sont signalés par un astérisque dans la colonne “Défaut”.

Exécution de la conversion

En cliquant sur le bouton “Convertir“, vous mettez en route la procédure de conversion. Elle se compose de deux phases de conversion et de la transposition de la liste d’assignation.

Au cours de la première phase de conversion, le programme S5 est converti en un fichier source S5 avec tous les blocs et tous les commentaires.

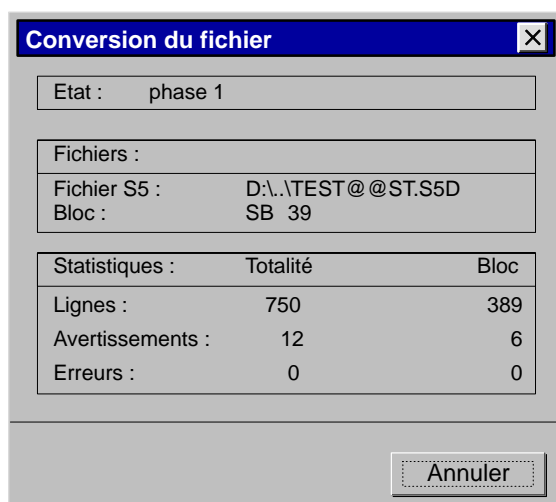
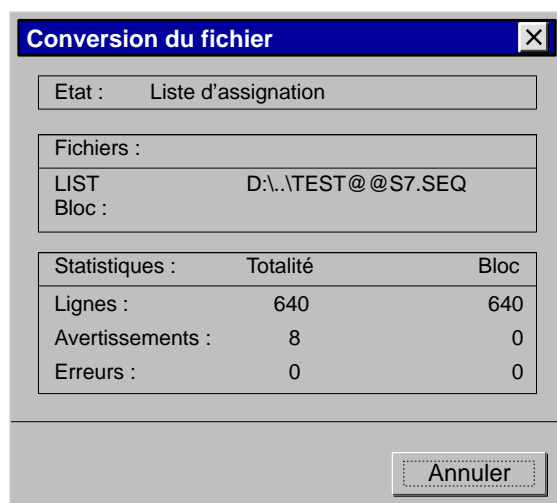


Figure 6-3 Première phase de la conversion

Au cours de la deuxième phase, le fichier source S5 est converti en un fichier source LIST avec les nouveaux numéros de bloc et la syntaxe de S7.

Conversion de la liste d'assignation

Au cours de la conversion de la liste d'assignation, les mnémoniques de la liste d'assignation de S5 sont convertis en un format que l'éditeur de mnémoniques peut importer.



Conversion du fichier [X]

Etat : Liste d'assignation

Fichiers :

LIST D:\..\TEST@@S7.SEQ

Bloc :

Statistiques :	Totalité	Bloc
Lignes :	640	640
Avertissements :	8	0
Erreurs :	0	0

Annuler

Figure 6-4 Conversion de la liste d'assignation

6.2 Fichiers générés

Le convertisseur S5/S7 génère les fichiers suivants lors de la conversion :

- Fichier <nom>A0.SEQ

Ce fichier est créé pendant la première phase de conversion. Il contient le fichier <nom>ST.S5D sous forme ASCII.

- Fichier <nom>AC.AWL

Ce fichier est créé pendant la seconde phase de conversion. Il contient le programme LIST. De cette seconde phase peuvent également provenir des messages résultant de définitions incorrectes de macro-instructions.

- Fichier <nom>S7.SEQ

Ce fichier est créé lors de la transposition de la liste d'assignation. Il contient la liste d'assignation convertie en un format que l'éditeur de mnémoniques peut importer.

- Fichier d'erreurs <nom>AF.SEQ

Ce fichier, affiché dans la partie supérieure de la fenêtre "Conversion de fichiers S5", contient les erreurs et les avertissements figurant dans le programme converti. Ces messages sont générés pendant les première et seconde phases de conversion et pendant la transposition de la liste d'assignation.

La conversion une fois terminée, une boîte de dialogue indiquant le nombre d'erreurs et d'avertissements s'affiche.

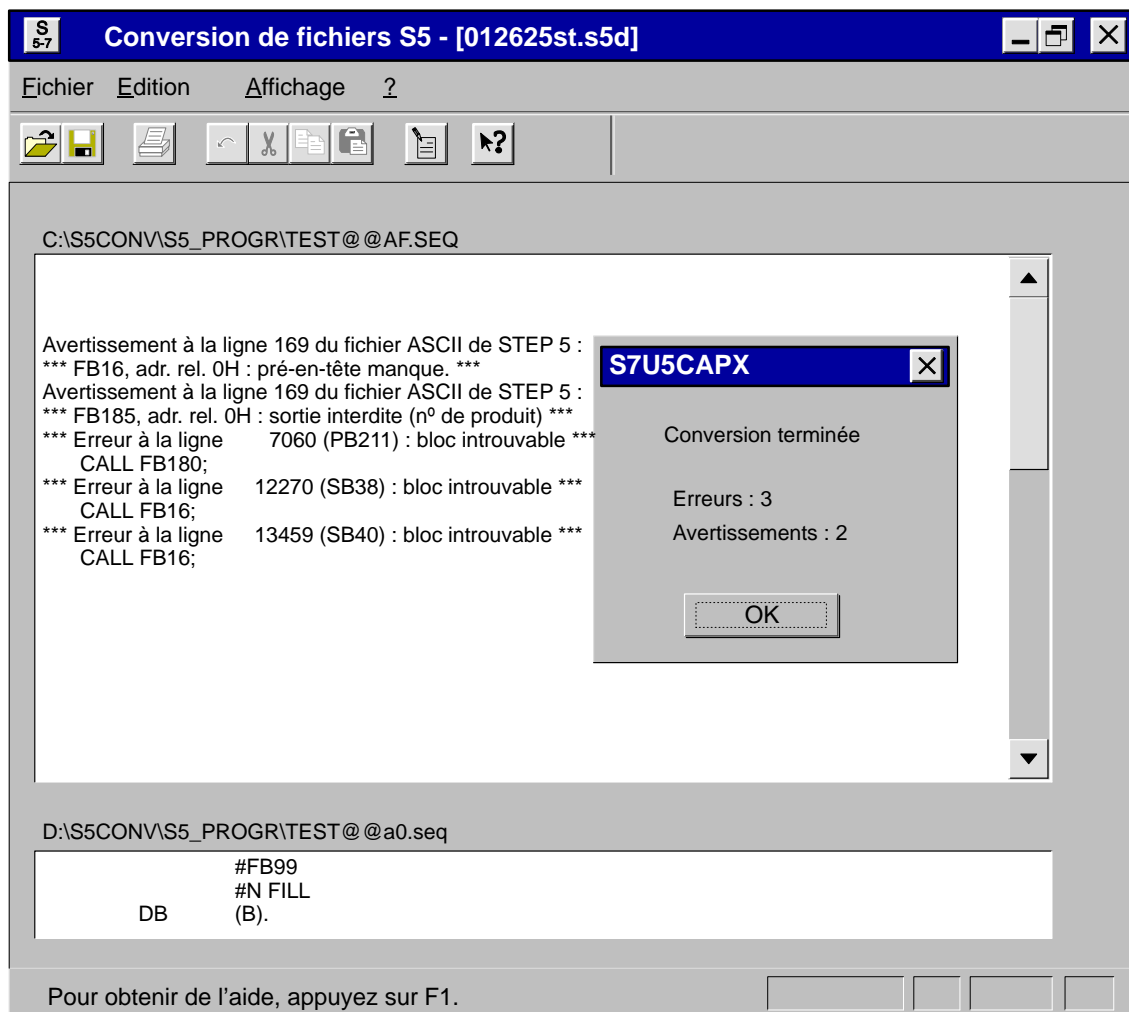


Figure 6-5 Messages affichés par le convertisseur

Localisation des erreurs

Dans la zone inférieure de la fenêtre “Conversion de fichiers S5”, vous pouvez visualiser dans le fichier concerné la position à laquelle l’erreur s’est produite.

Le fichier source LIST mentionne aussi les messages du convertisseur aux endroits du programme où des erreurs ont été constatées. En outre, il contient des avertissements ou des observations au sujet des problèmes qui peuvent se présenter (par suite de modifications de la sémantique des opérations, par exemple).

Impression des messages

La commande **Fichier > Imprimer** vous permet d’imprimer au choix les fichiers créés.

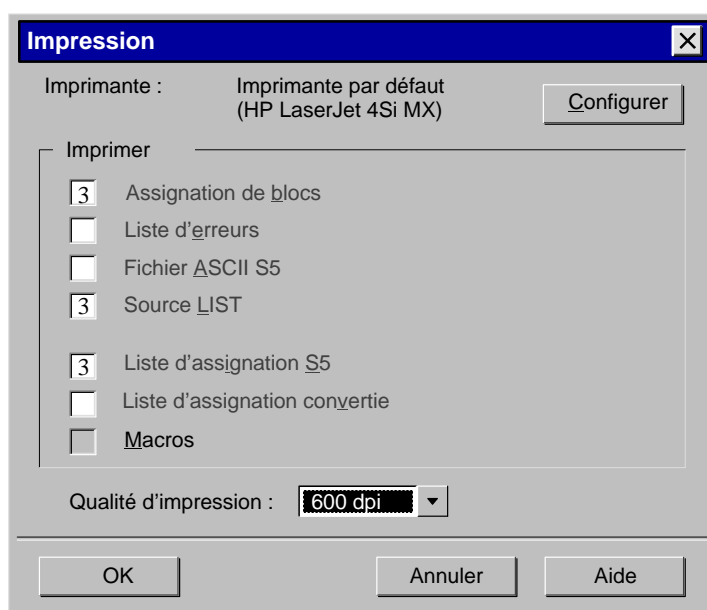


Figure 6-6 Boîte de dialogue “Impression”

6.3 Interprétation des messages

Analyse des messages

Parmi les messages du convertisseur, on distingue les messages d'erreur et les avertissements. Procédez comme suit pour analyser ces messages :

- Visualisez dans la zone inférieure de la fenêtre "Messages" le fichier dans lequel l'erreur s'est produite.
- Consultez l'aide en ligne pour comprendre la signification du message.
- Corrigez l'erreur comme il est proposé.

Messages d'erreur

Un message d'erreur est émis quand une partie du programme S5 n'est pas convertible et ne peut figurer qu'en tant que commentaire dans le programme S7. Le tableau suivant dresse la liste de tous les messages d'erreur avec leur signification et les mesures à prendre pour remédier à l'erreur.

Règles de conversion

Le chapitre 3, "Logiciel", présente les règles pour la conversion de programmes S5 en programmes S7. Vous y trouverez d'autres renseignements sur les sources d'erreur possibles et des aides permettant la retouche du programme LIST.

Tableau 6-1 Messages d'erreur, signification et remède

Message d'erreur	Origine	Signification	Remède
Paramètre absolu diverge de l'identificateur d'opérande	phase 1	L'identificateur d'opérande n'est pas correct.	Vérifiez l'instruction.
Bloc introuvable	phase 1	Le bloc appelé (FB, FX) manque ou il figure dans la liste des blocs mais n'existe pas dans le fichier programme.	Vérifiez la structure du programme.
	phase 2	Un bloc est appelé qui n'existe pas dans le fichier de programme.	Vérifiez que la liste de références croisées a bien été indiquée lors de la conversion ou contrôlez la structure du programme.
La commande n'est pas autorisée dans ce bloc.	phase 1	Saut à l'intérieur d'un bloc de programme, par exemple.	Vérifiez l'instruction.
Commande non définie.	phase 1	L'instruction MC5/LIST n'est pas valable.	Corrigez le fichier programme S5.
	phase 2	L'instruction n'existe pas dans S7.	Editez une macro-instruction ou remplacez l'instruction par la séquence d'instructions de S7 appropriée.
L'accès par bits au compteur/à la temporisation n'est pas possible. Veuillez vérifier.	phase 2	Le programme S5 contient des accès par bit à des temporisations et à des compteurs.	Vérifiez le programme LIST.
CALL OB n'est pas autorisé.	phase 2	L'appel de blocs d'organisation n'est pas autorisé dans S7.	Le cas échéant, utilisez l'instruction CALL SFC.
CALL SFCxy a été généré, veuillez compléter la liste des paramètres.	phase 2	Des paramètres SFC manquent.	Complétez la liste des paramètres SFC.
Fichier introuvable	globale	Le fichier sélectionné n'existe pas.	Vérifiez le fichier de programme.

Tableau 6-1 Messages d'erreur, signification et remède (suite)

Message d'erreur	Origine	Signification	Remède
Profondeur d'imbrication incorrecte.	phase 1	Toutes les parenthèses ne sont pas correctement fermées	Respectez les niveaux de parenthèse, éliminez l'erreur de programmation.
Opérande incorrect	phase 1	L'opérande ne convient pas à l'opération.	Vérifiez la source S5.
	phase 2	L'opérande ne convient pas à l'opération.	Modifiez le fichier LIST.
Erreur de conversion	phase 2	Opération BI sans constante	Complétez l'opération de chargement par une constante.
Erreur dans le fichier macro, xy non pris en compte	phase 2	Erreur de macro-instruction	Vérifiez la macro-instruction.
Paramètre formel non défini	phase 1	Il y a plus de paramètres que dans le bloc appelant.	Vérifiez le fichier programme S5.
Fichier ou répertoire introuvable	phase 1	Le fichier programme ne contient aucun bloc.	Vérifiez le fichier programme.
Longueur de commentaire incorrecte	phase 1	Erreur dans le fichier S5	Vérifiez le fichier programme.
Commentaire trop long	phase 1	Erreur dans le fichier S5	Vérifiez le fichier programme.
Aucun nom de bloc spécifié	phase 1	Le nom de bloc ne comporte que des espaces.	Entrez un nom de bloc.
Droits d'accès manquants	globale	Le fichier est protégé en écriture.	Supprimez la protection en écriture.
Marque non définie	phase 1	Le repère de saut n'est pas défini dans l'étiquette.	Vérifiez le fichier S5.
Marque incorrecte	phase 1	Le repère de saut contient des caractères non valables.	Vérifiez le fichier S5.
Opérateur incorrect	phase 1	L'opérateur dans le fichier S5 est inconnu ou impossible à convertir.	Remplacez l'opérateur par l'opération S7 appropriée.
Opérateur incorrect, peut éventuellement être remplacé par l'instruction : \\'''L P# paramètre formel\\'''.	phase 2	L'opérateur ne peut pas être chargé sous cette forme dans S7.	Utilisez éventuellement l'instruction indiquée.
Nombre de paramètres incorrect	phase 1	Erreur dans le programme S5	Vérifiez le fichier programme.
Paramètre erroné	phase 1	Erreur dans le programme S5	Vérifiez le fichier programme.
Type de paramètre incorrect.	phase 1	Erreur dans le programme S5	Vérifiez le fichier programme.
Erreur d'écriture disquette	globale	Le fichier est protégé en écriture ou il n'y a plus de place sur la disquette.	Supprimez la protection en écriture ou effacez les données dont vous n'avez pas besoin.
Débordement de la mémoire dans la PG (problèmes de place)	phase 1	La mémoire centrale est insuffisante.	Effacez de la mémoire centrale les fichiers dont vous n'avez plus besoin.
Le repère de saut ne peut être généré.	phase 2	L'opération SPR dépasse la limite du bloc.	Éliminez l'erreur dans le programme S5.
Un code MC5 incorrect a été converti.	phase 1	Conversion d'une ancienne opération de S5.	Aucun

Avertissements

Un avertissement est émis quand une partie du programme S5 est certes convertie, mais qu'il faudrait en vérifier la validité.

Tableau 6-2 Avertissements, signification et réaction conseillée

Avertissement	Origine	Signification	Réaction
Version incorrecte (no de produit)	phase 1	Un bloc fonctionnel standard de S5 doit être remplacé par une FC de S7.	Aucune
Version incorrecte (bloc GRAPH 5)	phase 1	Les blocs GRAPH 5 ne sont pas convertibles.	Utilisez éventuellement un bloc créé avec GRAPH pour S7.
Vérifiez la base de temps choisie.	phase 2	Dans S7, la base de temps peut être plus serrée que dans S5.	Paramétrez la base de temps à l'aide de l'application « Configuration matérielle ».
I/D n'influence que l'accu 1 -L, qui est maintenant l'accu 1.	phase 2	Les accumulateurs de S7 sont étendus à 32 bits.	Examinez les conséquences d'une opération indirecte d'incrémenta-tion ou de décrémentation dans le programme LIST.
Tenez compte de la nouvelle numérotation de blocs.	phase 2	L'appel indirect de bloc ne tient pas compte des nouveaux numéros de bloc (le numéro est prélevé dans le mot de memento ou de données ap-proprié).	Modifiez la logique dans S5 ou utili-sez des appels de bloc fixes.
OB23 et OB24 sont convertis en OB 122.	phase 2	Les OB23 et OB24 sont remplacés tous deux par l'OB122 dans S7.	Regroupez le contenu des OB23 et 24 dans un OB122 et effacez l'autre OB122.
L'OB a été interprété par l'AG115U comme un OB34.	phase 2	Selon la CPU employée, l'OB34 peut avoir des significations différentes.	Vérifiez que cet OB convient à votre programme.
'Masques DB S5' n'est plus utilisé pour le paramétrage de S7.	phase 1	Il y a MASK dans DW0 et dans DW1.	Paramétrez l'AP avec STEP 7.
L'opération de saut suivant l'opération 'B' ne peut être convertie (utilisez SPL).	phase 2	Une opération de substitution suivie d'une opération SPA ne peut être convertie automatiquement.	Remplacez l'opération par SPL dans le fichier LIST et revoyez le saut.
Le convertisseur ne définit pas les paramètres système	phase 2	DB1 et DX0 sont convertis mais ils n'ont plus la même signification que dans S5.	Effectuez le paramétrage du système dans la table de configuration.
Tenez compte des différentes opérations d'arrêt.	phase 2	Il n'est pas fait de différence entre STP, STS et STW.	Vérifiez le fichier programme.
RLG est mis à 1.	phase 2	Pour les opérations SU et RU de S5, le RLG est mis à 1 dans S7.	Ajoutez au besoin l'opération CLEAR.
Pré-en-tête manque.	phase 1	Pour les blocs FB et FX, les désigna-tions des repères de saut manquent ; pour les blocs DB et DX, ce sont les formats de données qui manquent.	Examinez si les étiquettes se trouvent dans un autre fichier.
Dans le cas d'un AG115U, changez-le en OB 100.	phase 2	L'OB21 de mise en route de S5 est converti automatiquement en OB101.	Si le programme S5 était exécuté sur un automate S5-115U, il faut transfor-mer l'OB101 en OB100.

Retouche du programme converti

Préparation

Les étapes de préparation suivantes sont nécessaires avant de retoucher le fichier source LIST généré :

- Imprimez les messages.
- Créez un programme S7 dans un projet via le gestionnaire de projets SIMATIC si vous ne l'avez pas déjà fait.
- Importez, à l'aide de la commande **Insertion > Source externe**, le programme source LIST généré dans le classeur "Sources" du programme S7 que vous avez créé.
- Ouvrez le fichier converti.

Exécution

Nous vous conseillons de retoucher le fichier source LIST généré de la manière suivante :

- Parcourez le programme en mode interactif et modifiez ou complétez les blocs d'organisation et les opérations S5 non convertibles (voir partie 1) en vous aidant des messages émis.

7.1 Modifications d'adresses

Ce sont surtout les modules d'entrées et de sorties qui sont touchés par les modifications d'adresses. Vous trouverez l'adresse des modules avec l'application HW Config.

7.1.1 Possibilités de modification des adresses

Réassignation dans S5

Vous pouvez, avant la conversion, adapter à l'aide de la fonction "Réassignation" de S5 les adresses d'opérandes individuels aux nouvelles adresses S7.

Réassignation dans S7

Il existe, dans le gestionnaire de projets SIMATIC, une fonction pour la réassignation automatique des blocs générés dans le fichier source.

Marche à suivre

1. Dans le gestionnaire de projets SIMATIC, sélectionnez les blocs de votre programme pour lesquels vous voulez procéder à des réassignations.
2. Ouvrez la table de réassignation avec la commande **Outils > Réassignation**.
3. Entrez dans la table les anciennes et les nouvelles adresses des différents opérandes, puis enregistrez.

Les blocs contiennent maintenant les adresses modifiées.

Modifications d'adresses dans le fichier source S7

Dans votre programme, adaptez les accès aux entrées et sorties ainsi que les accès directs de périphérie aux nouvelles adresses de modules dans S7.

Il vous suffit d'exécuter la commande **Edition > Remplacer** pour modifier les adresses absolues dans le fichier source S7.

Attention : Des changements involontaires peuvent se produire si les anciennes et nouvelles zones d'adresses se chevauchent.

Création d'un nouveau fichier source S7 (avec adressage symbolique)

Si vous désirez utiliser l'adressage symbolique, vous pouvez aussi exécuter la réassignation via la table des mnémoniques.

Condition préalable

Vous disposez déjà d'un programme compilé sans erreur et d'une table des mnémoniques contenant tous les noms symboliques pour les adresses absolues à modifier.

Marche à suivre

Procédez comme suit pour modifier les adresses :

- Ouvrez un bloc contenant des adresses à modifier et activez l'option **Adressage symbolique** dans l'onglet **Editeur** de la boîte de dialogue appelée via la commande **Outils > Paramètres**. Répétez cette opération pour tous les blocs contenant des adresses à modifier.
- Générez une source à partir de ces blocs avec la commande **Fichier > Générer la source**. Vous pouvez sélectionner les blocs concernés dans une boîte de dialogue après la saisie du nom de la source.

Tenez compte de la hiérarchie d'appel lors de la mise au point de la séquence de blocs. En effet, les blocs appelés doivent déjà exister ; il faut donc les insérer dans la source avant les blocs dans lesquels ils sont appelés.

Résultat : Les instructions avec adressage symbolique figurent dans la source générée.

- Vous pouvez maintenant exécuter la réassignation dans la table des mnémoniques : remplacez les adresses S5 qui ont changé par les nouvelles adresses S7.
- Les nouvelles adresses figurent dans les blocs après la compilation du fichier source.

7.2 Fonctions non convertibles

Les opérandes et opérations non convertibles figurent sous forme de commentaires dans le programme S7 généré ; vous devez procéder à leur retouche.

Deux méthodes de conversion vous sont proposées :

- Vous définissez pour ces opérations et opérandes – s'ils apparaissent dans le programme utilisateur – des macro-instructions spécifiques en LIST S7 qui seront utilisées lors de la conversion.
- Vous éditez les séquences d'instructions correspondantes dans le programme S7 résultant.

Le choix de la méthode dépend de la fréquence d'apparition de telles commandes dans votre programme utilisateur.

Les opérations et opérandes non convertibles sont détaillés aux paragraphes 3.11 et 3.12 qui contiennent également des conseils sur la façon de réaliser les fonctions non convertibles dans S7.

7.3 Conversion en cas d'adressage indirect

Le convertisseur S5/S7 convertit à l'aide d'instructions STEP 7 l'adressage indirect réalisé via B MW et B DW. La séquence d'instructions générée est souvent très importante, car il faut recalculer le pointeur STEP 5 en format STEP 7 ce qui nécessite une sauvegarde intermédiaire du contenu des accumulateurs et du mot d'état.

Si votre programme emploie souvent l'adressage indirect, il est intéressant d'effectuer une adaptation à l'adressage indirect dans STEP 7. En effet, une programmation adaptée permet de gagner beaucoup d'espace en mémoire.

L'énumération suivante montre comment le convertisseur S5/S7 convertit l'adressage indirect dans les différents cas.

Temporisations et compteurs

L'adressage indirect de temporisations et de compteurs est converti en adressage indirect en mémoire avec utilisation d'un mot de données locales temporaires.

Blocs

L'adressage indirect de blocs est converti en adressage indirect en mémoire avec utilisation d'un mot de données locales temporaires.

Il n'est pas possible de tenir compte des nouveaux numéros de blocs lors de la conversion ; il faut donc les corriger.

Opérandes

L'adressage indirect d'opérandes est converti par bits et par mots en adressage indirect par registre avec utilisation du registre d'adresse AR1 et de données locales temporaires comme mémoire intermédiaire pour le mot d'état, l'accumulateur 1 et l'accumulateur 2.

Adressage indirect via le registre BR

Ces instructions ne sont pas converties. L'adressage indirect doit être reprogrammé dans S7.

Autre mode d'adressage indirect

Il faut reprogrammer les instructions correspondantes dans S7.

L'adressage indirect est décrit en détails dans le paragraphe 3.13.4.

7.4 Accès directs à la mémoire

Certaines fonctions de STEP 5 accède à des adresses absolues en mémoire. Ces possibilités d'accès n'existent plus dans STEP 7.

STEP 5	STEP 7
Adressage d'opérandes dans des blocs de données dépassant la taille normale	Il est maintenant possible d'accéder avec des opérations normales (L, T...) aux adresses supérieures à 255 dans les blocs de données.
Adressage indirect avec le registre BR	Il est possible de réaliser cet adressage indirect en utilisant l'adressage indirect par registre (voir <i>Adressage indirect par registre</i> au paragraphe 3.13.4 ainsi que le manuel <i>LIST (232)</i>).
Transfert par blocs	Il existe une fonction système, SFC20 BLKMOV, pour le transfert par blocs. Les zones de mémoire à copier sont transmises aux paramètres de bloc. Si ces zones de mémoire doivent être variables, il est possible de transmettre aux paramètres des pointeurs ANY qui seront définis dans le programme utilisateur.

7.5 Transmission de paramètres

Commande S5 B<paramètres de bloc>

La commande B<paramètre formel de type B> s'exécute dans S5 selon le type de bloc transmis comme :

- “SPA bloc de code“
- ou bien comme “A DB bloc de données”.

Le type n'étant pas indiqué dans le paramètre formel, une conversion automatique est dans ce cas impossible. Vous devez donc rechercher dans votre programme toutes les occurrences d'opérations X avec paramètres de type B et convertir manuellement ces opérations.

Paramètres effectifs

Pour les blocs fonctionnels paramétrés, le convertisseur S5/S7 reprend sans modification les paramètres effectifs figurant aux appels de blocs. Vous devez vérifier et modifier si besoin est les adresses que vous avez éventuellement indiquées avec les paramètres effectifs.

Exemples :

- Indication d'un numéro de mot de données :
à convertir en adressage par octets
- Indication d'une adresse de périphérie :
il faut utiliser la nouvelle adresse de module.
- Transmission d'un bloc :
il faut indiquer le nouveau numéro de bloc.

7.6 Fonctions standard

Blocs fonctionnels standard de S5

Si votre programme S5 contient des blocs fonctionnels standard, ces derniers sont signalés :

- avant la conversion, par un astérisque dans la colonne “Défaut” de la boîte de dialogue “Conversion de fichiers S5” – [<nom>ST.S5D]
- et, après la conversion, par l’émission du message “Sortie interdite (n° du produit)”.

Le logiciel de base S7 livré comprend des fonctions S7 déjà converties (anciens blocs fonctionnels standard de S5) pour arithmétique à virgule flottante, fonctions de signalisation, fonctions intégrées, fonctions de base et fonctions mathématiques ; elles portent les noms FC61 à FC125 (voir paragraphe 3.9).

Intégration des FC

Procédez comme suit pour intégrer les fonctions S7 dans votre programme S7 :

1. Ouvrez le projet dans lequel vous souhaitez insérer les fonctions.
2. Ouvrez la bibliothèque standard du gestionnaire de projets SIMATIC contenant les fonctions S5 converties (StdLib30).
3. Copiez les fonctions S7 nécessaires de la bibliothèque standard dans votre programme S7.

Compilation

Vous devez compiler le programme converti et éventuellement retouché avec le compilateur LIST afin de le rendre exécutable. Procédez pour ce faire exactement comme pour un fichier de texte que vous venez de créer.

Vérification de la cohérence

La commande **Fichier > Vérifier la cohérence** vous permet de vérifier à tout moment la syntaxe et la cohérence du fichier source sans toutefois déclencher la génération des blocs. La vérification porte sur :

- la syntaxe,
- les mnémoniques
- et l'existence des blocs appelés dans le programme.

Vous obtenez ensuite un protocole indiquant le nom du fichier compilé, le nombre de lignes compilées ainsi que le nombre d'erreurs et d'avertissements.

Compilation du fichier source

La commande **Fichier > Compiler** vous permet de compiler votre fichier source afin de générer les blocs correspondants.

Un protocole s'affiche après la compilation ; il indique les erreurs éventuelles comme après la vérification de cohérence. Si vous avez programmé plusieurs blocs dans un fichier source, seuls ceux sans erreur sont compilés et sauvegardés.

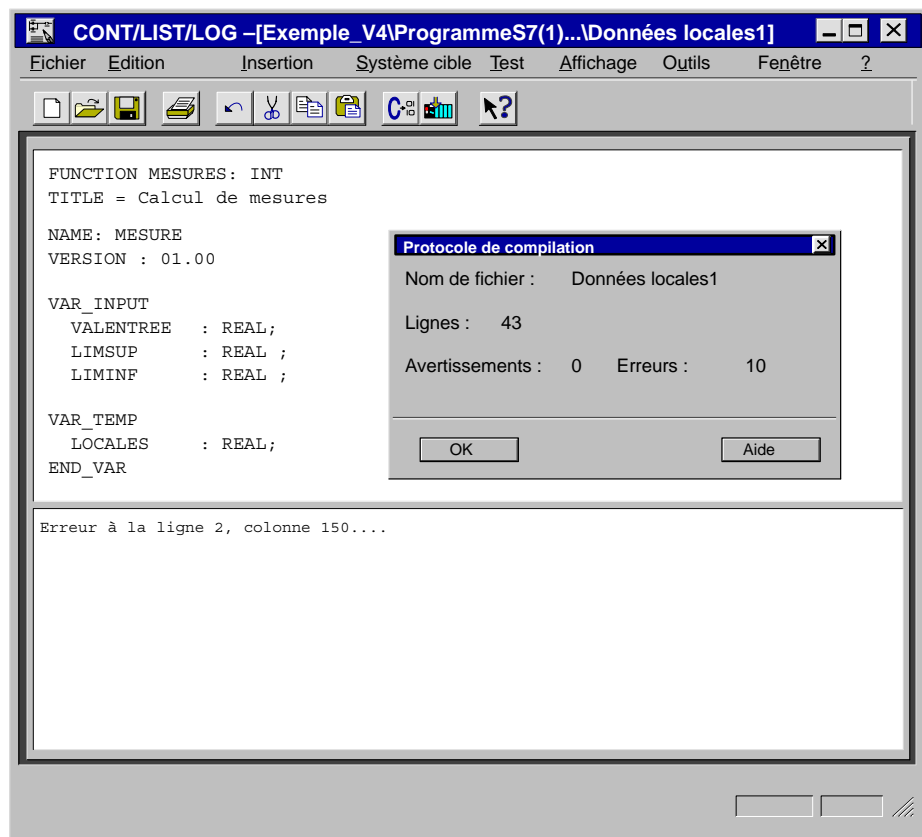


Figure 8-1 Vérification de la cohérence et compilation de fichiers source

Correction des erreurs

Si votre programme converti contient des erreurs ou des avertissements, ceux-ci sont énumérés après la vérification de cohérence ou la compilation dans une sous-fenêtre en dessous du fichier source. La cause de l'erreur est également précisée. Lorsque vous sélectionnez un message d'erreur, l'emplacement correspondant du fichier source s'affiche dans la fenêtre supérieure. Cela vous permet de remédier rapidement aux erreurs éventuelles.

Vous pouvez procéder aux corrections et aux modifications en mode de substitution que vous activez à l'aide de la touche d'insertion.

Exemple d'application

Ce chapitre présente, à l'aide d'un exemple, quatre domaines constituant des nouveautés de S7 ou réalisés différemment dans S5 :

- le traitement des valeurs analogiques,
- les données locales,
- l'évaluation des informations de déclenchement des blocs d'organisation,
- le transfert par blocs.

Dans cet exemple, un moteur avec marche à droite et à gauche est commandé par un module d'entrées/sorties TOR. Un module d'entrées analogiques lit le régime qui peut être commandé via un module de sorties analogiques. Pour cet exemple, les modules TOR et analogiques doivent disposer de la fonction de diagnostic.

Installation

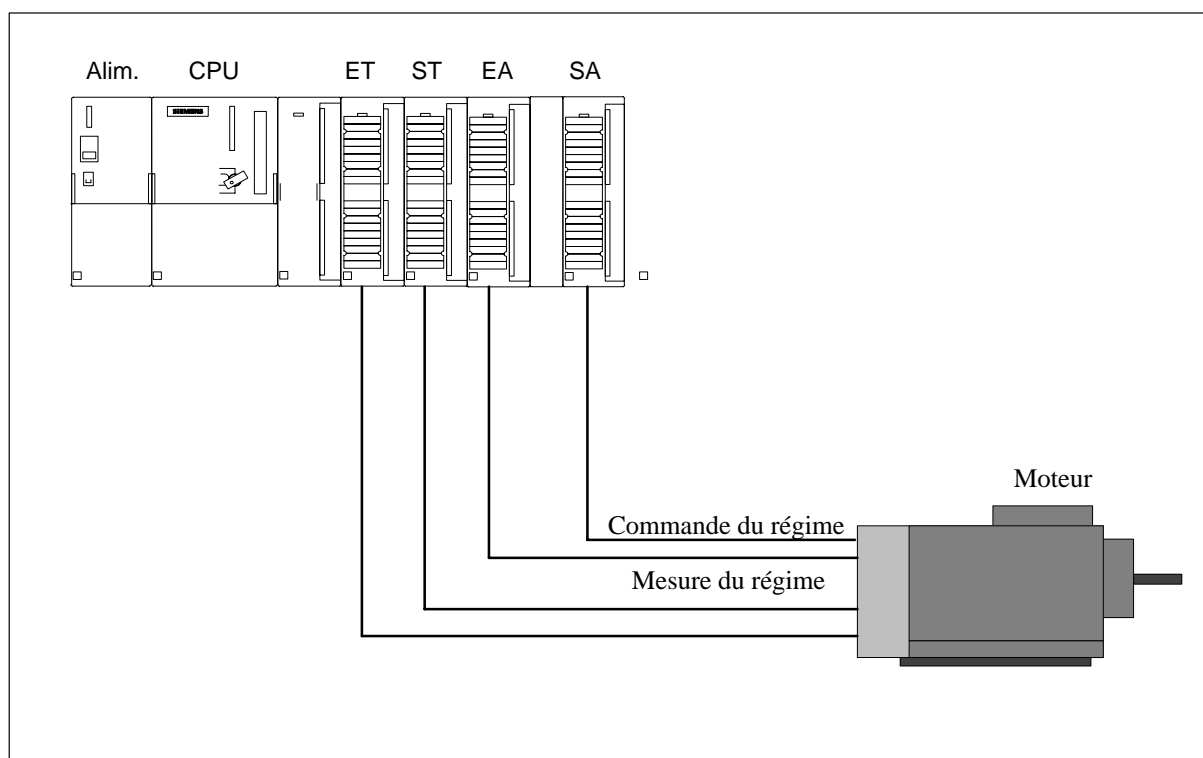


Figure 9-1 Installation

9.1 Traitement des valeurs analogiques

Conversion de valeurs analogiques

La CPU ne traite les valeurs analogiques que sous forme numérique.

Les modules d'entrées analogiques convertissent le signal analogique du processus en valeur numérique.

Quant aux modules de sorties analogiques, ils convertissent la valeur de sortie numérique en un signal analogique.

Représentation des valeurs analogiques dans S5

Tableau 9-1 Exemple pour la carte d'entrées analogiques 6ES5 460-7LA13

Résolution	Valeur analogique															
Numéro de bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Poids des bits	S	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	A	E	D

Dans les cartes de sorties analogiques, les valeurs sont représentées par leur complément à deux sur 12 bits.

Les cartes d'entrées analogiques peuvent évaluer la valeur soit comme nombre de 12 bits avec signe, soit comme complément à deux de 13 bits.

Le bit D signale les débordements.

Le bit d'erreur E est mis à 1 en cas d'erreur (par exemple, rupture de fil si cette erreur a été paramétrée).

Le bit A est le bit d'activité. Il est à 0 lorsque la valeur affichée est correcte.

Représentation des valeurs analogiques dans S7

La valeur analogique numérisée est la même pour les valeurs d'entrée et de sortie pour une même plage nominale.

Les valeurs analogiques sont représentées par leur complément à deux.

Tableau 9-2 Exemple pour modules d'entrées analogiques dans S7

Résolution	Valeur analogique															
Numéro de bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Poids des bits	S	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Le signe (S) de la valeur analogique se trouve toujours dans le bit numéro 15. 0 correspond à une valeur positive et 1 à une valeur négative.

Il n'existe pas de bit d'erreur dans S7 ; la valeur W#16#7FFF est émise en cas d'erreur.

Une alarme de diagnostic peut être déclenchée en cas d'erreur pour les modules aptes au diagnostic. Vous paramétrez cette alarme dans l'application HW Config.

Si la résolution d'un module analogique est inférieure à 15 bits, la valeur analogique est cadrée à gauche dans les données utiles. Les positions de poids faible non occupées prennent l'état de signal 0.

Exemple

Dans l'exemple, un module d'entrées analogiques de 14 bits de résolution lit le régime du moteur. La mesure est une valeur bipolaire (par exemple, plage de mesure +/- 10 V).

Les limites supérieure et inférieure sont transmises comme paramètres.

Une vérification est effectuée afin de s'assurer que la valeur analogique ne franchit pas la limite supérieure ou inférieure. Si la valeur lue se situe en dehors de la plage autorisée, une erreur est signalée par la mise à zéro du bit de résultat binaire et la valeur 0 est émise. Si la valeur lue est correcte, elle est émise.

La valeur analogique est émise via la valeur de retour RET_VAL de la fonction. RET_VAL représente une valeur de fonction ; il s'agit d'une nouveauté de S7.

```

FUNCTION FC1: REAL
TITLE = Traitement de valeurs analogiques
NAME:      ANALOG
VERSION:   01.00

VAR_INPUT
    VALENTREE      : INT;          // Valeur d'entrée
    LIMSUP          : REAL;        // Limite supérieure pour la valeur analogique
    LIMINF          : REAL;        // Limite inférieure pour la valeur analogique
END_VAR

BEGIN
NETWORK
TITLE = Vérification des limites supérieure et inférieure

    O(
    L    VALENTREE;    // valeur d'entrée > limite supérieure
    L    +27648;
    >I;
    );
    O(
    L    VALENTREE;    // ou
    L    -27648;        // valeur d'entrée < limite inférieure
    <I;
    );
    NOT;
    L    0;
    SPBNB ENDE;        // Pas d'autre traitement en cas de dépassement haut ou bas
                        // valeur en retour = 0 et BIE = 0.
                        // BIE = 1 s'il n'y a pas de dépassement de limite.

NETWORK
TITLE = Conversion de la valeur numérique en régime

    L    LIMSUP;        // Formule pour la conversion de VALENTREE en régime
    L    LIMINF;        // valeur analogique = (LIMSUP - LIMINF) * VALENTREE
    -R;                // / (55296 (nombre d'unités))
    L    VALENTREE;
    ITD;                // Convertir valeur en nombre à virgule flottante
    DTR;
    *R;
    L    55296.0;
    /R;
ENDE:    T            RET_VAL;

    BE;
END_FUNCTION

```

Figure 9-2 Traitement de valeurs analogiques

9.2 Données locales temporaires

Les données locales temporaires servent de mémoire intermédiaire et remplacent donc les mementos banalisés de S5. Vous pouvez les utiliser dans tous les blocs de code après le traitement desquels elles sont perdues. Les données se trouvent dans la pile des données locales (pile L).

Exemple 1

L'exemple 1 emploie des données locales temporaires adressées **symboliquement** comme mémoire intermédiaire. Un régime donné est converti en valeur de mesure numérisée pour un module de sorties analogiques de 14 bits de résolution. La mesure est une valeur bipolaire (par exemple, plage de mesure +/- 10 V).

Les limites supérieure et inférieure sont transmises comme paramètres.

La mesure est émise via la valeur de retour RET_VAL de la fonction. Chaque fonction peut fournir en option une valeur en retour dont le type de données est indiqué avec la désignation de la fonction. Si la fonction ne doit pas renvoyer de valeur, il faut indiquer VOID comme type de données.

```

FUNCTION FC2: INT
TITLE = Calcul de mesures
NAME:      MESURE
VERSION:   01.00

VAR_INPUT
    VALENTREE    : REAL;    // Valeur d'entrée (courant)
    LIMSUP       : REAL;    // Limite supérieure
    LIMINF       : REAL;    // Limite inférieure
END_VAR

VAR_TEMP
    LOCALES      : REAL;    // Données locales pour résultat intermédiaire
END_VAR

BEGIN
NETWORK
TITLE = Calcul de la mesure

    L    VALENTREE;          // Formule pour le calcul des unités :
    L    55296.0;            // mesure = VALENTREE
    *R;                      //      * 55296 (nombre d'unités)
                              //      / (LIMSUP - LIMINF)
    T    LOCALES;            // Sauvegarder le résultat intermédiaire
    L    LIMSUP;             // dans les données locales
    L    LIMINF;
    -R;
    L    LOCALES;
    TAK;
    /R;
    RND;                     // Convertir nombre à virgule flottante en nombre entier
    T    RET_VAL;

END_FUNCTION

```

Figure 9-3 Calcul de mesures

Exemple 2

L'exemple 2 emploie des données locales comme des mementos banalisés S5 adressés **absolument**. On réalise une commande de moteur avec marche à droite et à gauche. Dans l'exemple, l'octet d'entrée et l'octet de sortie sont copiés dans la zone des données locales. Vous devez réserver une zone dans la pile L pour l'utilisation des données locales temporaires, car cette pile sert également à l'éditeur de programme. Vous trouverez les adresses absolues des données locales dans la section de déclaration du bloc. Les bits de données locales sont combinés entre eux dans le programme. Il en résulte des signaux de sortie qui sont récrits, à la fin du bloc, des données locales dans l'octet de sortie. Les adresses des octets d'entrée et de sortie sont paramétrables.

Nota

L'insertion de nouvelles variables avant les données locales existantes entraîne le décalage des données locales suivantes.

Tableau 9-3 Affectation des entrées, sorties et données locales

Adresse	Données locales	Désignation	Description
E n.0	L 0.0	MARCHE	Commutateur de mise en marche
E n.1	L 0.1	ARRET	Arrêter le moteur
E n.2	L 0.2	ARRET_URG	Commutateur d'arrêt d'urgence
E n.3	L 0.3	MOTEUR_DROITE	Activer la marche à droite du moteur
E n.4	L 0.4	MOTEUR_GAUCHE	Activer la marche à gauche du moteur
E n.5	L 0.5	COMMFIN_DROITE	Commutateur de fin pour la droite
E n.6	L 0.6	COMMFIN_GAUCHE	Commutateur de fin pour la gauche
E n.7	L 0.7	-	Libre
A m.0	L 1.0	PRET	Moteur prêt
A m.1	L 1.1	MARCHE_DROITE	En marche à droite
A m.2	L 1.2	MARCHE_GAUCHE	En marche à gauche
A m.3	L 1.3	POSITION_ATTEINTE	Position atteinte

Fonctionnement

La tension est appliquée avec le commutateur de mise en marche. Le moteur est alors prêt, ce qui est signalé par la sortie PRET. Les boutons-poussoirs MOTEUR_DROITE et MOTEUR_GAUCHE permettent de faire marcher le moteur dans la direction désirée. Le moteur ne peut être piloté que dans une direction à la fois ; il faut l'arrêter (ARRET) avant de changer de direction. Le moteur s'arrête en cas d'actionnement d'un commutateur de fin. L'arrêt d'urgence entraîne également l'arrêt du moteur, qui ne peut être remis en marche qu'après remise à zéro du commutateur d'arrêt d'urgence.

```

FUNCTION FC3: VOID
TITLE = Commande de moteur
NAME:      MOTEUR
VERSION:   01.00

VAR_INPUT
    OCTENTREE          : BYTE; // Octet d'entrée
END_VAR

VAR_IN_OUT
    OCTSORTIE          : BYTE; // Octet de sortie
END_VAR

VAR_TEMP
    IMAGE_OCTENTREE     : BYTE; // Image de l'octet d'entrée
    IMAGE_OCTSORTIE     : BYTE; // Image de l'octet de sortie
END_VAR

BEGIN
NETWORK
TITLE = Commande de moteur

    L    OCTENTREE;      // Copier l'octet d'entrée dans la zone des données locales
    T    IMAGE_OCTENTREE;
    L    OCTSORTIE;      // Copier l'octet de sortie dans la zone des données locales
    T    IMAGE_OCTSORTIE;

    ON    L0.0;          // Moteur pas en marche (pas de tension)
    ON    L0.2;          // ou commutateur d'arrêt d'urgence actionné
    R    L1.0;          // => Mettre à 0 Moteur prêt
    R    L1.1;          // => Mettre à 0 Commande du moteur
    R    L1.2;
    R    L1.3;          // => Mettre à 0 Position atteinte
    SPB   ENDE;          // => Pas d'autre évaluation des signaux

    U    L0.0;          // Moteur en marche
    S    L1.0;          // => Mettre à 1 Moteur prêt

    U    L0.3;          // Commande du moteur pour marche à droite
    UN    L0.4;          // Verrouillage : pas de commande pour marche à gauche
    UN    L1.2;          // et marche à gauche inactive
    FP    M0.0;          // Constituer le front montant
    S    L1.1;          // Alors : activer la marche à droite
    R    L1.3;          // Mettre à 0 Position atteinte

    U    L0.4;          // Commande du moteur pour marche à gauche
    UN    L0.3;          // Verrouillage : pas de commande pour marche à droite
    UN    L1.1;          // et marche à droite inactive
    FP    M0.1;          // Constituer le front montant
    S    L1.2;          // Alors : activer la marche à gauche
    R    L1.3;          // Mettre à 0 Position atteinte

```

Suite à la page suivante

Figure 9-4 Fonction de commande de moteur

```
O(;  
U    L0.5;          // Commutateur de fin pour la droite atteint  
U    L1.1;          // et marche à droite active  
);  
O(;  
U    L0.6;          // Commutateur de fin pour la gauche atteint  
U    L1.2;          // et marche à gauche active  
);  
S    L1.3;          // => Mettre à 1 Position atteinte  
O    L0.1;          // Arrêter le moteur actionné  
O    L1.3;          // ou Position atteinte  
R    L1.1;          // => Mettre à 0 la commande du moteur  
R    L1.2;  
  
ENDE:    L          IMAGE_OCTSORTIE;          // Copier les données locales  
                                                // dans l'octet de sortie  
        T          OCTSORTIE;  
  
END_FUNCTION
```

Figure 9-5 Fonction de commande de moteur (suite)

9.3 Evaluation des informations de déclenchement de l'OB82 (alarme de diagnostic)

Informations de déclenchement

Lorsque le système d'exploitation appelle les blocs d'organisation, il met à votre disposition dans la pile des données locales des informations de déclenchement uniformes au niveau du système. Ces informations, d'une longueur de 20 octets, sont disponibles après le lancement du traitement de l'OB.

Informations de déclenchement de l'OB82

Les informations de déclenchement de l'OB d'alarme de diagnostic – présentées en détail dans le manuel de référence **/235/** – contiennent l'adresse de base logique ainsi que quatre octets d'informations de diagnostic. Vous trouverez des modèles pour la table de déclaration de variables correspondante dans la bibliothèque standard "StdLib30" sous "StdOBs".

Les modules TOR posent une demande d'alarme de diagnostic à la CPU (aussi bien pour des événements entrants que pour des événements sortants) si vous avez validé l'alarme de diagnostic lors de la configuration du matériel. Le système d'exploitation appelle alors l'OB82.

Vous pouvez inhiber ou retarder, puis à nouveau valider l'appel de l'OB d'alarme de diagnostic à l'aide des fonctions système SFC39 à 42. Reportez-vous au manuel de référence **/235/** pour plus d'informations.

Exemple

La tension auxiliaire externe est évaluée dans le programme-exemple. Si elle est interrompue, le bit TENSION_EXT_MANQUE est mis à 1 dans le bloc de données DB82, DB_DIAG. L'adresse du module et l'instant sont également sauvegardés. Ces informations peuvent être traitées dans la suite du programme.

Avant la compilation de la source LIST, il faut inscrire le mnémonique du bloc de données DB82, DB_DIAG, dans la table des mnémoniques.

```

DATA_BLOCK DB_DIAG
TITLE = Données de diagnostic
NAME:      DB_DIAG
VERSION:    01.00

STRUCT
    MDL_ADDR      : INT;          // Adresse du module
    TENSION_EXT_MANQUE : BOOL;    // Bit d'erreur : tension externe manque
    DATE_TIME     : DATE_AND_TIME; // Date et heure auxquelles l'alarme
                                    // de diagnostic a été déclenchée
    SFC_RET_VAL    : INT;          // Code renvoyé par la SFC BLKMOV
END_STRUCT;

BEGIN
END_DATA_BLOCK

ORGANIZATION_BLOCK OB82
TITLE = Alarme de diagnostic
NAME:      Diagnostic
VERSION:    01.00

VAR_TEMP
    OB82_EV_CLASS      : BYTE; // Classe d'événement et indicateurs :
                                // B#16#38 : événement sortant
                                // B#16#39 : événement entrant

    OB82_FLT_ID         : BYTE; // Code d'erreur (B#16#42)
    OB82_PRIORITY       : BYTE; // Classe de priorité 26 ou 28
    OB82_OB_NUMBR       : BYTE; // Numéro d'OB
    OB82_RESERVED_1     : BYTE; // Réservé
    OB82_IO_FLAG        : BYTE; // Module d'entrées : B#16#54
                                // Module de sorties : B#16#55

    OB82_MDL_ADDR       : INT;    // Adresse de base logique du module
                                // dans lequel l'erreur s'est produite

    OB82_MDL_DEFECT      : BOOL;   // Module défaillant
    OB82_INT_FAULT       : BOOL;   // Erreur interne
    OB82_EXT_FAULT       : BOOL;   // Erreur externe
    OB82_PNT_INFO        : BOOL;   // Erreur de voie
    OB82_EXT_VOLTAGE     : BOOL;   // Tension auxiliaire externe manque
    OB82_FLD_CONNCTR     : BOOL;   // Connecteur en face avant manque
    OB82_NO_CONFIG       : BOOL;   // Module non paramétré
    OB82_CONFIG_ERR      : BOOL;   // Paramètres erronés dans le module
    OB82_MDL_TYPE        : BYTE;   // Bits 0-3 : Classe du module
                                // Bit 4 : Informations sur voie présentes
                                // Bit 5 : Informations utilisateur présentes
                                // Bit 6 : Alarme de diagnostic de remplaçant
                                // Bit 7 : Réserve

    OB82_SUB_MDL_ERR     : BOOL;   // Cartouche utilisateur erronée ou manquante
    OB82_COMM_FAULT      : BOOL;   // Communication défaillante
    OB82_MDL_STOP        : BOOL;   // Etat de fonctionnement (0 : Marche, 1 : Arrêt)
    OB82_WTCH_DOG_FLT    : BOOL;   // Réaction de la surveillance de temps
    OB82_INT_PS_FLT      : BOOL;   // Tension d'alimentation interne au module
                                // défaillante

    OB82_PRIM_BATT_FLT   : BOOL;   // Pile épuisée
    OB82_BCKUP_BATT_FLT  : BOOL;   // Sauvegarde entière défaillante
    OB82_RESERVED_2     : BOOL;   // Réservé
    OB82_RACK_FLT        : BOOL;   // Châssis/profilé support défaillant
    OB82_PROC_FLT        : BOOL;   // Processeur défaillant
    OB82_EPROM_FLT       : BOOL;   // Erreur d'EPROM
    OB82_RAM_FLT         : BOOL;   // Erreur de RAM

```

Suite en page suivante

Figure 9-6 Evaluation des données de diagnostic

```

OB82_ADU_FLT      : BOOL;          // Erreur CAN/CNA
OB82_FUSE_FLT     : BOOL;          // Fusible défaillant
OB82_HW_INTR_FLT  : BOOL;          // Perte d'alarme de processus
OB82_RESERVED_3   : BOOL;          // Réserve
OB82_DATE_TIME    : DATE_AND_TIME; // Date et heure auxquelles l'OB
                                     // a été requis

END_VAR

BEGIN
NETWORK
TITLE = Alarme de diagnostic

    L      OB82_MDL_ADDR;            // Sauvegarder l'adresse du module
    T      DB_DIAG.MDL_ADDR;

    L      OB82_EV_CLASS;            // Classe d'événement = B#16#38 :
    L      B#16#38;                  // Événement sortant
    ==I;
    SPB    SORT;

    U      OB82_EXT_VOLTAGE;          // Événement entrant :
                                     // Vérifier si la tension auxiliaire
                                     // externe manque
    S      DB_DIAG.TENSION_EXT_MANQUE; // Mettre le bit à 1
    SPA    HEUR;

SORT:      U      OB82_EXT_VOLTAGE;  // Événement sortant :
                                     // Tension auxiliaire externe
                                     // de nouveau présente
    R      DB_DIAG.TENSION_EXT_MANQUE; // Mettre le bit à 0

NETWORK

TITLE = Sauvegarder date et heure
HEUR:      CALL    SFC 20(           // SFC BLKMOV
    SRCBLK :=OB82_DATE_TIME,         // Sauvegarder les date et heure
    RET_VAL:=DB_DIAG.SFC_RET_VAL,    // auxquelles l'alarme de
    DSTBLK :=DB_DIAG.DATE_TIME);     // diagnostic a été requise

END_ORGANIZATION_BLOCK

```

Figure 9-7 Evaluation des données de diagnostic (suite)

9.4 Transfert par blocs

La fonction système SFC20 BLKMOV pour le transfert par blocs permet de copier le contenu d'une zone de mémoire (tableau source) dans une autre zone de mémoire (tableau cible).

Il est possible de copier toutes les entrées, toutes les sorties, tous les mementos et toutes les données.

Paramètres

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
SRCBLK	INPUT	ANY	E, A, M, D, L	Indication de la zone de mémoire à copier (tableau source)
RET_VAL	OUTPUT	INT	E, A, M, D, L	La valeur en retour contient un code d'erreur si une erreur se produit pendant le traitement.
DSTBLK	OUTPUT	ANY	E, A, M, D, L	Indication de la zone de mémoire où copier (tableau cible)

Nota

Les tableaux source et cible ne doivent pas se chevaucher. Si le tableau cible indiqué est plus grand que le tableau source, le volume de données copié dans le tableau cible correspondra à celui du tableau source.

Si le tableau cible indiqué est plus petit que le tableau source, le volume de données copié ne dépassera pas ce que peut contenir le tableau cible.

Si vous ne voulez pas affecter de pointeurs constants aux paramètres pour les zones source et cible de la SFC20 BLKMOV mais désirez indiquer des zones variables, il vous suffit d'utiliser des variables temporaires de type ANY.

**Pointeur ANY
comme pointeur
de données**

Les tableaux suivants présentent la structure du pointeur ANY.

Tableau 9-4 Pointeur ANY

Octet n	Octet n+1	Octet n+2	Octet n+3	Octet n+4	Octet n+5	Octet n+6	Octet n+7	Octet n+8	Octet n+9
B#16#10	Type (voir tableau 9-5)	Longueur		Numéro de bloc de données pour bloc de données		Pointeur de zone (voir figure 9-8)			

Tableau 9-5 Type (octet n+1)

Valeur	01	02	03	04	05	06	07
Type	BOOL	BYTE	CHAR	WORD	INT	DWORD	DINT
Valeur	08	09	0A	0B	0C	0E	13
Type	REAL	DATE	TOD	TIME	S5TIME	DT	STRING

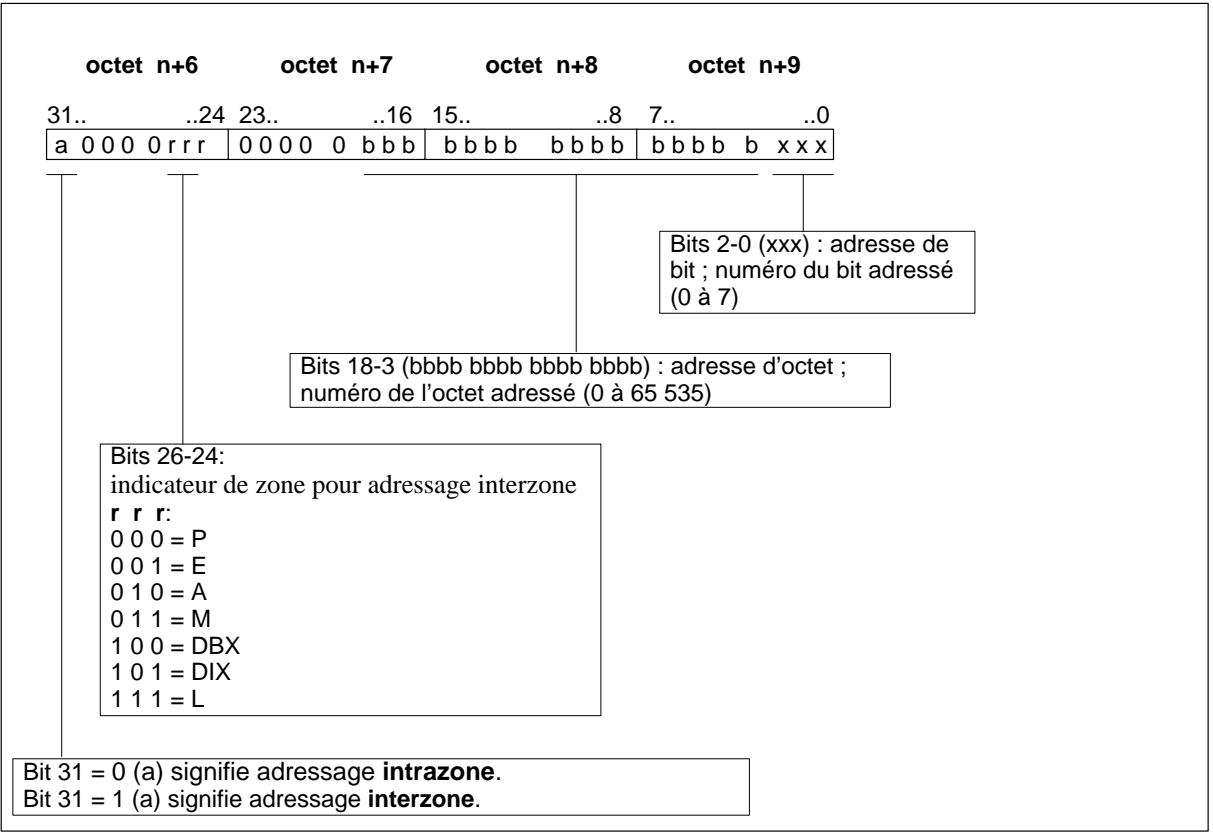


Figure 9-8 Pointeur de zone (octets n+6 à n+9)

Exemple

L'exemple contient une fonction permettant de copier, à l'aide de la fonction système SFC20 BLKMOV, des zones de données dans des blocs de données. Vous pouvez transmettre aux paramètres des zones source et cible variables.

Principe

Cette fonction contient deux pointeurs ANY dans la zone des données locales : un pour la zone source et un pour la zone cible. En règle générale, le type de données ANY n'est autorisé que pour des variables dans la zone des données locales.

Dans la fonction, les pointeurs ANY sont définis selon la structure présentée ci-avant et transmis aux paramètres lors de l'appel de la SFC20 BLKMOV.

```

FUNCTION FC4: INT
  TITLE = Copie de zones de données
  NAME:      COPIE
  VERSION:   01.00

  VAR_INPUT
    NODB_SOURCE      : INT;    // Numéro de DB de la zone source
    DEBUT_SOURCE     : INT;    // Numéro de DW du début de la zone source
    LONG_SOURCE      : INT;    // Longueur de la zone source en octets
    NODB_CIBLE       : INT;    // Numéro de DB de la zone cible
    DEBUT_CIBLE      : INT;    // Numéro de DW du début de la zone cible
    LONG_CIBLE       : INT;    // Longueur de la zone cible en octets
  END_VAR

  VAR_TEMP
    POINTEUR_SOURCE: ANY;     // Pointeur ANY pour la zone source
    POINTEUR_CIBLE : ANY;     // Pointeur ANY pour la zone cible
  END_VAR

  BEGIN
  NETWORK
  TITLE = Définir le pointeur source

    L   P##POINTEUR_SOURCE; // Charger l'adresse du pointeur sur la zone source
    LAR1;                    // dans le registre d'adresse 1
    L   W#16#1002;           // Ecrire l'indicateur pour la zone de données
    T   LW[AR1, P#0.0];      // dans le pointeur ANY pour la source
    L   NODB_SOURCE;         // Ecrire le no de DB dans le pointeur ANY pour la source
    T   LW[AR1, P#4.0];
    L   DEBUT_SOURCE;        // Convertir le début de la zone de données
    SLD 3;                   // en format de pointeur
    OD   DW#16#84000000;     // Le combiner avec l'indicateur de zone
    T   LD[AR1, P#6.0];      // et l'écrire dans le pointeur ANY pour la source
    L   LONG_SOURCE;         // Ecrire la longueur de la zone de données
    T   LW[AR1, P#2.0];      // dans le pointeur ANY pour la source
  
```

Suite en page suivante

Figure 9-9 Copie de zones de données

```

NETWORK
TITLE = Définir le pointeur cible

L      P##POINTEUR_CIBLE; // Charger l'adresse du pointeur sur la zone cible
LAR1; // dans le registre d'adresse 1
L      W#16#1002; // Ecrire l'indicateur pour la zone de données
T      LW[AR1, P#0.0]; // dans le pointeur ANY pour la cible
L      NODB_CIBLE; // Ecrire le no de DB dans le pointeur ANY pour la cible
T      LW[AR1, P#4.0];
L      DEBUT_CIBLE; // Convertir le début de la zone de données
SLD    3; // en format de pointeur
OD     DW#16#84000000; // Le combiner avec l'indicateur de zone
T      LD[AR1, P#6.0]; // et l'écrire dans le pointeur ANY pour la cible
L      LONG_CIBLE; // Ecrire la longueur de la zone de données
T      LW[AR1, P#2.0]; // dans le pointeur ANY pour la cible

NETWORK
TITLE = Copier les données
CALL   SFC 20( // Copier les données avec la SFC BLKMOV (transfert par blocs)
SRCBLK :=POINTEUR_SOURCE, // Pointeur sur le tableau source
RET_VAL:= RET_VAL, // Code renvoyé par la SFC BLKMOV
DSTBLK :=POINTEUR_CIBLE); // Pointeur sur le tableau cible
END_FUNCTION

```

Figure 9-10 Copie de zones de données (suite)

9.5 Appel des exemples

Ce paragraphe contient la table des mnémoniques, les blocs de données nécessaires à la définition des paramètres de bloc et le bloc d'organisation OB1 avec l'appel des fonctions décrites précédemment.

Tableau 9-6 Table des mnémoniques

Mnémonique	Adresse	Type de données	Commentaire
DB_DIAG	DB82	DB82	Bloc de données de diagnostic
DB_MESURES	DB100	DB100	Bloc de données pour mesures
DB_MOTEUR_1	DB110	DB110	Bloc de données pour Moteur 1
ERREUR	MW100	WORD	Valeur renvoyée par la fonction FC4 pour le transfert par blocs

```

DATA_BLOCK DB_MESURES
TITLE = Mesures
NAME:      DB_MES
VERSION:   01.00
STRUCT
    VAL_ANALOG_1    : REAL;      // Valeur analogique de FC1
    VAL_ANALOG_2    : REAL;      // Valeur analogique de FC2
    VAL_NUMER_2     : INT;       // Mesure numérisée de FC2
END_STRUCT;
BEGIN
END_DATA_BLOCK

DATA_BLOCK DB_MOTEUR_1
TITLE = Données pour moteur
NAME:      DB_MOT_1
VERSION:   01.00
STRUCT
    MOT_COMMANDE    : WORD;      // Commande de moteur 1
    REGIME          : REAL;      // Régime de moteur 1
    TEMPERATURE     : REAL;      // Température de moteur 1
    COURANT         : REAL;      // Consommation en courant de moteur 1
END_STRUCT;
BEGIN
END_DATA_BLOCK

ORGANIZATION_BLOCK OB1
TITLE = Appel cyclique
NAME:      CYCLE
VERSION:   01.00
VAR_TEMP
    STARTINFO: ARRAY [1..20] OF BYTE;
END_VAR
BEGIN
NETWORK
TITLE = Appel des fonctions
CALL FC 1 (
    VALENTREE      := EW 0,      // Appel de la fonction pour le
                                // traitement de valeurs analogiques
    LIMSUP         := +10.0,     // Plage de mesure : +/- 10 V
    LIMINF         := -10.0,
    RET_VAL        := DB_MESURES.VAL_ANALOG_1;
                                // RET_VAL = valeur analogique
                                // Appel de la fonction pour le
                                // calcul de la mesure numérisée
CALL FC 2 (
    VALENTREE      := DB_MESURES.VAL_ANALOG_2, //
    LIMSUP         := +10.0,     // Plage de mesure : +/- 10 V
    LIMINF         := -10.0,
    RET_VAL        := DB_MESURES.VAL_NUMER_2;
                                // RET_VAL = mesure numérisée
                                // Appel de la fonction pour la commande du moteur
CALL FC 3 (
    OCTENTREE      := EB 4,
    OCTSORTIE      := AB 8);
CALL FC 4 (
                                // Appel de la fonction pour le transfert par blocs
    DEBUT_SOURCE   := 0,        // A partir de l'octet DBB0
    LONG_SOURCE    := 8,        // Longueur : 4 octets
    NODB_CIBLE     := 110,     // Cible : DB110
    DEBUT_CIBLE    := 2,        // A partir de l'octet DBB6
    LONG_CIBLE     := 8,        // Longueur : 4 octets
    RET_VAL        := ERREUR;   // RET_VAL = code d'erreur de la SFC20 BLKMOV
END_ORGANIZATION_BLOCK

```

Figure 9-11 DB et OB1 utilisés dans l'exemple

Annexes

Listes d'opérandes et
d'opérations

A

Bibliographie

B

Listes d'opérandes/opérations

A.1 Opérandes

Opérandes convertibles

Les opérandes suivants sont convertis.

Tableau A-1 Opérandes convertibles

LIST de S5 (SIMATIC)	LIST de S5 (CEI)	LIST de S7 (SIMATIC)	LIST de S7 (CEI)
A	Q	A	Q
AB	QB	AB	QB
AD	QD	AD	QD
AW	QW	AW	QW
BF	BN	(vide)	(vide)
D	D	DBX	DBX
DW	DW	DBW	DBW
DD	DD	DBD	DBD
DR	DR	DBB	DBB
DL	DL	DBB	DBB
E	I	E	I
EB	IB	EB	IB
ED	ID	ED	ID
EW	IW	EW	IW
M	F	M	M
MB	FY	MB	MB
MD	FD	MD	MD
MW	FW	MW	MW
PW	PW	PEW/PAW	PIW/PQW
PY	PY	PEB/PAB	PIB/PQB
QB	OY	PEB/PAB	PIB/PQB
QW	OW	PEW/PAW	PIW/PQW

Tableau A-1 Opérandes convertibles (suite)

LIST de S5 (SIMATIC)	LIST de S5 (CEI)	LIST de S7 (SIMATIC)	LIST de S7 (CEI)
S	S	M	M
SD	SD	MD	MD
SW	SW	MW	MW
SY	SY	MB	MB
T	T	T	T
Z	C	Z	C
= <paramètre formel>	= <paramètre formel>	# <paramètre formel>	# <paramètre formel>

Opérandes non convertibles

Le tableau A-2 présente les opérandes qui **ne peuvent pas être convertis**.

Tableau A-2 Opérandes non convertibles

LIST de S5 (SIMATIC)	LIST de S5 (CEI)
A1	A1
A2	A2
BA	RI
BB	RJ
BR	BR
BS	RS
BT	RT
CB	CY
CD	CD
CW	CW
GB	GY
GD	GD
GW	GW
SA	SA

A.2 Opérations

Opérations convertibles, sans opérande

Le tableau A-3 présente toutes les opérations de S5 (sans opérande) en LIST qui sont converties automatiquement en LIST de S7 par le convertisseur.

Tableau A-3 Opérations convertibles (sans opérande)

LIST de S5 (SIMATIC)	LIST de S5 (CEI)	LIST de S7 (SIMATIC)	LIST de S7 (CEI)
AF	RA	CALL SFC42	CALL SFC42
AS	IA	CALL SFC41	CALL SFC41
BEA	BEU	BEA	BEU
BEB	BEC	BEB	BEC
+D	+D	+D	+D
-D	-D	-D	-D
!=D	!=D	==D	==D
><D	><D	<>D	<>D
>D	>D	>D	>D
>=D	>=D	>=D	>=D
<D	<D	<D	<D
<=D	<=D	<=D	<=D
DED	DED	BTD	BTD
DEF	DEF	BTI	BTI
DUD	DUD	DTB	DTB
DUF	DUF	ITB	ITB
ENT	ENT	ENT	ENT
+F	+F	+I	+I
-F	-F	-I	-I
:F	:F	/I	/I
xF	xF	*I	*I
!=F	!=F	==I	==I
><F	><F	<>I	<>I
>F	>F	>I	>I
>=F	>=F	>=I	>=I
<F	<F	<I	<I
<=F	<=F	<=I	<=I
FDG	FDG	DTR	DTR
+G	+G	+R	+R
-G	-G	-R	-R
:G	:G	/R	/R
xG	xG	*R	*R
!=G	!=G	==R	==R
><G	><G	<>R	<>R
>G	>G	>R	>R

Tableau A-3 Opérations convertibles (sans opérande) (suite)

LIST de S5 (SIMATIC)	LIST de S5 (CEI)	LIST de S7 (SIMATIC)	LIST de S7 (CEI)
>=G	>=G	>=R	>=R
<G	<G	<R	<R
<=G	<=G	<=R	<=R
GFD	GFD	RND	RND
KEW	CFW	INVI	INVI
KZD	CSD	NEGD	NEGD
KZW	CSW	NEGI	NEGI
O	O	O	O
O(O(O(O(
OW	OW	OW	OW
STP	STP	CALL SFC46	CALL SFC46
STS	STS	CALL SFC46	CALL SFC46
STW	STW	CALL SFC46	CALL SFC46
TAK	TAK	TAK	TAK
U(A(U(A(
UW	AW	UW	AW
XOW	XOW	XOW	XOW
))))
***	***	NETWORK	NETWORK

Opérations convertibles, avec opérandes

Le tableau A-4 présente toutes les opérations de S5 (avec opérandes) en LIST qui sont converties automatiquement en LIST de S7 par le convertisseur.

Tabelle A-4 Opérations convertibles (avec opérandes)

LIST de S5 (SIMATIC)	LIST de S5 (CEI)	LIST de S7 (SIMATIC)	LIST de S7 (CEI)
A	C	AUF	OPN
ADD BF ADD DH ADD KF	ADD BF ADD DH ADD KF	+	+
AX	CX	AUF	OPN
B	DO	Séquence d'instructions pour adressage indirect	Séquence d'instructions pour adressage indirect
BA	BA	(vide)	(vide)
BAB	DOC	SPB	JC
D	D	DEC	DEC
E	G	CALL SFC22	CALL SFC22
EX	GX	CALL SFC22	CALL SFC22
FR	FR	FR	FR

Table A-4 Opérations convertibles (avec opérandes), suite

LIST de S5 (SIMATIC)	LIST de S5 (CEI)	LIST de S7 (SIMATIC)	LIST de S7 (CEI)
I	I	INC	INC
L	L	L	L
LC	LD	LC	LC
NOP	NOP	NOP	NOP
O	O	O	O
ON	ON	ON	ON
P	TB	SET; U	SET; A
PN	TBN	SET; UN	SET; AN
R	R	R	R
RB	RB	R	R
RD	RD	R	R
RLD	RLD	RLD	RLD
RLW	RLW	RLW	RLW
RRD	RRD	RRD	RRD
RRW	RRW	RRW	RRW
RU	RU	SET; R	SET; R
S	S	S	S
SA	SF	SA	SF
SAR	SFD	SA Temporisation ZR Compteur	SF Timer CD Counter
SE	SD	SE	SD
SI	SP	SI	SP
SLD	SLD	SLD	SLD
SLW	SLW	SLW	SLW
SPA	JU	SPA	JU
SPB	JC	SPB	JC
SPM	JM	SPM	JM
SPN	JN	SPN	JCN
SPO	JO	SPO	JO
SPP	JP	SPP	JP
SPR	JUR	SPA	JU
SPS	JOS	SPS	JOS
SPZ	JZ	SPZ	JZ
SRD	SRD	SRD	SRD
SRW	SRW	SRW	SRW
SS	SS	SS	SS
SSV	SSU	SS Temporisation ZV Compteur	SS Timer CU Counter

Table A-4 Opérations convertibles (avec opérandes), suite

LIST de S5 (SIMATIC)	LIST de S5 (CEI)	LIST de S7 (SIMATIC)	LIST de S7 (CEI)
SU	SU	SET; S	SET; S
SV	SE	SV	SE
SVD	SSD	SSD	SSD
SVW	SSW	SSI	SSI
SVZ	SEC	SV Temporisation S Compteur	SE Timer S Counter
T	T	T	T
TNB	TNB	CALL SFC20	CALL SFC20
TNW	TNW	CALL SFC20	CALL SFC20
U	A	U	A
UN	AN	UN	AN
ZR	CD	ZR	CD
ZV	CU	ZV	CU
=	=	=	=

Opérations non convertibles

Le tableau suivant présente les opérations LIST de S5 qui ne sont pas converties automatiquement par le convertisseur.

Tableau A-5 Opérations non convertibles

LIST de S5 (SIMATIC)	LIST de S5 (CEI)
AAS	IAI
AAF	RAI
ABR	ABR
ACR	ACR
AFF	RAE
AFS	IAE
ASM	ASM
BAF	BAF
BAS	BAS
BI (convertible seulement pour type de paramètre D ou constante)	DI (convertible seulement pour type de paramètre D ou constante)
BLD	BLD
LB	LB
LD	LD
LD=<paramètre formel> (convertible seulement pour type de paramètre D ou constante)	LD=<paramètre formel> (convertible seulement pour type de paramètre D ou constante)
LDI	LDI
LIM	LIM

Tableau A-5 Opérations non convertibles (suite)

LIST de S5 (SIMATIC)	LIST de S5 (CEI)
LIR	LIR
LRB	LRB
LRD	LRD
LRW	LRW
LW	LW
LW=<paramètre formel> (convertible seulement pour type de paramètre D ou constante)	LW=<paramètre formel> (convertible seulement pour type de paramètre D ou constante)
MA1	MA1
MAB	MAB
MAS	MAS
MBA	MBA
MBR	MBR
MBS	MBS
MSA	MSA
MSB	MSB
SEF	SEE
SES	SED
SIM	SIM
TB	TB
TDI	TDI
TIR	TIR
TSC	TSC
TSG	TSG
TRB	TRB
TRD	TRD
TRW	TRW
TW	TW
TXB	TXB
TXW	TXW
UBE	UBE

Bibliographie

- /21/ Guide : *Systèmes d'automatisation SIMATIC S7/M7*,
Décentraliser avec PROFIBUS-DB et AS-I
- /30/ Petit manuel illustré : *Faites connaissance avec le S7-300...*
- /70/ Manuel : *Automate programmable S7-300*,
Installation et configuration - Caractéristiques des CPU
- /71/ Manuel de référence : *Systèmes d'automatisation S7-300, M7-300*
Caractéristiques des modules
- /72/ Liste des opérations : *Automate programmable S7-300*
- /100/ Manuel de mise en œuvre : *Systèmes d'automatisation S7-400, M7-400*,
Installation et configuration
- /101/ Manuel de référence : *Systèmes d'automatisation S7-400, M7-400*
Caractéristiques des modules
- /102/ Liste des opérations : *Automate programmable S7-400*
- /231/ Guide de l'utilisateur : *Logiciel de base pour SIMATIC S7 et M7*,
STEP 7
- /232/ Manuel : *Langage LIST pour SIMATIC S7-300/400*,
Programmation de blocs
- /233/ Manuel : *Langage CONT pour SIMATIC S7-300/400*,
Programmation de blocs
- /234/ Manuel de programmation : *Logiciel système pour SIMATIC S7-300/400*
Conception de programmes
- /235/ Manuel de référence : *Logiciel système pour SIMATIC S7-300/400*
Fonctions standard et fonctions système
- /236/ Manuel : *Langage LOG pour SIMATIC S7-300/400*,
Programmation de blocs
- /250/ Manuel : *SCL pour SIMATIC S7-300/400*,
Programmation de blocs
- /251/ Manuel : *GRAPH pour SIMATIC S7-300/400*,
Programmation de commandes séquentielles
- /252/ Manuel : *HiGraph pour SIMATIC S7-300/400*,
Programmation de graphes d'état
- /253/ Manuel : *C pour SIMATIC S7-300/400*,
Programmation en C
- /254/ Manuel : *CFC, Continuous Function Chart*,
volume 1
- /249/ Manuel : *CFC, Continuous Function Chart*,
volume 2 : S7 / M7

- /270/ Manuel : *Diagnostic processus en CONT, LOG et LIST.*
- /271/ Manuel : *NETPRO*,
Configuring Networks (disponible en anglais seulement)
- /280/ Manuel de programmation : *Logiciel système pour SIMATIC M7-300/400*,
Conception de programmes
- /281/ Manuel de référence : *Logiciel système pour SIMATIC M7-300/400*,
Fonctions standard et fonctions système
- /282/ Guide de l'utilisateur : *Logiciel système pour SIMATIC M7-300/400*,
Installation et mise en œuvre
- /290/ Guide de l'utilisateur : *ProC/C++ pour SIMATIC M7-300/400*,
Programmation en C
- /291/ Guide de l'utilisateur : *ProC/C++ pour SIMATIC M7-300/400*,
Débogueur pour programmes en C
- /500/ Manuel : *SIMATIC NET*,
NCM S7 pour Industrial Ethernet
- /501/ Manuel : *SIMATIC NET*,
NCM S7 pour PROFIBUS
- /800/ *DOCPRO*,
Documentation normalisée d'un projet (uniquement sur CD-ROM)
- /801/ *Téléservice pour automates programmables de type S7, C7 et M7*,
(uniquement sur CD-ROM)
- /802/ *PLC Simulation pour SIMATIC S7-300 et S7-400*
(uniquement sur CD-ROM)
- /803/ Manuel de référence : *Logiciel standard pour S7-300/400*,
STEP 7 Fonctions standard, 2^{ème} partie (uniquement sur CD-ROM)

Glossaire

A

Adresse	Identifie un opérande ou une zone d'opérande particulier(e). Exemple : entrée E 12.1 ; mot de memento MW 25 ; bloc de données DB3.
Aide en ligne	Vous avez la possibilité, tout en travaillant avec STEP 7, d'afficher en ligne une aide contextuelle.
Appel de bloc	Aiguillage du traitement du programme dans le bloc appelé.

B

Bloc	Les blocs constituent des parties délimitées d'un programme utilisateur par leur fonction, leur structure et leur objet. Il existe dans STEP 7 : <ul style="list-style-type: none">• des blocs de code (FB, FC, OB, SFB, SFC),• des blocs de données (DB, SDB)• et des types de données utilisateur (UDT).
Bloc de code	Dans SIMATIC S7, un bloc de code est un bloc contenant une partie du programme utilisateur STEP 7, contrairement aux blocs de données qui ne contiennent que des données. Il existe les blocs de code suivants : blocs d'organisation (OB), blocs fonctionnels (FB), fonctions (FC), blocs fonctionnels système (SFB), fonctions système (SFC).
Bloc de données (DB)	Les blocs de données sont des zones de données dans le programme utilisateur qui contiennent des données utilisateur. Il existe des blocs de données globaux auxquels tous les blocs de code peuvent accéder et des blocs de données d'instance associés à un appel de FB précis. Contrairement aux autres blocs, les blocs de données ne contiennent pas d'instruction.
Bloc de données d'instance	Un bloc de données d'instance permet de sauvegarder les paramètres formels et les données statiques de blocs fonctionnels. Il peut être affecté à un appel de bloc fonctionnel ou à une hiérarchie d'appel de blocs fonctionnels.

Bloc d'organisation (OB) Les blocs d'organisation constituent l'interface entre le système d'exploitation de la CPU et le programme utilisateur. L'ordre de traitement du programme utilisateur est défini dans les blocs d'organisation.

Bloc fonctionnel (FB) Selon la norme CEI 1131-3, un bloc fonctionnel (FB) est un bloc de code avec données statiques. Un bloc fonctionnel permet la transmission de paramètres dans le programme utilisateur. Aussi, les blocs fonctionnels se prêtent-ils à la programmation de fonctions complexes se répétant souvent, comme par exemple les régulations et le choix du mode de fonctionnement. Comme un FB dispose d'une mémoire, le bloc de données d'instance, il est possible d'accéder à ses paramètres – par exemple, à ses sorties – à tout moment et à tout endroit dans le programme utilisateur.

C

Compilateur Programme de compilation avec lequel un programme écrit dans un langage évolué peut être compilé dans le code machine avec lequel la CPU travaille.

Configuration La configuration consiste à choisir et à assembler les différentes composantes d'un automate programmable ou à installer le logiciel nécessaire et à les adapter à l'usage prévu (par exemple, par paramétrage des modules).

D

Données globales Les données globales sont des données accessibles à chaque bloc de code (FB, FC, OB), à savoir des mementos (M), des entrées (E), des sorties (A), des temporisations, des compteurs et des éléments de blocs de données. Il est possible d'y accéder par adressage absolu ou symbolique.

Données locales Les données locales sont les données associées à un bloc de code qui sont déclarées dans la section de déclaration ou dans la déclaration des variables de ce bloc. Elles comprennent – selon le bloc – les paramètres formels, les données statiques, les données temporaires.

Données statiques Les données statiques sont des données locales d'un bloc fonctionnel, sauvegardées dans le bloc de données d'instance et donc conservées jusqu'au traitement suivant du bloc fonctionnel.

Données temporaires Les données temporaires sont des données locales d'un bloc qui sont rangées dans la pile L pendant l'exécution de ce bloc et ne sont plus disponibles une fois l'exécution achevée.

F**Fonction (FC)**

Selon la norme CEI 1131-3, une fonction (FC) est un bloc de code sans mémoire. Elle permet la transmission de paramètres dans le programme utilisateur. Aussi, les fonctions se prêtent-elles à la programmation de fonctions complexes se répétant souvent, comme les calculs. Comme il n'y a pas de mémoire, il faut traiter les valeurs calculées par la fonction directement après l'appel de cette dernière.

I**Instance**

Toutes les fois qu'un bloc fonctionnel est appelé, une instance de ce bloc est créée et un bloc de données d'instance lui est affecté.

Instruction

Unité élémentaire d'un programme utilisateur écrit dans un langage littéral. Elle représente une instruction de travail pour le processeur.

L**Langage de programmation**

Un langage de programmation permet de créer des programmes utilisateur en mettant à votre disposition des instructions graphiques ou littérales. Vous saisissez ces instructions, qui sont ensuite compilées en un programme utilisateur exécutable, à l'aide d'un éditeur.

Liste d'instructions (LIST)

La liste d'instructions est un langage de programmation littéral proche de la machine.

M**Macro-instruction**

Séquence d'instructions regroupées sous un nom symbolique permettant ainsi de minimiser les coûts d'exécution.

Mnémonique

Un mnémonique est un nom que l'utilisateur définit en respectant les règles de syntaxe imposées. Ce nom peut être utilisé pour la programmation et le contrôle-commande une fois son affectation déterminée (par exemple, variable, type de données, repère de saut, bloc).

Exemple : opérande : E 5.0, type de données : BOOL, mnémonique : Bouton_Arret_urg.

O

Opérande	Un opérande est la partie d'une instruction S7 indiquant l'objet que le processeur doit traiter. On peut y accéder par une adresse absolue ou symbolique.
Opération	Une opération est la partie d'une instruction S7 indiquant ce que le processeur doit faire.

P

Paramétrage	On entend par paramétrage la définition du comportement d'un module.
Paramètres de bloc	Marques de réservation dans des blocs à utilisations multiples. Ils reçoivent des valeurs actuelles quand le bloc concerné est appelé.
Paramètres effectifs	Les paramètres effectifs remplacent les paramètres formels lors de l'appel d'un bloc fonctionnel (FB) ou d'une fonction (FC). Exemple : le paramètre formel "Demarr" est remplacé par le paramètre effectif "E 3.6".
Paramètres formels	<p>Un paramètre formel "réserve la place" pour le paramètre effectif dans les blocs de code paramétrables. Pour les FB et les FC, c'est l'utilisateur qui déclare les paramètres formels ; ils existent déjà pour les SFB et les SFC.</p> <p>A l'appel du bloc, un paramètre effectif est affecté au paramètre formel afin que le bloc appelé utilise les valeurs en cours. Les paramètres formels font partie des données locales du bloc et se répartissent en paramètres d'entrée, de sortie et d'entrée/sortie.</p>
Périphérie décentralisée	La périphérie décentralisée est constituée de modules éloignés dans l'espace de l'appareil de base. Sa caractéristique principale est sa technique de montage qui vise à réduire au minimum les besoins de câblage (donc les coûts) en installant les modules de signaux à proximité du processus.
Pointeur	Variable qui ne contient pas de valeur particulière, mais l'adresse d'une autre variable. Pour les opérations de pointeur, il faut que les types indiqués à droite et à gauche de l'opérateur soient identiques.
Programme S7	Le programme S7 est un classeur de blocs, sources et diagrammes pour modules S7 programmables. Il contient également la table des mnémoniques.
Projet	Un projet est un classeur pour tous les objets d'une solution d'automatisation, indépendamment du nombre de stations, de modules et de leur mise en réseau.

R

Rémanence On qualifie de rémanentes des données qui conservent leur valeur pendant une coupure de la tension d'alimentation. La rémanence est possible grâce à deux types de sauvegardes : par pile et par mémoire tampon.

S

Section de déclaration Section qui sert à déclarer les données locales d'un bloc de code lorsque les programmes sont créés à l'aide d'un éditeur de texte.

Symbolique (mnémoniques) On distingue symbolique globale et symbolique locale de bloc. Les mnémoniques spécifiés globalement sont connus dans toutes les parties du programme, le mnémonique attribué doit être sans équivoque pour l'ensemble du programme utilisateur. Les mnémoniques locaux de bloc ne sont connus que dans le bloc dans lequel ils ont été spécifiés.

T

Table des mnémoniques Cette table contient l'affectation de mnémoniques (ou noms symboliques) à des adresses pour les données globales et les blocs.

Exemple : Arret_urg (mnémonique), E 1.7 (adresse)
 Regulateur (mnémonique), SFB24 (bloc)

Type de données On définit, à l'aide du type de données, comment la valeur d'une variable ou d'une constante doit être utilisée dans le programme utilisateur. Il existe, dans SIMATIC S7, deux catégories de types de données se basant sur la norme CEI 1131-3 : les types de données simples et les types de données complexes.

Type de données complexe Les types de données complexes sont créés par l'utilisateur avec la déclaration de type de données. Ils n'ont pas de nom en propre et ne sont donc pas réutilisables. On distingue les tableaux et les structures. Les types de données STRING et DATE_AND_TIME sont également des types de données complexes.

Type de données simple Les types de données simples sont des types de données prédéfinis selon CEI 1131-3. Exemples : le type de données BOOL définit une variable binaire (bit) et le type de données INT une variable entière de 16 bits.

V

Variable

Une variable est une donnée à contenu variable pouvant être utilisée dans le programme utilisateur STEP 7. Une variable est constituée d'une adresse et d'un type de données ; elle peut être identifiée par un mnémonique.

Index

A

Actuator Sensor Interface, 2-10
Adressage
 absolu, 3-38
 indirect, 3-42
 indirect par mémoire, 3-43
 opérandes, 3-40
 symbolique, 3-38
Adressage indirect, conversion, 7-4
Adressage par page, 3-37
Adresse absolue, 4-2
Adresses, modification, 7-2
Alarme, 3-20, 3-22
Alarme cyclique, 3-20
Alarme de diagnostic, 2-15, 9-2
Alarme de processus, 2-15, 3-20
Alarme horaire, 3-20
Alarme multiprocesseur, 3-20
Alarme temporisée, 3-20
Arithmétique en virgule flottante, 3-28
Arithmétique sur nombres entiers, 3-36
Arithmétique virgule flottante, 3-36
AS511, 2-3
Autorisation, 3-2
Avertissements, 6-10

B

Bibliothèque standard, 3-15
Bloc d'organisation, 3-17
Bloc de commentaire, 3-17
Bloc de dialogue, 2-20
Bloc de données, 3-17
Bloc de données système, 3-17, 3-19
Bloc de programme, 3-17
Bloc fonctionnel, 3-17, 3-18

Bloc fonctionnel système, 3-17, 3-19
Bloc séquentiel, 3-17
Blocs, de la CPU, 2-6
Blocs , S5, de dialogue, 2-20
Blocs d'organisation, 3-20, 5-7
Blocs fonctionnels standard, 7-6
Blocs S7, créer, 3-15
Boîtier d'adaptation, 2-13

C

Came électronique, 2-13
Catalogue des modules, 3-10
CD-ROM, 2-1
Changement de front, 2-15
Classeur des blocs, objet de STEP 7, 3-6
Cohérence, vérification, 8-1
Commandes d'interruption, 3-37
Communication
 commandée par événement, 2-19
 FMS, 2-18
 fonctions de, 2-18
Commutateur DIL, 2-5
Compilation, 8-1
Compteurs, de la CPU, 2-6
Configuration
 liaisons de communication, 3-11
 matérielle, 3-9
 outil de, 2-21
Contrôle-commande, 2-21
Conversion
 adressage indirect, 7-4
 conditions préalables, 4-2
COROS, 2-3
Couplage point-à-point, interface dans le programme utilisateur, 2-20

CP, 2-10

CPU

blocs, 2-6

compteurs, 2-6

DB, 2-6

données locales, 2-6

données rémanentes, 2-6

entrées analogiques, 2-6

entrées TOR, 2-6

FB, 2-6

FC, 2-6

mémentos, 2-6

mémoire de chargement, 2-6, 2-7

Mémoire de travail, 2-6

mémoire image, 2-6

OB, 2-6

S7-300, 2-6

S7-400, 2-7

SFB, 2-6

SFC, 2-6

sorties analogiques, 2-6

sorties TOR, 2-6

temporisations, 2-6

D

DB1, 3-26

DB1 / DX0, 4-3, 5-4

Débordement de zone, 3-22

Défaillance de la pile, 3-22

Démarrage, 3-20

Données locales, 3-33

de la CPU, 2-6

Données, rémanentes, de la CPU, 2-6

DP

modules pouvant être utilisés comme esclaves

DP, 2-17

modules pouvant être utilisés comme maître DP,
2-17

DX0, 3-26

E

Entrées

analogiques, 2-6

TOR, 2-6

Espace mémoire, 4-2

ET 200, 2-17

Etat des modules, 5-3

Ethernet, 2-10

Exemple

données locales temporaires, 9-5

informations de déclenchement, 9-9

traitement des valeurs analogiques, 9-2

transfert par blocs, 9-12

Extension, châssis d'extension S5, 2-9

F

FDL (SDA), 2-18

Fichier de projet, 3-4

FM, 2-13

FMS

esclaves, 2-17

maître, 2-17

Fonction, 3-18

Fonction standard, 3-28

Fonction système, 3-17, 3-19

Fonctions analogiques, 3-29

Fonctions de base, 3-29

Fonctions de signalisation, 3-28

Fonctions de substitution, (B MW, B DW), 4-2

Fonctions intégrées, 3-28

Fonctions mathématiques, 3-29, 3-37

Fonctions spéciales, 3-22

Format de fichier, 3-39

Format pointeur, 3-42

Formats de constante, 3-31

G

Gestionnaire de projets SIMATIC, 3-3

Groupe de puissance, 2-2

H

HMI (Human Machine Interface), 2-3, 2-21

Homogénéité, des données, 1-1

Horodateur interne, réglage et lecture, 3-22

I

Importation

source ASCII, 3-16

table des mnémoniques, 3-39

Industrial Ethernet, 2-10, 2-18

interface dans le programme utilisateur, 2-20

modules, 2-11

Informations de déclenchement, 3-34, 9-9

Installation, STEP 7, 3-2
 Intégration totale, 1-1
 Interface, MPI, 2-3
 Interface AS-i, 2-10
 Interface PG, AS511, 2-3
 Interruption, 3-20
 ISO, transport, 2-18
 ISO-on-TCP, 2-18

L

Liaisons, à une station S5, configurer, 3-11
 Licence d'utilisation, 3-2
 LIR, 4-2
 Liste d'assignation, 3-38, 6-1, 6-4
 Liste des références croisées, 6-1
 Logiciel, production, 3-13
 insertion des composants, 3-15
 vue d'ensemble des composants, 3-13

M

Macro-instructions, 5-5
 édition, 5-8
 pour OB, 5-7
 pour opérations, 5-6
 Matériel, objet de STEP 7, 3-5
 Mémento de couplage, 3-23
 Mémentos, de la CPU, 2-6
 Mémentos banalisés, 3-33, 9-6
 Mémoire de chargement
 CPU S7-300, 2-6
 CPU S7-400, 2-7
 Mémoire de travail, de la CPU, 2-6
 Mémoire image, de la CPU, 2-6
 Mémoire tampon de diagnostic, 2-15
 Messages d'erreur, 6-8
 Mini-automate, 2-2
 Mise en parallèle des blocs STEP 5/STEP 7, 3-17
 Mise en route, 3-20
 Mnémonique, local, 3-39
 Modification d'adresses, 7-2
 Module d'alimentation, 2-8
 Module de comptage, 2-13
 Module de dosage, 2-13
 Module de positionnement, 2-13
 Module de régulation, 2-13
 Module programmable, 3-5
 Modules
 de simulation, 2-16
 gamme des, 2-4
 IP, 2-13
 prétraitement de signaux, 2-13
 WF, 2-13

Modules de communication, 2-10
 Modules de fonction, 2-13
 MPI, 2-3, 2-18
 Multi Point Interface, 2-3

O

OB. *Siehe* Bloc d'organisation
 OB spéciaux, 3-17
 OB1, exemple, 9-15
 Opérandes
 convertibles, A-1
 non convertibles, A-2
 Opérations, A-3, A-4
 convertibles, A-3, A-4
 non convertibles, A-6
 vue d'ensemble, 3-35
 Opérations d'arrêt, 3-37
 Opérations de chargement, 3-35
 Opérations de comparaison, 3-36
 Opérations de comptage, 3-35
 Opérations de conversion, 3-36
 Opérations de décalage, 3-36
 Opérations de saut, 3-37
 Opérations de transfert, 3-35
 Opérations de validation de sorties, 3-37
 Opérations nulles, NOP, 3-37
 Opérations sur accumulateurs, 3-35
 Opérations sur bits, 3-35
 Opérations sur blocs, 3-37
 Opérations sur blocs de données, 3-36
 Opérations sur mots, 3-36
 Opérations sur registres, 3-35
 Outil, de conversion du matériel, 2-1

P

Paramétrage des modules, comparaison S5/S7, 2-5
 Paramètres système S5, 3-26
 Périphérie décentralisée, 2-17
 PG, interface, 2-10
 Point-à-point
 couplage, 2-10
 modules pour couplage, 2-12
 Pointeur ANY, 9-13
 Pro Tool, 2-22
 PROFIBUS, 2-3, 2-10, 2-18
 interface dans le programme utilisateur, 2-20
 modules, 2-11
 Projet, 3-4
 Projet S7, création, 4-3
 Projet STEP 7, sauvegarde, 3-8
 Projets, création de , 3-7
 Pupitres opérateur (OP), 2-21

R

Réassignation, 5-4, 7-2
 Redémarrage, 3-20
 Registre à décalage, 3-23
 Registre BR, 7-5
 Registre d'adresse, 3-44
 Registre DB, 3-40, 3-41
 Rémanence, 2-7
 Réseau, objet de STEP 7, 3-5
 RET_VAL, 9-3

S

Sauvegarde, 2-7
 SIMATIC Manager, 3-3
 fenêtre, 3-13
 SIMATIC S7, vue d'ensemble, 2-2
 SINEC H1, 2-11
 SINEC L1, 2-11, 3-26
 SINEC L2, 2-11, 3-26
 SINEC S1, 2-11
 Sorties
 analogiques, 2-6
 TOR, 2-6
 Source, objet de STEP 7, 3-6
 Source ASCII, 3-16
 Station, objet de STEP 7, 3-5
 STEP 5, projet, 3-4
 STEP 5, blocs, 3-17
 STEP 7
 archiver un projet, 3-8
 composants du projet, 3-5
 création du projet, 3-7
 démarrage de, 3-3
 installation, 3-2
 projet, 3-4
 Surveillance du temps de cycle, 3-23
 Systèmes d'automatisation, série, 2-2

T

Table de mnémoniques, 3-39

Table des liaisons, 3-11
 objet de STEP 7, 3-6
 Table des mnémoniques
 créer, 3-15
 exemple, 9-15
 objet de STEP 7, 3-6
 Temporisations, 3-35
 de la CPU, 2-6
 TIR, 4-2
 Total de contrôle, 3-23
 Traitement d'erreur, 3-21
 Traitement des valeurs analogiques, exemple, 9-2
 Traitement en arrière-plan, 3-20
 Transfert de blocs, 3-37
 Transfert par blocs, 7-5
 exemple, 9-12
 Types de blocs, dans S5 et S7, 3-25

U

Unité centrale
 S7-300, 2-6
 S7-400, 2-7

V

Valeur de retour de la fonction, 9-3
 Valeur en retour, d'une fonction, 3-22
 Valeurs analogiques, traitement, 9-2
 Visualisation, 2-22

W

WinCC, 2-22

Z

Zones d'opérande, vue d'ensemble, 3-32

Siemens AG
A&D AS E81

Oestliche Rheinbrueckenstr. 50
D-76181 Karlsruhe
République Fédérale d'Allemagne

Expéditeur :

Vos . Nom : _ _ _ _ _
Fonction : _ _ _ _ _
Entreprise : _ _ _ _ _
Rue : _ _ _ _ _
Code postal : _ _ _ _ _
Ville : _ _ _ _ _
Pays : _ _ _ _ _
Téléphone : _ _ _ _ _

Indiquez votre secteur industriel :

- | | |
|---|---|
| <input type="checkbox"/> Industrie automobile | <input type="checkbox"/> Industrie pharmaceutique |
| <input type="checkbox"/> Industrie chimique | <input type="checkbox"/> Traitement des matières plastiques |
| <input type="checkbox"/> Industrie électrique | <input type="checkbox"/> Industrie du papier |
| <input type="checkbox"/> Industrie alimentaire | <input type="checkbox"/> Industrie textile |
| <input type="checkbox"/> Contrôle/commande | <input type="checkbox"/> Transports |
| <input type="checkbox"/> Construction mécanique | <input type="checkbox"/> Autres _ _ _ _ _ |
| <input type="checkbox"/> Pétrochimie | |



Remarques / suggestions

Vos remarques et suggestions nous permettent d'améliorer la qualité générale de notre documentation. C'est pourquoi nous vous serions reconnaissants de compléter et de renvoyer ces formulaires à Siemens.

Répondez aux questions suivantes en attribuant une note comprise entre 1 pour très bien et 5 pour très mauvais.

- | | | |
|----|--|--------------------------|
| 1. | Le contenu du manuel répond-il à votre attente ? | <input type="checkbox"/> |
| 2. | Les informations requises peuvent-elles facilement être trouvées ? | <input type="checkbox"/> |
| 3. | Le texte est-il compréhensible ? | <input type="checkbox"/> |
| 4. | Le niveau des détails techniques répond-il à votre attente ? | <input type="checkbox"/> |
| 5. | Quelle évaluation attribuez-vous aux figures et tableaux ? | <input type="checkbox"/> |

Vos remarques et suggestions:
