

SIMATIC

Programmer avec STEP 7 Lite V3.0

Manuel

Avant-propos, Sommaire	
Introduction sur le produit et installation	1
Principes de conception d'une programme	2
Démarrage et utilisation du programme	3
Création et édition du projet	4
Configuration matérielle	5
Programmation de blocs	6
Etablissement d'une liaison en ligne et choix de la CPU	7
Importer, exporter, enregistrer sous	8
Chargement dans la CPU et la PG	9
Test	10
Diagnostic	11
Imprimer une documentation du projet	12
Astuces et conseils	13
Annexe	A
Index	

Consignes de 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.



Précaution

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.

Avertissement

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

Attention

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'appareil ne doivent être effectuées que conformément au manuel. Seules des **personnes qualifiées** sont autorisées à effectuer des interventions sur l'appareil. Il s'agit de personnes qui ont l'autorisation de mettre en service, de mettre à la terre et de repérer des appareils, des systèmes et circuits électriques conformément aux règles de sécurité en vigueur.

Utilisation conforme

Tenez compte des points suivants :



Précaution

L'appareil, le système ou le composant 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 l'appareil sont les conditions indispensables pour garantir son fonctionnement correct et sûr.

Marque de fabrique

SIMATIC®, SIMATIC NET® et SIMATIC HMI® sont des marques déposées par SIEMENS AG.

Les autres désignations 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 2004 Tous droits réservés

Toute communication et 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.

Exclusion de responsabilité

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.

Avant-propos

Objet du manuel

Ce manuel vous donne une vue d'ensemble de la programmation avec STEP 7 Lite. Il vous aidera à installer le logiciel et à le mettre en route. Il explique la démarche pour créer un programme, la structure de programmes utilisateur et les différents éléments de langage.

Ce manuel s'adresse aux personnes responsables de la réalisation de tâches de commande avec STEP 7 Lite sur la base des automates programmables SIMATIC.

Nous vous conseillons de prendre connaissance des exemples figurant dans le manuel "Mise en route de STEP 7 Lite". Il vous permettra de vous familiariser de manière conviviale avec la programmation avec STEP 7 Lite.

Connaissances requises

La compréhension du manuel requiert des connaissances générales dans le domaine de la technique d'automatisation de processus.

Nous supposons donc pour la suite que l'emploi d'ordinateurs ou de machines analogues (telles que les consoles de programmation) ainsi que les environnements MS Windows 2000 Professional, MS Windows XP Home et MS Windows XP Professional vous sont familiers.

Domaine de validité du manuel

Le présent manuel est valable pour le logiciel STEP 7 Lite V3.0.

Place du manuel dans la documentation

Le présent manuel est inclus à la livraison du logiciel STEP 7 Lite.

Le tableau suivant donne une vue d'ensemble de la documentation de STEP 7 Lite :

Manuels	Contenu	Numéro de référence
Programmer avec STEP 7 Lite	Notions de base pour la réalisation de tâches de commande avec STEP 7 Lite.	Intégré au logiciel STEP 7 Lite
Mise en route de STEP 7 Lite	Décrit les opérations les plus importantes à l'aide d'exercices pratiques.	Intégré au logiciel STEP 7 Lite

Aides en ligne	Contenu	Numéro de référence
Aide de STEP 7 Lite	Notions de base pour la réalisation de tâches de commande avec STEP 7 Lite.	Intégré au logiciel STEP 7 Lite
Aide de référence de CONT, LOG, LIST Aides de référence pour les bibliothèques de blocs	Aide de référence contextuelle	Intégré au logiciel STEP 7 Lite
Aide succincte et aide directe	Donne des informations sur le contexte actuel, par exemple pour les commandes de menu, les éléments de l'interface utilisateur et les boîtes de dialogue	Intégré au logiciel STEP 7 Lite

Assistance supplémentaire

Si des questions sont restées sans réponse dans ce manuel, veuillez vous adresser à votre interlocuteur Siemens dans la filiale ou l'agence de votre région.

<http://www.ad.siemens.de/partner>

Vous trouvez un fil rouge pour la recherche de documentations techniques sur les produits et systèmes SIMATIC à l'adresse suivante sur Internet :

<http://www.siemens.com/simatic-tech-doku-portal>

Centre de formation

Nous vous proposons des cours de formation pour vous faciliter l'apprentissage des automates programmables SIMATIC S7. Veuillez vous adresser à votre centre de formation régional ou au centre principal à D 90327 Nürnberg.

Téléphone : +49 (911) 895-3200.

<http://www.sitrain.com>

Automation and Drives, Service & Support

Accessible dans le monde entier à toute heure :



Worldwide (Nuernberg) Technical Support Heure locale : 0h à 24h / 365 jours Tél. : +49 (180) 5050-222 Fax: +49 (180) 5050-223 mailto:adsupport@siemens.com GMT: +1:00		
Europe / Africa (Nuernberg) Authorization Heure locale : lu-ve. 8h à 17h Tél. : +49 (180) 5050-222 Fax: +49 (180) 5050-223 mailto:adsupport@siemens.com GMT: +1:00	United States (Johnson City) Technical Support and Authorization Heure locale : lu-ve 8h à 17h Tél. : +1 (423) 262 2522 Fax: +1 (423) 262 2289 mailto:simatic.hotline@sea.siemens.com GMT: -5:00	Asia / Australia (Beijing) Technical Support and Authorization Heure locale : lu-ve 8h à 17h Tél. : +86 10 64 75 75 75 Fax: +86 10 64 74 74 74 Mailto:adsupport.asia@siemens.com GMT: +8:00
Les langues parlées au Technical Support et sur la Hotline des autorisations sont généralement l'Allemand et l'Anglais.		

Service & Support sur Internet

En plus de la documentation offerte, vous trouvez la totalité de notre savoir-faire en ligne sur Internet à l'adresse suivante :

<http://www.siemens.com/automation/service&support>

Vous y trouvez :

- le bulletin d'informations qui vous fournit constamment les dernières informations sur le produit,
- les documents dont vous avez besoin à l'aide de la fonction de recherche du Service & Support,
- le forum où utilisateurs et spécialistes peuvent échanger informations,
- votre interlocuteur Automation & Drives sur place,
- des informations sur le service après-vente, les réparations, les pièces de rechange à la rubrique "Service".

Sommaire

1	Introduction sur le produit et installation	1-1
1.1	Présentation de STEP 7 Lite	1-1
1.2	Fenêtre du projet et vues dans STEP 7 Lite	1-5
1.3	Aide de STEP 7 Lite	1-9
1.4	Installation	1-10
1.4.1	Automation License Manager	1-10
1.4.1.1	Licence d'utilisation avec Automation License Manager	1-10
1.4.1.2	Installation de Automation License Manager	1-12
1.4.1.3	Règles d'utilisation des License Keys	1-14
1.4.2	Installation de STEP 7 Lite	1-15
1.4.2.1	Marche à suivre pour installer STEP 7 Lite	1-16
1.4.2.2	Paramétrage de l'interface PG/PC	1-18
1.4.3	Désinstallation de STEP 7 Lite	1-19
2	Principes de conception d'une programme.....	2-1
2.1	Programmes dans une CPU	2-1
2.2	Blocs dans le programme utilisateur	2-2
2.2.1	Blocs d'organisation et structure du programme.....	2-3
2.2.2	Hiérarchie d'appel dans le programme utilisateur	2-8
2.2.3	Traitement de programme cyclique et paramètres de CPU.....	2-10
2.2.3.1	Bloc d'organisation pour l'exécution cyclique du programme (OB1)	2-10
2.2.3.2	Charge due à la communication	2-13
2.2.4	Traitement de programme déclenché par alarme	2-14
2.2.4.1	Blocs d'organisation pour le traitement de programme déclenché par alarme.....	2-14
2.2.4.2	Blocs d'organisation pour l'alarme horaire (OB10 à OB17)	2-15
2.2.4.3	Blocs d'organisation pour l'alarme temporisée (OB20 à OB23).....	2-18
2.2.4.4	Blocs d'organisation pour l'alarme cyclique (OB30 à OB38)	2-19
2.2.4.5	Blocs d'organisation pour l'alarme de processus (OB40 à OB47)	2-20
2.2.4.6	Blocs d'organisation pour la mise en route (OB100/OB102)	2-21
2.2.4.7	Bloc d'organisation pour l'exécution du programme en arrière-plan (OB90)	2-23
2.2.4.8	Blocs d'organisation pour le traitement d'erreurs (OB80 à OB87 / OB121 à OB122).....	2-24
2.2.5	Catégorie de blocs pour le traitement de programme structuré.....	2-26
2.2.5.1	Fonctions (FC).....	2-26
2.2.5.2	Blocs fonctionnels (FB)	2-26
2.2.5.3	Blocs de données d'instance.....	2-29
2.2.6	Blocs de données globaux (DB).....	2-31
2.2.6.1	Blocs fonctionnels système (SFB) et fonctions système (SFC).....	2-32

3	Démarrage et utilisation du programme	3-1
3.1	Démarrage de STEP 7 Lite	3-1
3.2	Appel des fonctions d'aide	3-2
3.3	Interface utilisateur et manipulation	3-3
3.3.1	Interface utilisateur	3-3
3.3.2	Icônes utilisées dans la fenêtre du projet.....	3-5
3.3.3	Eléments des fenêtres et des boîtes de dialogue	3-6
3.3.4	Mémoire de session	3-9
3.3.5	Modifier la disposition des fenêtres	3-9
3.3.6	Enregistrer et restaurer la disposition des fenêtres	3-10
3.3.7	Recherche ou remplacement de termes.....	3-11
3.3.8	Manipulation des objets.....	3-13
3.3.8.1	Renommer un objet.....	3-13
3.3.8.2	Déplacer un objet	3-13
3.3.8.3	Supprimer un objet	3-13
3.4	Utilisation du clavier	3-14
3.4.1	Combinaisons de touches pour les commandes de menu	3-14
3.4.2	Combinaisons de touches pour le déplacement du pointeur	3-16
3.4.3	Combinaisons de touches pour la sélection de texte.....	3-17
3.4.4	Combinaisons de touches pour accéder à l'aide en ligne.....	3-17
3.4.5	Combinaisons de touches pour la bascule entre les différents types de fenêtres	3-18
3.5	Commande d'une installation à distance grâce au Téléservice	3-19
4	Création et édition du projet.....	4-1
4.1	Qu'est-ce qu'un projet STEP 7 Lite ?	4-1
4.2	Création d'un projet	4-4
4.2.1	Création d'un projet	4-4
4.2.2	Insertion d'un programme	4-4
4.3	Edition d'un projet.....	4-6
4.3.1	Application et enregistrement des modifications.....	4-7
4.3.2	Détails sur l'édition	4-9
4.3.2.1	Copie d'un projet	4-9
4.3.2.2	Copie d'un élément de projet	4-9
4.3.2.3	Configuration du matériel (principe)	4-10
4.3.2.4	Création du logiciel dans le projet (principe)	4-10
4.4	Effacer et renommer un projet	4-10
5	Configuration matérielle	5-1
5.1	Principes de la configuration matérielle avec STEP 7 Lite	5-1
5.1.1	Introduction à la configuration matérielle	5-1
5.1.2	Manipulations de base pour la configuration matérielle.....	5-2
5.1.2.1	Marche à suivre pour configurer une station.....	5-3
5.1.2.2	La vue 'Configuration HW'	5-4
5.1.2.3	Table de configuration comme reproduction d'un profilé support ou châssis	5-5
5.1.2.4	Définir les propriétés de composants.....	5-6
5.1.2.5	Informations sur les règles d'enchâssement et autres règles.....	5-6
5.2	Configuration des modules.....	5-7
5.2.1	Règles pour disposer des modules (SIMATIC 300).....	5-7
5.2.1.1	Règles particulières pour le module fictif DM 370 Dummy	5-8
5.2.1.2	Règles particulières pour le module de simulation TOR SIM 374 IN/OUT 16.....	5-8

5.2.2	Règles pour disposer des modules ET 200S et ET 200X.....	5-9
5.2.2.1	Règles pour disposer des modules ET 200S.....	5-9
5.2.2.2	Règles pour disposer des modules ET 200X.....	5-9
5.2.3	Marche à suivre pour la configuration des modules	5-10
5.2.3.1	Configuration et paramétrage d'une station	5-10
5.2.3.2	Sélection d'un type de station	5-10
5.2.3.3	Disposer des modules dans le profilé support/châssis	5-11
5.2.3.4	Affichage de la version du système d'exploitation CPU dans la liste des modules.....	5-12
5.2.3.5	Disposer des systèmes intégrés compacts C7 (particularités)	5-12
5.2.3.6	Définir les propriétés de modules/interfaces	5-13
5.2.3.7	Attribution des adresses.....	5-14
5.2.3.8	Attribution des adresses d'entrée/sortie	5-14
5.2.3.9	Conseils pour éditer une configuration de station.....	5-15
5.2.4	Ce qu'il faut savoir sur démarreurs de moteur ET 200S (High Feature).....	5-16
5.2.4.1	Reconnaissance des états de l'installation à l'aide des valeurs du courant	5-16
5.2.4.2	Courant à l'état bloqué	5-16
5.2.4.3	Temps de blocage	5-16
5.2.4.4	Réaction en cas de détection du courant résiduel	5-17
5.2.4.5	Déséquilibre	5-17
5.2.4.6	Modèle de moteur thermique	5-17
5.2.4.7	Temps de récupération	5-17
5.2.4.8	Tableau : Actions possibles du démarreur de moteur	5-18
5.2.4.9	Affectation du démarreur de moteur dans la mémoire image.....	5-19
5.3	Enregistrement d'une configuration et vérification de cohérence	5-20
6	Programmation de blocs.....	6-1
6.1	Définition de mnémoniques.....	6-1
6.1.1	Adressage absolu et adressage symbolique	6-1
6.1.2	Mnémoniques globaux et mnémoniques locaux.....	6-2
6.1.3	Représentation des mnémoniques globaux et des mnémoniques locaux.....	6-3
6.1.4	Paramétrage du classement d'opérandes (absolu/symbolique)	6-4
6.1.5	Table des mnémoniques pour mnémoniques globaux	6-5
6.1.5.1	Structure et éléments de la table des mnémoniques.....	6-5
6.1.5.2	Opérandes et types de données autorisés dans la table des mnémoniques	6-7
6.1.5.3	Mnémoniques incomplets ou non univoques dans la table des mnémoniques	6-8
6.1.6	Possibilités de saisie de mnémoniques globaux.....	6-9
6.1.6.1	Remarques générales sur la saisie de mnémoniques	6-10
6.1.6.2	Saisie de mnémoniques globaux individuels dans des boîtes de dialogue	6-11
6.1.6.3	Saisie de plusieurs mnémoniques globaux dans la table des mnémoniques	6-12
6.1.6.4	Exportation et importation de tables de mnémoniques.....	6-13
6.1.7	Edition de la table des mnémoniques	6-13
6.1.7.1	Ouverture d'une table des mnémoniques	6-13
6.1.7.2	Définition de mnémoniques individuels.....	6-13
6.1.7.3	Insertion de lignes de mnémonique	6-14
6.1.7.4	Suppression de lignes de mnémonique	6-14
6.1.7.5	Filtrage de la table des mnémoniques	6-15
6.1.7.6	Opérandes libres	6-15
6.1.7.7	Mnémoniques manquants	6-16
6.1.7.8	Tri de la table des mnémoniques	6-16
6.1.7.9	Sélection de lignes de mnémonique	6-16
6.1.7.10	Copie de lignes de mnémonique dans le presse-papiers.....	6-16
6.1.7.11	Enregistrement d'une table des mnémoniques.....	6-16

6.1.8	Modifier les paramètres de fenêtre	6-17
6.1.8.1	Afficher ou masquer la barre d'outils.....	6-17
6.1.8.2	Afficher ou masquer la barre d'état	6-17
6.1.8.3	Positionner la barre d'outils	6-17
6.1.8.4	Taille de l'affichage dans une fenêtre de travail.....	6-17
6.2	Utilisation de blocs	6-18
6.2.1	Editeur de bloc	6-18
6.2.2	Choix du langage de programmation	6-19
6.2.2.1	Langages de programmation de l'éditeur de bloc	6-19
6.2.2.2	Langage de programmation CONT (schéma à contacts)	6-20
6.2.2.3	Langage de programmation LOG (logigramme)	6-21
6.2.2.4	Langage de programmation LIST (liste d'instructions).....	6-21
6.2.3	Création de blocs	6-22
6.2.3.1	Types de données utilisateur (UDT)	6-22
6.2.3.2	Propriétés de bloc	6-23
6.2.3.3	Protection du bloc.....	6-25
6.2.3.4	Propriétés de bloc autorisées pour chaque type de bloc	6-26
6.2.3.5	Affichage des longueurs de bloc	6-26
6.2.3.6	Comparaison de blocs.....	6-27
6.2.4	Utilisation de bibliothèques.....	6-29
6.2.4.1	Présentation des bibliothèques de blocs.....	6-29
6.3	Création des blocs de code.....	6-30
6.3.1	Principes de la création de blocs de code.....	6-30
6.3.1.1	Marche à suivre pour créer des blocs de code	6-30
6.3.1.2	Présélections pour l'éditeur de bloc CONT/LOG/LIST	6-31
6.3.1.3	Instructions tirées des bibliothèques d'opérations	6-31
6.3.1.4	Choix de l'affichage dans la fenêtre d'édition.....	6-32
6.3.1.5	Fractionnement de la fenêtre de travail.....	6-33
6.3.1.6	Régler la largeur des colonnes	6-33
6.3.2	Edition de la table de déclaration des variables.....	6-34
6.3.2.1	Utilisation de la déclaration des variables dans les blocs de code	6-34
6.3.2.2	Interaction entre la table de déclaration des variables et la section des instructions.....	6-35
6.3.2.3	Structure de la table de déclaration des variables	6-36
6.3.2.4	Remarques générales sur les tables de déclaration de variables	6-37
6.3.2.5	Utilisation de la table de déclaration des variables	6-38
6.3.2.6	Saisie de types de données simples dans la table de déclaration	6-38
6.3.2.7	Saisie d'éléments de données du type STRUCT	6-38
6.3.2.8	Copie de variables dans une table de déclaration	6-40
6.3.2.9	Suppression de variables dans une table de déclaration	6-40
6.3.3	Multi-instances dans la table de déclaration des variables.....	6-41
6.3.3.1	Utilisation de multi-instances.....	6-41
6.3.3.2	Règles de formation des multi-instances	6-42
6.3.3.3	Saisie de la multi-instance dans la table de déclaration des variables	6-42
6.3.3.4	Remarques générales sur la saisie d'instructions et de commentaires	6-43
6.3.4	Structure de la section des instructions	6-43
6.3.4.1	Marche à suivre pour saisir des instructions	6-44
6.3.4.2	Saisie de mnémoniques globaux dans un programme.....	6-45
6.3.4.3	Titres et commentaires de blocs et de réseaux	6-45
6.3.4.4	Saisie de commentaires de blocs et de réseaux	6-46
6.3.4.5	Fonction de recherche d'erreurs dans la section des instructions	6-47

6.3.5	Edition d'instructions CONT dans la section des instructions	6-47
6.3.5.1	Paramètres pour le langage de programmation CONT	6-47
6.3.5.2	Règles de saisie des instructions CONT	6-48
6.3.5.3	Branchements interdits en CONT	6-50
6.3.5.4	Saisie d'éléments CONT	6-51
6.3.5.5	Création de nouvelles branches dans les réseaux CONT	6-54
6.3.5.6	Création de branches dans les réseaux CONT	6-57
6.3.6	Edition d'instructions LOG dans la section des instructions	6-57
6.3.6.1	Paramètres pour le langage de programmation LOG	6-57
6.3.6.2	Règles de saisie des instructions LOG	6-58
6.3.6.3	Comment saisir des éléments LOG	6-60
6.3.7	Edition d'instructions LIST dans la section des instructions	6-64
6.3.7.1	Paramètres pour le langage de programmation LIST	6-64
6.3.7.2	Règles de saisie des instructions LIST	6-64
6.3.7.3	Saisie d'instructions LIST	6-65
6.3.8	Actualisation d'appels de bloc	6-67
6.4	Création des blocs de données	6-68
6.4.1	Principes de la création des blocs de données	6-68
6.4.2	Vue des déclarations d'un bloc de données	6-69
6.4.3	Vue des données d'un bloc de données	6-70
6.4.4	Saisie et enregistrement des blocs de données	6-71
6.4.4.1	Saisie de la structure de données de blocs de données globaux	6-71
6.4.4.2	Saisie / affichage de la structure de données de blocs de données associés à un FB (DB d'instance)	6-71
6.4.4.3	Saisie de la structure de types de données utilisateur (UDT)	6-73
6.4.4.4	Saisie / affichage de la structure de blocs de données associés à un UDT	6-73
6.4.4.5	Modification de valeurs dans la vue des données	6-74
6.4.4.6	Réinitialisation de valeurs en leur substituant leur valeur initiale	6-75
6.5	Affichage des références	6-76
6.5.1	Références possibles	6-76
6.5.2	Vue synoptique des adresses	6-77
6.5.3	Liste des références croisées	6-77
6.5.4	Opérandes utilisés	6-79
6.5.5	Structure du programme	6-81
6.5.6	Utilisation des données de référence	6-86
6.5.6.1	Positionnement rapide sur des occurrences dans le programme	6-86
6.5.6.2	Exemple d'utilisation d'occurrences	6-87
6.5.6.3	Marche à suivre pour utiliser des données de référence	6-89
6.6	Vérification de la cohérence du programme et horodatage comme propriété de bloc	6-90
6.6.1	Vérification de la cohérence du programme	6-90
6.6.2	Horodatage et conflits d'horodatage	6-91
6.6.3	Horodatage des blocs de code	6-92
6.6.4	Horodatage des blocs de données globaux	6-93
6.6.5	Horodatage des blocs de données d'instance	6-93
6.6.6	Horodatage des UDT et des DB dérivés d'UDT	6-94
6.6.7	Comment éviter des erreurs lors de l'appel de bloc	6-95
6.6.8	Remarques sur la modification du contenu des registres	6-97
7	Etablissement d'une liaison en ligne et choix de la CPU	7-1
7.1	Etablissement de liaisons en ligne	7-1
7.1.1	Protection de l'accès aux systèmes cible par mot de passe	7-2
7.2	Affichage de l'état de fonctionnement et changement	7-3
7.3	Affichage et réglage de l'heure et de la date	7-3

8	Importer, exporter, enregistrer sous	8-1
8.1	Importer, exporter, enregistrer sous	8-1
8.2	Enregistrer des projets sur un support de données	8-2
8.3	Enregistrer des données de projet sur une carte mémoire Micro (MMC)	8-2
8.4	Utiliser une carte mémoire Micro comme support de données	8-5
8.5	Echange de données de projet entre STEP 7 Lite et STEP 7	8-6
8.6	Exportation de données de projet pour éditeurs externes	8-8
8.6.1	Format de données pour l'importation/exportation d'une table des mnémoniques	8-8
8.6.2	Gestion multilingue des textes	8-8
8.6.2.1	Types de textes à gestion multilingue	8-10
8.6.2.2	Structure du fichier d'exportation	8-10
8.6.2.3	Détails sur la gestion multilingue des textes	8-12
8.6.2.4	Conseils pour la traduction	8-14
9	Chargement dans la CPU et la PG	9-1
9.1.1	Conditions requises pour le chargement	9-1
9.1.2	Charger quoi et quand dans la CPU ?	9-2
9.1.3	Différence entre l'enregistrement et le chargement de blocs	9-3
9.1.4	Mémoire de chargement et mémoire de travail dans la CPU	9-3
9.1.5	Possibilités de chargement selon la mémoire de chargement	9-4
9.1.6	Chargement de blocs et de la configuration dans la CPU et enregistrement sur Memory Card	9-5
9.1.6.1	Chargement a posteriori de blocs dans la CPU	9-5
9.1.6.2	Enregistrer des blocs chargés dans l'EPROM intégrée ou dans la carte mémoire de la CPU	9-6
9.1.6.3	Chargement d'une configuration dans un système cible	9-7
9.2	Chargement depuis la CPU dans la PG	9-8
9.2.1	Recharger quoi et quand dans la PG ?	9-9
9.2.2	Chargement d'objets de la CPU dans la PG/le PC	9-10
9.2.3	Edition de blocs chargés dans la PG/le PC	9-10
9.2.4	Edition d'une configuration matérielle chargée dans votre PG/PC	9-11
9.3	Effacement dans la CPU	9-12
9.3.1	Effacement de la mémoire de chargement/travail et effacement général de la CPU	9-12
9.3.2	Suppression de certains blocs dans la CPU	9-13
9.3.3	Effacement de la carte mémoire dans la CPU	9-14
9.4	Compression de la mémoire utilisateur (RAM)	9-14
9.4.1	Intervalles dans la mémoire utilisateur (RAM)	9-14
9.4.2	Compression du contenu de la mémoire d'une CPU	9-15

10	Test.....	10-1
10.1	Présentation des différents modes de test.....	10-1
10.2	Test avec des tables de variables et des tables de forçage	10-1
10.2.1	Introduction au test avec des tables de variables et des tables de forçage	10-1
10.2.2	Marche à suivre pour visualiser et forcer avec la table des variables	10-2
10.2.3	Marche à suivre de principe pour la visualisation et le forçage avec des tables de forçage	10-2
10.2.4	Edition et enregistrement de tables de variables et des tables de forçage	10-3
10.2.4.1	Création et ouverture d'une table de variables	10-3
10.2.4.2	Créer et ouvrir une table de forçage	10-3
10.2.4.3	Copier / Dupliquer une table de variables	10-4
10.2.4.4	Copier / Dupliquer une table de forçage	10-5
10.2.4.5	Enregistrement d'une table de variables.....	10-5
10.2.4.6	Enregistrer une table de forçage.....	10-5
10.2.5	Saisie de variables dans des tables de variables et des tables de forçage.....	10-6
10.2.5.1	Saisie d'opérandes ou de mnémoniques dans une table de variables.....	10-6
10.2.5.2	Saisie d'opérandes ou de mnémoniques dans une table de forçage	10-8
10.2.5.3	Insertion d'une plage d'opérandes continue dans une table de variables	10-9
10.2.5.4	Insertion d'un plage d'opérandes continue dans une table de forçage.....	10-9
10.2.5.5	Limites supérieures pour la saisie de temporisations	10-10
10.2.5.6	Limites supérieures pour la saisie de compteurs.....	10-11
10.2.5.7	Exemples.....	10-12
10.2.6	Edition de variables dans des tables de variables et des tables de forçage	10-16
10.2.6.1	Définition du format d'affichage.....	10-16
10.2.6.2	Couper des zones sélectionnées et les copier dans le presse-papiers.....	10-16
10.2.6.3	Coller le contenu du presse-papiers dans la table de variables ou dans la table de forçage.....	10-16
10.2.6.4	Copie de zones sélectionnées dans le presse-papiers.....	10-16
10.2.7	Visualisation de variables.....	10-17
10.2.7.1	Introduction à la visualisation de variables.....	10-17
10.2.7.2	Définition du mode de visualisation.....	10-17
10.2.7.3	Visualisation de variables.....	10-18
10.2.7.4	Visualisation unique et immédiate de variables	10-19
10.2.8	Forçage de variables.....	10-20
10.2.8.1	Introduction au forçage de variables	10-20
10.2.8.2	Définition du mode de forçage	10-20
10.2.8.3	Forçage de variables.....	10-22
10.2.8.4	Forçage immédiat de variables	10-23
10.2.8.5	Forçage : initialiser la CPU à l'arrêt avec vos propres valeurs	10-23
10.2.8.6	Forçage des sorties de périphérie à l'état "Arrêt" de la CPU	10-23
10.2.9	Forçage permanent de variables	10-24
10.2.9.1	Introduction au forçage permanent de variables.....	10-24
10.2.9.2	Mesures de sécurité pour le forçage permanent de variables	10-25
10.2.9.3	Affichage des valeurs forcées de manière permanente par la CPU	10-25
10.2.9.4	Création d'une tâche de forçage permanent.....	10-25
10.2.9.5	Annulation d'une tâche de forçage permanent	10-26
10.2.9.6	Différences entre forçage de variables et forçage permanent de variables	10-26

10.3	Test avec la visualisation d'état du programme	10-27
10.3.1	Test avec la visualisation d'état du programme	10-27
10.3.2	Affichage dans la visualisation d'état de programme.....	10-28
10.3.3	Etat du programme de blocs de données	10-30
10.3.4	Détails sur le test dans l'état du programme	10-31
10.3.4.1	Définition de l'affichage de l'état du programme	10-31
10.3.4.2	Définition de l'environnement d'appel du bloc.....	10-32
10.3.4.3	Choix du mode de fonctionnement pour le test.....	10-33
10.3.4.4	Forçage de variables dans l'état du programme.....	10-34
10.3.4.5	Activation et désactivation du test avec état du programme.....	10-34
11	Diagnostic.....	11-1
11.1	Fonctions de diagnostic.....	11-1
11.2	Diagnostic du matériel et recherche d'erreurs	11-2
11.3	Comparaison des configurations 'en ligne/hors ligne/physique'	11-2
11.4	La vue 'Comparaison HW'	11-4
11.5	Détecter les modules défectueux	11-5
11.6	La vue 'Diagnostic HW'	11-6
11.7	L'état du module	11-7
11.7.1	Appel de l'état du module	11-7
11.7.2	Fonctions d'information de l'état du module	11-9
11.7.3	Volume d'informations selon le type de module dans l'état du module	11-11
11.8	Diagnostic à l'état de fonctionnement STOP	11-12
11.8.1	Marche à suivre pour déterminer la cause d'un passage à l'état d'arrêt.....	11-12
11.8.2	Contenu des piles à l'état d'arrêt.....	11-12
11.8.3	Ouverture du bloc pour une entrée en mémoire tampon de diagnostic ou dans la pile	11-14
11.8.3.1	Ouverture du bloc pour une entrée en mémoire tampon de diagnostic.....	11-14
11.8.3.2	Ouverture du bloc dans la liste de la pile B.....	11-15
11.8.3.3	Ouverture du bloc dans la liste de la pile L	11-15
11.9	Contrôle des temps de cycle pour éviter les erreurs de temps.....	11-16
11.10	Transmission d'informations de diagnostic	11-17
11.10.1	Transmission d'informations de diagnostic	11-17
11.10.2	Liste d'états système (SZL)	11-18
11.10.3	Envoi de vos propres messages de diagnostic.....	11-20
11.11	Mesures à prendre dans le programme pour traiter les erreurs	11-21
11.11.1	Exploitation du paramètre de sortie RET_VAL	11-22
11.11.2	OB d'erreur en réaction à la détection d'une erreur	11-23
11.11.3	Insertion de valeurs de remplacement en cas d'erreur	11-26
11.11.4	Erreur de temps (OB80)	11-29
11.11.5	Erreur d'alimentation (OB81).....	11-29
11.11.6	Alarme de diagnostic (OB82)	11-30
11.11.7	Erreur matérielle CPU (OB84).....	11-31
11.11.8	Erreur d'exécution du programme (OB85)	11-32
11.11.9	Défaillance d'unité (OB86)	11-33
11.11.10	Erreur de communication (OB87)	11-34
11.11.11	Erreur de programmation (OB121)	11-35
11.11.12	Erreur d'accès à la périphérie (OB122).....	11-36

12	Imprimer une documentation du projet.....	12-1
12.1	Documentation du projet	12-1
12.2	Etablir la documentation du projet.....	12-3
12.3	Objets à imprimer	12-5
12.4	Options, police et mise en page.....	12-7
12.5	Définir des modèles et les utiliser	12-10
12.6	Imprimer une documentation du projet	12-13
13	Astuces et conseils	13-1
13.1	Remplacement de modules dans la table de configuration	13-1
13.2	Test à l'aide de la table des variables	13-1
13.3	Travailler sur la PG/le PC sans projet original	13-2
A	Annexe.....	A-1
A.1	Etats de fonctionnement.....	A-1
A.1.1	Etats de fonctionnement et changements d'état de fonctionnement	A-1
A.1.2	Etat de fonctionnement "Arrêt" (STOP).....	A-4
A.1.3	Etat de fonctionnement "Mise en route"	A-4
A.1.4	Etat de fonctionnement "Marche" (RUN)	A-9
A.1.5	Etat de fonctionnement "Attente"	A-11
A.2	Zones de mémoire des CPU S7	A-12
A.2.1	Organisation des zones de mémoire	A-12
A.2.2	Mémoire de chargement et mémoire de travail.....	A-13
A.2.3	Mémoire système	A-15
A.2.3.1	Utilisation des zones de mémoire système.....	A-15
A.2.3.2	Mémoire image des entrées/sorties	A-17
A.2.3.3	Pile des données locales.....	A-18
A.2.3.4	Pile des interruptions.....	A-19
A.2.3.5	Pile des blocs	A-20
A.2.3.6	Mémoire tampon de diagnostic	A-21
A.2.3.7	Exploitation de la mémoire tampon de diagnostic	A-21
A.2.3.8	Zones de mémoire rémanentes des CPU S7-300	A-23
A.3	Types de données et de paramètre	A-26
A.3.1	Introduction aux types de données et de paramètre.....	A-26
A.3.2	Types de données simples.....	A-27
A.3.2.1	Format du type de données INT (entiers de 16 bits)	A-28
A.3.2.2	Format du type de données DINT (entiers de 32 bits)	A-28
A.3.2.3	Format du type de données REAL (nombres à virgule flottante).....	A-29
A.3.2.4	Format du type de données WORD.....	A-33
A.3.2.5	Format du type de données DWORD	A-33
A.3.2.6	Format des types de données WORD et DWORD pour les nombres décimaux codés binaire	A-34
A.3.2.7	Format du type de données S5TIME (durée).....	A-35
A.3.2.8	Format du type de données TIME	A-36
A.3.3	Types de données complexes	A-37
A.3.3.1	Format du type de données DATE_AND_TIME (date et heure).....	A-38
A.3.3.2	Format du type de données STRING.....	A-39
A.3.3.3	Format du type de données ARRAY	A-40
A.3.3.4	Format du type de données STRUCT.....	A-41
A.3.3.5	Utilisation de types de données complexes	A-42
A.3.3.6	Utilisation de tableaux pour l'accès aux données	A-43
A.3.3.7	Utilisation de structures pour l'accès aux données	A-46
A.3.3.8	Utilisation de types de données utilisateur pour l'accès aux données.....	A-48

A.3.4	Types de paramètre	A-50
A.3.4.1	Format des types de paramètre BLOCK, COUNTER et TIMER.....	A-51
A.3.4.2	Format du type de paramètre POINTER.....	A-51
A.3.4.3	Utilisation du type de paramètre POINTER	A-53
A.3.4.4	Bloc pour modifier le pointeur	A-54
A.3.4.5	Format du type de paramètre ANY	A-57
A.3.4.6	Utilisation du type de paramètre ANY	A-60
A.3.4.7	Affectation de types de données aux données locales de blocs de code	A-63
A.3.4.8	Types de données autorisés pour la transmission de paramètres	A-65
A.3.4.9	Transmission au paramètre IN_OUT d'un FB.....	A-70
A.4	Exemples de programme	A-71
A.4.1	Exemples de projet et de programme.....	A-71
A.4.2	Exemple de masquage et de démasquage d'événements d'erreurs synchrones.....	A-72
A.4.3	Exemple d'inhibition et de validation d'événements d'alarme et d'événements asynchrones (SFC 39 et 40).....	A-75
A.4.4	Exemple de traitement différé d'événements d'alarme et d'événements asynchrones (SFC 41 et 42).....	A-76
A.4.5	Exemple de programme pour un processus de mélange industriel.....	A-77
A.4.5.1	Exemple de programme pour un processus de mélange industriel.....	A-77
A.4.5.2	Définition de blocs de code	A-80
A.4.5.3	Affectation de mnémoniques.....	A-81
A.4.5.4	Création du bloc fonctionnel pour le moteur	A-84
A.4.5.5	Création de la fonction pour les soupapes.....	A-88
A.4.5.6	Création de l'OB1	A-90
A.4.6	Exemple d'utilisation d'alarmes horaires	A-96
A.4.6.1	Structure du programme utilisateur 'Alarme horaire'.....	A-96
A.4.6.2	FC12.....	A-98
A.4.6.3	OB10	A-100
A.4.6.4	OB1 et OB80	A-102
A.4.7	Exemple d'utilisation d'alarmes temporisées	A-103
A.4.7.1	Structure du programme utilisateur 'Alarme temporisée'	A-103
A.4.7.2	OB20	A-105
A.4.7.3	OB1	A-107
A.5	Accès aux zones de données du processus et de périphérie.....	A-109
A.5.1	Accès à la zone de données du processus.....	A-109
A.5.2	Accès à la zone de données de périphérie	A-110
A.6	Définition du comportement en fonctionnement	A-112
A.6.1	Définition du comportement en fonctionnement	A-112
A.6.2	Modification du comportement et des propriétés des modules	A-112
A.6.3	Avantage des fonctions d'horodatage.....	A-114
A.6.4	Utilisation de mémentos de cadence et de temporisations.....	A-115

Index

1 Introduction sur le produit et installation

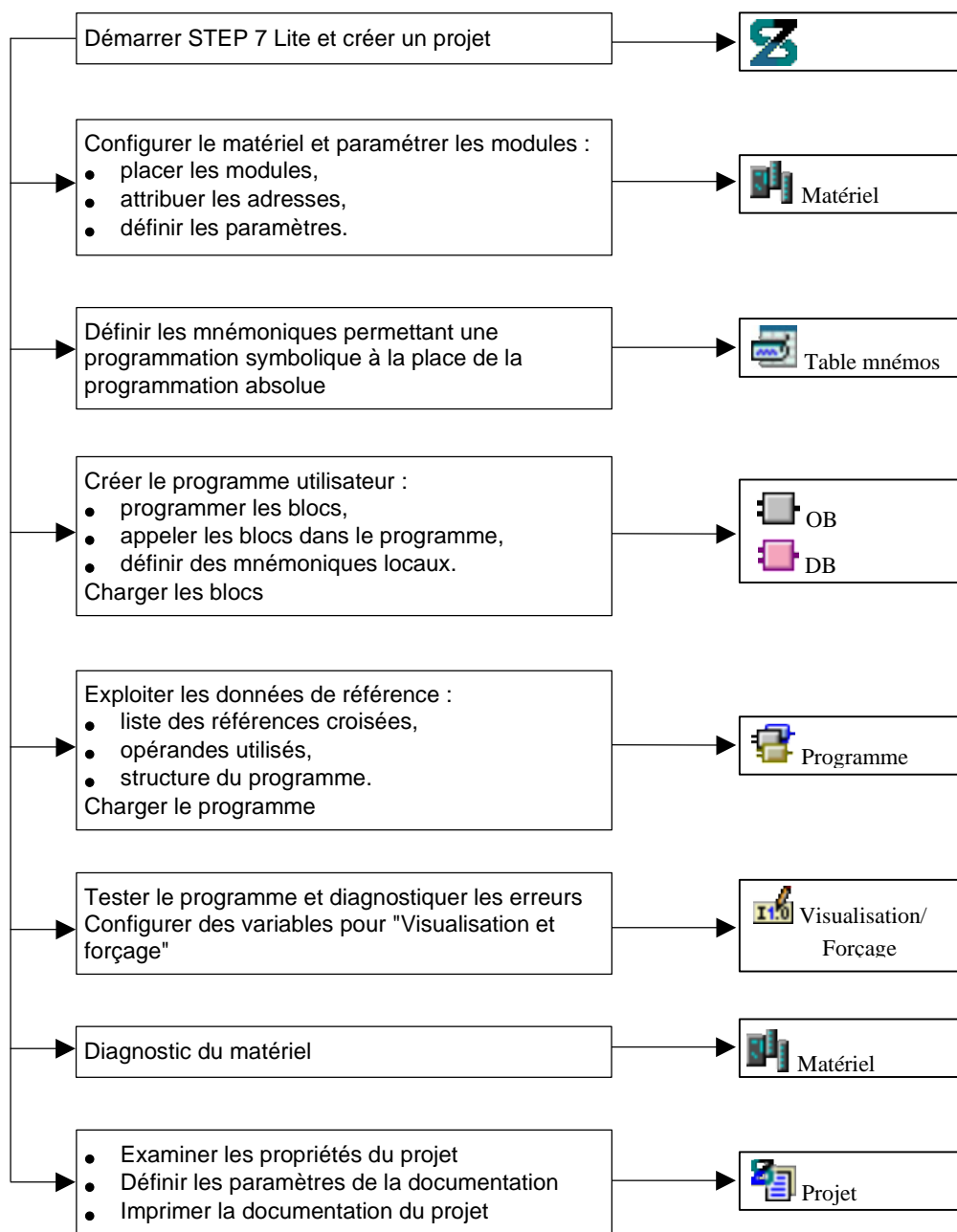
1.1 Présentation de STEP 7 Lite

Matériel pris en charge

Le logiciel de configuration et de programmation STEP 7 Lite s'applique aux systèmes d'automatisation SIMATIC des familles S7-300 et C7 ainsi qu'aux familles ET 200X et ET 200S (autonome).

Etapes de la réalisation d'une solution d'automatisation

Pour créer une solution d'automatisation avec STEP 7 Lite, vous devrez exécuter différentes tâches. Dans la figure ci-après, celles qui se présentent dans la plupart des projets sont énumérées dans un ordre qui peut être pris comme exemple.



Description succincte des différentes étapes

- Installation et autorisation
La première fois que vous utilisez STEP 7 Lite, vous devez l'installer et transférer l'autorisation de la disquette sur le disque dur (voir aussi Installation de STEP 7 Lite et Autorisation).
- Conception de la structure du programme
Dans la conception de la commande d'automatisation, vous décrivez les tâches que vous transformerez en structure de programme à l'aide des blocs que STEP 7 Lite met à votre disposition (voir aussi Blocs dans le programme utilisateur).
- Démarrage de STEP 7 Lite
Vous démarrez STEP 7 Lite depuis l'interface Windows (voir aussi Démarrage de STEP 7 Lite).

- **Création d'un projet**
Un projet est comme un dossier dans lequel toutes les données sont stockées selon une structure hiérarchique et restent toujours disponibles. Quand vous avez créé un projet, toutes les étapes suivantes y sont exécutées (voir aussi Qu'est-ce qu'un projet STEP 7 Lite ?).
- **Création d'une station**
En créant une station, vous déterminez le type d'automate à mettre en œuvre : par ex. SIMATIC S7-300.
- **Configuration matérielle**
Au cours de la configuration, vous déterminez les modules que vous souhaitez employer pour votre solution d'automatisation ainsi que les adresses permettant de communiquer avec ces modules depuis le programme utilisateur. En outre, des paramètres servent à définir les propriétés des modules (voir aussi Manipulations de base pour la configuration matérielle).
- **Définition de mnémoniques**
Vous pouvez définir, dans une table, des mnémoniques locaux ou globaux très expressifs afin de les utiliser dans votre programme à la place des adresses (voir aussi Ouverture d'une table des mnémoniques).
- **Création d'un programme**
Vous élaborez un programme dans l'un des langages de programmation disponibles et vous le stockez sous forme de blocs (voir aussi Marche à suivre pour créer des blocs de code).
- **Exploitation des données de référence**
Ces données peuvent vous faciliter le test et la modification de votre programme (voir aussi Données de référence possibles).
- **Chargement du programme dans la CPU**
Quand vous avez terminé la configuration, le paramétrage et l'élaboration du programme, vous pouvez transférer le programme entier ou certains blocs seulement dans la CPU (voir aussi Conditions requises pour le chargement). La CPU contient déjà le système d'exploitation.
- **Test du programme**
Pour cela, vous avez la possibilité d'afficher les valeurs de certaines variables de votre programme utilisateur ou d'une CPU, de leur affecter des valeurs et d'établir une table de variables contenant celles que vous souhaitez visualiser ou forcer (voir aussi Introduction au test avec des tables de variables).
- **Surveillance du déroulement, diagnostic du matériel**
Vous trouverez la cause d'un défaut de module en affichant des informations en ligne sur ce module. Pour trouver la cause d'un défaut dans l'exécution du programme, consultez le contenu de la mémoire tampon de diagnostic et celui des piles. Vous pouvez contrôler en outre si un programme est exécutable sur une CPU particulière (voir aussi Diagnostic du matériel et recherche d'erreurs).
- **Impression**

Langages de programmation

Les langages de programmation SIMATIC intégrés dans STEP 7 Lite satisfont à la norme EN 6.1131-3.

- Le schéma à contacts (CONT) est un langage de programmation graphique. La syntaxe des instructions fait penser aux schémas de circuits. CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts, les éléments complexes et les bobines.
- La liste d'instructions (LIST) est un langage de programmation textuel proche de la machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes suivant lesquelles la CPU traite le programme. Pour faciliter la programmation, LIST a été complété par quelques structures de langage évolué (comme, par exemple, des paramètres de blocs et accès structurés aux données).
- Le logigramme (LOG) est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques.

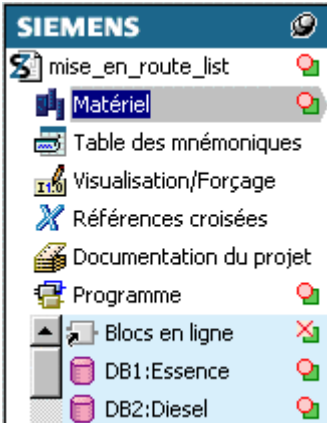
1.2 Fenêtre du projet et vues dans STEP 7 Lite

Fenêtre du projet et vues

Une fois que vous avez démarré STEP 7 Lite, le volet gauche de la fenêtre présente la fenêtre du projet.

Tous les objets essentiels (par ex. le programme) et les fonctions essentielles (par ex. visualisation et forçage) sont accessibles dans STEP 7 Lite par l'intermédiaire de cette fenêtre du projet.

En cliquant sur l'onglet "CPU en ligne" au bas de la fenêtre du projet, à côté de l'onglet "Projet", vous affichez les objets se trouvant dans la CPU (voir Commuter sur CPU en ligne).

Fenêtre du projet (onglet Projet)	Renvois aux explications
	<ul style="list-style-type: none"> Projet Matériel Table des mnémoniques Visualisation et forçage Références croisées Documentation du projet Programme Blocs Icônes utilisées dans la fenêtre du projet Importer, exporter, enregistrer sous

Un double-clic sur un élément dans la fenêtre du projet ouvre dans la zone de travail une fenêtre contenant une ou plusieurs vues. Lorsqu'il y a plusieurs vues sur un élément, vous pouvez passer de l'une à l'autre à l'aide des onglets situés au bas de la fenêtre.

Exemple

L'élément "Matériel" comporte les vues "Configuration matérielle", "Comparaison matérielle" et "Diagnostic du matériel".

Projet

Vous pouvez modifier la désignation par défaut "Nouveau projet" au moyen de la commande de menu **Fichier > Enregistrer** ou **Enregistrer sous**.

Vues possibles	Signification	Voir aussi
Vue d'ensemble du projet	Configuration matérielle ainsi que tableau de tous les blocs du projet avec indication de leurs propriétés, comme par ex. mnémonique du bloc, taille, langage de création, etc.	Blocs dans le programme utilisateur

Matériel

Vous ne pouvez pas modifier la désignation par défaut dans la fenêtre du projet.

Vues possibles	Signification	Voir aussi
Comparaison HW	Pour mettre en face l'une de l'autre la configuration paramétrée et celle chargée dans la CPU ; mise en évidence des divergences (par ex. types de module différents ou modules manquants).	Comparaison entre configuration chargée et configuration paramétrée
Configuration HW	Pour disposer des modules tirés d'un catalogue dans des profilés supports et paramétrer ces modules.	Introduction à la configuration matérielle Définir les propriétés de composants
Diagnostic HW	Pour consulter les états des modules ; il est possible d'appeler des informations de diagnostic détaillées depuis cette vue.	Diagnostic du matériel et recherche d'erreurs

Table des mnémoniques

Vous ne pouvez pas modifier la désignation par défaut dans la fenêtre du projet.

Vues possibles	Signification	Voir aussi
Table des mnémoniques	Pour afficher et éditer des mnémoniques globaux pour les blocs et les opérandes	Table des mnémoniques globaux

Visualisation et forçage

Vous ne pouvez pas modifier la désignation par défaut dans la fenêtre du projet.

Vues possibles	Signification	Voir aussi
Visualisation/ Forçage	Pour visualiser l'état de certains opérandes contenus dans une table de variables et pour forcer des opérandes	Introduction à la visualisation de variables Introduction au forçage de variables
Forçage permanent	Pour le forçage permanent de certains opérandes et pour annuler des tâches de forçage permanent	Introduction au forçage permanent de variables

Références croisées

Vous ne pouvez pas modifier la désignation par défaut dans la fenêtre du projet.

Vues possibles	Signification	Voir aussi
Liste des références croisées	Indique pour tous les opérandes utilisés dans quel bloc et à quelle position ils le sont.	Liste des références croisées Réassignation
Opérandes utilisés	Indique les opérandes avec leur type d'accès (niveau bit, octet, mot ou double mot).	Opérandes utilisés
Structure du programme	Indique sous forme arborescente tous les blocs présents avec la hiérarchie d'appel qui les relie.	Structure du programme

Documentation du projet

Vous ne pouvez pas modifier la désignation par défaut dans la fenêtre du projet.

Vues possibles	Signification	Voir aussi
Documentation du projet	Pour établir la documentation du projet ; permet de disposer les objets à imprimer, tels que les références croisées ou la table des mnémoniques, de gérer des modèles de documentation, d'ajuster une mise en page pour l'impression.	Documentation du projet

Programme

Vous ne pouvez pas modifier la désignation par défaut dans la fenêtre du projet..

Vues possibles	Signification	Voir aussi
Vue d'ensemble du projet	Tableau de tous les blocs du projet avec indication de leurs propriétés, comme par ex. mnémonique du bloc, taille, langage de création, etc.	Blocs dans le programme utilisateur Vérification de la cohérence du programme

Blocs

La désignation utilisée dans la fenêtre du projet résulte du type et du numéro que vous avez attribués à chaque bloc en le générant.

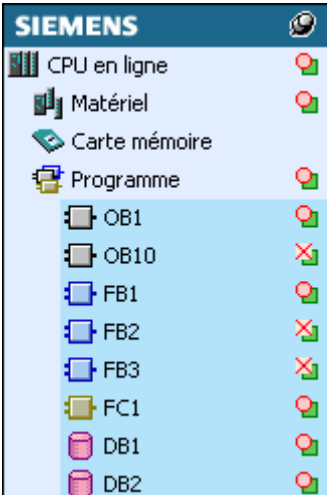
Vues possibles	Signification	Voir aussi
Editeur de bloc	Editeur avec section de déclaration et section d'instructions pour élaborer le programme au sein du bloc sélectionné	Editeur de bloc Editer des instructions LIST dans la section des instructions Editer des instructions LOG dans la section des instructions Editer des instructions CONT dans la section des instructions Test avec la visualisation d'état du programme
Propriétés	Pour afficher les propriétés de bloc avec nom, longueurs, mémoire requise, etc. et pour saisir ou modifier le mnémonique, divers commentaires et les attributs de bloc.	Propriétés de bloc Paramétrage des propriétés de bloc

Commuter sur "CPU en ligne"

Pour pouvez changer de fenêtre du projet. Au bas de la fenêtre, cliquez sur l'onglet "CPU en ligne" pour pouvoir accéder aux contenus de la CPU (blocs chargés et configuration matérielle chargée), à la carte mémoire ou à la carte mémoire Micro (MMC) dans la CPU. Contrairement à la fenêtre hors ligne du projet, la vue en ligne est sur fond de couleur.

Vous ne pouvez pas effectuer de modification dans la vue "CPU en ligne".

Exemple :

Vue d'ensemble du projet (CPU en ligne)	Renvois aux explications
	<p>Particularités :</p> <p>Les symboles s'affichant au bord droit de la vue en ligne indiquent par ex. l'égalité ou la différence entre l'objet hors ligne et l'objet en ligne (tel que le programme ou la configuration HW).</p> <p>Quand le projet contient des blocs qui n'ont pas été chargés dans la CPU, l'icône "Blocs hors ligne" le signale.</p> <p>Matériel</p> <p>Carte mémoire Micro (MMC)</p> <p>Programme</p> <p>Blocs</p> <p>Icônes dans la fenêtre du projet</p> <p>Conditions requises pour le chargement</p> <p>Travailler sur la PG/le PC sans projet original</p>

1.3 Aide de STEP 7 Lite

La documentation de STEP 7 Lite est à votre disposition sous forme d'aide en ligne en deux parties :

- la combinaison de touches MAJ + F1 ou le pointeur d'aide affichent des conseils relatifs au contexte fonctionnel ;
- la commande ? > **Aide de STEP 7 Lite** ouvre les contenus plus généraux de l'aide qui sont basés sur le format HTML.

En plus de l'aide en ligne, vous disposez de manuels électroniques au format PDF. Vous les trouverez en choisissant **Démarrer > Simatic > Documentation** dans la barre des tâches.

Pour obtenir la documentation sur papier, adressez-vous, comme d'habitude, à votre interlocuteur Siemens.

Si vous souhaitez des renseignements plus détaillés sur l'emploi de la documentation, veuillez consulter le chapitre "Remarques sur la documentation" dans le fichier LISEZMOI.WRI du CD de STEP 7 Lite. Ce fichier mentionne aussi les éventuelles modifications de dernière minute apportées à l'aide en ligne et aux manuels électroniques, modifications qui n'ont pu être prises en compte dans cette édition.

1.4 Installation

1.4.1 Automation License Manager

1.4.1.1 Licence d'utilisation avec Automation License Manager

Automation Licence Manager

Pour pouvoir utiliser le logiciel de programmation STEP 7 Lite, une "License Key" spécifique au produit (licence d'utilisation) est nécessaire. A partir de la version V3.0 de STEP 7 Lite, celle-ci est installée par Automation License Manager.

Automation License Manager est un produit logiciel de Siemens AG. Il permet de gérer des "License Keys" (représentants techniques de licences) dans l'ensemble du système.

Automation License Manager est disponible :

- sur le support d'installation du produit logiciel correspondant pour lequel une "License Key" est requise ou
- sur un support d'installation distinct, ainsi que
- sur les pages Internet de A&D Customer Support de Siemens AG, sous forme de produit à télécharger.

Automation License Manager dispose d'une aide en ligne contextuelle intégrée, que vous pouvez appeler après l'installation en appuyant sur la touche F1 ou en choisissant la commande de menu ? > **Aide de License Manager**. Vous y trouverez des informations détaillées sur la fonctionnalité et l'utilisation de Automation License Manager.

Licences

Des licences sont nécessaires pour l'utilisation des progiciels de STEP 7 Lite protégés par des droits de licence. Une licence est accordée comme droit d'utilisation de produits. Les représentants de ce droit sont :

- CoL (**C**ertificate **o**f **L**icense) et
- License Key

Certificate of License (CoL)

Le "Certificate of License" contenu dans la livraison des produits respectifs constitue la preuve juridique du droit d'utilisation. Seul le propriétaire du CoL ou les personnes mandatées sont autorisées à utiliser le produit respectif.

License Keys

La "License Key" est le représentant technique d'une licence (marque de licence électronique).

SIEMENS AG attribue une "License Key" pour tout logiciel protégé par des droits de licence. Le logiciel correspondant ne peut être utilisé conformément aux conditions de licence et d'utilisation liées à la "Licence Key" correspondante que si la présence de la "License Key" valide est constatée sur un ordinateur, après démarrage du logiciel.

Nota

- Vous avez également la possibilité d'utiliser le logiciel de base de STEP 7 Lite sans "License Key" pour en découvrir l'interface utilisateur et les fonctionnalités.
 - L'utilisation sans restrictions, conforme aux déclarations de droits de licence, n'est cependant autorisée et possible que lorsque la "License Key" est installée.
 - Si vous n'avez **pas** installé la "License Key", vous êtes sollicité à intervalles réguliers de réaliser l'installation.
-

Les "License Keys" sont disponibles sur les supports de données suivants et peuvent être transférées d'un support à l'autre :

- disquettes "License Key",
- disques durs locaux et
- disques durs d'ordinateurs dans le réseau.

Si vous avez installé des produits logiciels pour lesquels vous ne disposez pas de "License Key", vous pouvez encore commander après-coup les licences requises.

Des informations supplémentaires sur l'utilisation des "License Keys" sont fournies dans l'aide en ligne de Automation License Manager.

Types de licences

Pour les produits logiciels de Siemens AG, il existe les types suivants de licences orientées application. Le comportement du logiciel est déterminé par les diverses "License Keys" correspondant à ces types de licences. Le type d'utilisation est défini par le "Certificate of License" respectif.

Type de licence	Description
Single License	L'utilisation du logiciel est autorisée pour une durée illimitée sur un ordinateur quelconque.
Floating License	L'utilisation du logiciel est autorisée pour une durée illimitée sur un réseau (utilisation "à distance").
Trial License	L'utilisation du logiciel est restreinte : <ul style="list-style-type: none">• à une validité de 14 jours au maximum,• à un nombre de jours défini à partir de la première utilisation,• à l'utilisation pour test et validation (à l'exclusion de toute responsabilité).
Upgrade License	Une mise à niveau peut être nécessaire en raison d'exigences spécifiques de l'état du système : <ul style="list-style-type: none">• Une licence de mise à niveau permet de transformer une licence d'une "ancienne" version x en une licence de version >x+....• Une mise à niveau peut p. ex. être requise en raison d'extension des capacités.

1.4.1.2 Installation de Automation License Manager

L'installation de Automation License Manager est réalisée par un Setup MSI. Le logiciel d'installation de Automation License Manager se trouve sur le CD du produit STEP 7 Lite.

Vous pouvez installer Automation License Manager avec STEP 7 Lite ou ultérieurement.

Nota

- Des informations détaillées sur la procédure d'installation de Automation License Manager sont données dans le fichier Lisezmoi.wri actuel.
 - Tous les informations relatives à la fonctionnalité et à l'utilisation des "License Keys" de Automation License Manager sont données dans l'aide en ligne.
-

Installation ultérieure de "License Keys"

Si vous démarrez le logiciel STEP 7 Lite et qu'aucune "License Key" n'est présente, vous obtenez un message correspondant.

Nota

- Vous avez également la possibilité d'utiliser le logiciel de base de STEP 7 Lite sans "License Key" pour en découvrir l'interface utilisateur et les fonctionnalités.
 - L'utilisation sans restrictions, conforme aux déclarations de droits de licence, n'est cependant autorisée et possible que lorsque la "License Key" est installée.
 - Si vous n'avez **pas** installé la "License Key", vous êtes sollicité à intervalles réguliers de réaliser l'installation.
-

Pour installer les "License Keys" ultérieurement, vous avez les possibilités suivantes :

- installez les "License Keys" depuis les disquettes,
- installez les "License Keys" en les téléchargeant depuis Internet (une commande préalable est requise),
- utilisez des "Floating License Keys" disponibles sur Internet.

Vous trouverez des informations détaillées sur la procédure dans l'aide en ligne contextuelle de Automation License Manager, que vous pouvez appeler après l'installation en appuyant sur la touche F1 ou en choisissant la commande de menu ? > **Aide de License Manager**.

Nota

- Les "License Keys" sont uniquement fonctionnelles sous Windows 2000/XP si elles se trouvent sur un disque dur possédant des droits d'accès en écriture.
 - Les "Floating Licenses" peuvent également être utilisées via un réseau, c'est-à-dire "à distance".
-

1.4.1.3 Règles d'utilisation des License Keys



Avertissement

Tenez compte des indications d'utilisation des "License Keys" décrites dans l'aide en ligne de Automation License Manager et dans le fichier Lisezmoi.wri sur le CD-ROM de STEP 7 Lite. Le non-respect de ces règles risque d'entraîner la perte irréversible des "License Keys".

Vous pouvez appeler l'aide en ligne contextuelle de Automation License Manager en appuyant sur la touche F1 ou en choisissant la commande de menu ? > **Aide de License Manager**.

Cette aide en ligne contient toutes les informations relatives à la fonctionnalité et à l'utilisation des "License Keys".

1.4.2 Installation de STEP 7 Lite

STEP 7 Lite contient un programme SETUP qui exécute l'installation automatiquement. Des messages s'affichant à l'écran vous guident étape par étape tout au long de la procédure d'installation. Vous l'appellez via la procédure d'installation de logiciel standard sous Windows.

Les phases principales de l'installation sont :

- la copie des données dans votre outil de développement,
- la saisie du numéro d'identification,
- l'installation des pilotes pour la communication,
- l'autorisation (si cela vous convient à ce moment-là).

Conditions préalables à l'installation

- Le progiciel tourne sous le système d'exploitation
 - Microsoft Windows XP Home
 - Microsoft Windows XP Professional
 - Microsoft Windows 2000
- Matériel de base
PC ou console de programmation avec les exigences requises pour le système d'exploitation. Ces dernières sont précisées dans la documentation du système d'exploitation ou sur les pages WEB de Microsoft.

Une console de programmation (PG) est un ordinateur personnel compact tout spécialement conçu pour être utilisé dans un environnement industriel. Elle est équipée en série de tous les programmes nécessaires à la programmation de systèmes d'automatisation SIMATIC.

- Mémoire requise
Pour l'espace mémoire nécessaire sur le disque dur voir le fichier LISEZMOI.WRI.
- Interface MPI (optionnelle)
L'interface multipoint MPI entre l'outil de développement (console de programmation ou ordinateur personnel) et le système cible n'est nécessaire que si vous voulez communiquer, sous STEP 7 Lite et via MPI, avec le système cible.
A cet effet, vous devez utiliser :
 - un adaptateur PC et un câble modem neutre (RS232) reliés à l'interface de communication de votre console ou
 - une carte MPI (par ex. CP 5611) installée dans votre console.L'interface MPI est déjà intégrée à certaines consoles de programmation.

Nota

Veuillez également tenir compte des remarques sur l'installation de STEP 7 Lite figurant dans le fichier LISEZMOI.WRI.

Vous trouverez le fichier Lisezmoi en cliquant dans la barre des tâches sur **Démarrer > Simatic > Informations**.

1.4.2.1 Marche à suivre pour installer STEP 7 Lite

Préparatifs

Vous devez démarrer le système d'exploitation Windows avant de commencer l'installation.

- Un support de données externe est inutile si le logiciel STEP 7 Lite installable se trouve déjà sur le disque dur de la PG.
- Pour effectuer l'installation à partir du CD-ROM, insérez le CD-ROM dans le lecteur de CD-ROM de votre PC.

Lancement du programme d'installation

Procédez comme suit pour lancer l'installation :

1. Insérez le CD-ROM et lancez le Setup en cliquant sur "setup.exe".
2. Suivez étape par étape les instructions affichées par le programme d'installation.

Ce programme vous guide pas à pas tout au long de la procédure d'installation. Vous avez toujours la possibilité de revenir à l'étape précédente ou d'aller à l'étape suivante.

Pendant l'installation, des questions vous sont posées ou des options proposées dans des boîtes de dialogue. Tenez compte des indications ci-après qui vous permettront de répondre plus rapidement et aisément aux dialogues.

Version de STEP 7 Lite déjà installée...

Si le programme d'installation constate qu'une version de STEP 7 Lite se trouve déjà sur l'outil de développement, un message vous le signale et vous avez les possibilités suivantes :

- interrompre l'installation pour, ensuite, désinstaller l'ancienne version de STEP 7 Lite sous Windows puis relancer l'installation ou
- poursuivre l'installation et substituer ainsi la nouvelle version à l'ancienne.

Une maintenance correcte du logiciel exigerait que vous désinstalliez toute version antérieure existante avant de procéder à une nouvelle installation. L'écrasement pur et simple d'une ancienne version présente, en outre, l'inconvénient qu'une désinstallation ultérieure n'effacerait pas les parties éventuellement encore existantes d'une installation précédente.

Choisir son installation

Trois variantes d'installation au choix sont possibles :

- Standard : toutes les langues de l'interface utilisateur ; l'information produit actuelle indique la place mémoire requise à cet effet.
- Compacte : une seule langue ; l'information produit actuelle indique la place mémoire requise à cet effet.
- Personnalisée : c'est vous qui choisissez les programmes, la base de données, etc. à installer.

Numéro d'identification

Un numéro d'identification vous est demandé durant l'installation. Entrez ce dernier. Vous le trouverez sur le certificat du logiciel ou sur la disquette d'autorisation correspondante.

Installation de "License Keys"

Lors de l'installation, le programme vérifie si une "License Key" correspondante existe sur le disque dur. Si aucune "License Key" valide n'est reconnue, un message vous avertit que vous ne pouvez utiliser le logiciel qu'en présence d'une "License Key". Vous pouvez, si vous le désirez, installer immédiatement les "License Keys" ou bien poursuivre l'installation de STEP 7 Lite et installer les "License Keys" ultérieurement. Dans le premier cas, insérez la disquette "License Key" fournie lorsque le message correspondant vous y invite.

Paramétrage de l'interface PG/PC

Une boîte de dialogue de paramétrage de l'interface PG/PC s'affiche pendant l'installation. Lisez à cet effet "Paramétrage de l'interface PG/PC".

Erreurs pendant l'installation

Les erreurs suivantes entraînent l'interruption de l'installation :

- Si une erreur d'initialisation se produit immédiatement après le démarrage du SETUP, vous avez certainement lancé l'installation dans un environnement autre que Windows.
- L'espace mémoire est insuffisant : selon l'option d'installation choisie, vous avez besoin d'environ 100 Mo d'espace libre sur votre disque dur pour le logiciel de base.
- CD défectueux : adressez-vous à votre agence Siemens si vous constatez qu'un CD est défectueux.
- Erreur de manipulation : recommencez l'installation en observant rigoureusement les instructions.

Fin de l'installation

Un message s'affiche à l'écran pour vous signaler que l'installation a réussi.

Si l'installation a entraîné l'actualisation des fichiers système, vous êtes invité à relancer Windows. Une fois Windows redémarré (démarrage à chaud), vous pouvez démarrer STEP 7 Lite.

Une installation sans erreur s'achève par la création d'un groupe de programmes pour STEP 7 Lite.

1.4.2.2 Paramétrage de l'interface PG/PC

Le paramétrage que vous réalisez ici sert à définir la communication entre PG/PC et système d'automatisation. Une boîte de dialogue de paramétrage de l'interface PG/PC s'affiche pendant l'installation. Vous pouvez aussi afficher cette boîte après l'installation en appelant le programme "Paramétrage de l'interface PG/PC" dans le groupe de programmes STEP 7 Lite ou dans le Panneau de configuration. Ce programme vous permet de modifier les jeux de paramètres après coup, indépendamment d'une quelconque installation.

Principe

L'utilisation d'une interface nécessite :

- des paramétrages dans le système d'exploitation,
- un jeu de paramètres adéquat.

Quand vous utilisez une PG via une liaison MPI, il n'est pas nécessaire de procéder à d'autres adaptations spécifiques au système d'exploitation.

Quand vous utilisez un PC avec une carte MPI ou des processeurs de communication (CP), vous devez vérifier l'affectation des interruptions et des adresses dans le "Panneau de configuration" de Windows, pour vous assurer qu'il n'y a pas de conflits d'interruptions ni de recouvrement de plages d'adresses.

Des jeux de paramètres prédéfinis vous sont proposés dans la boîte de dialogue afin de simplifier le paramétrage de l'interface PG/PC.

Paramétrage de l'interface PG/PC

Procédez de la manière suivante (une description plus détaillée est donnée dans l'aide en ligne) :

1. Dans le "Panneau de configuration", effectuez un double clic sur "Paramétrage de l'interface PG/PC".
3. Sélectionnez "S7ONLINE" comme "Entrée de l'application".
4. Sélectionnez le jeu de paramètres souhaité dans la liste "Jeux de paramètres utilisés". Si le jeu de paramètres souhaité ne figure pas dans la liste proposée, vous devez d'abord installer un module ou un protocole en cliquant sur le bouton "Sélectionner". Le jeu de paramètres est alors automatiquement créé.
 - Si vous sélectionnez une interface **avec détection automatique des paramètres de bus**, (par exemple un CP 5611 (Auto)), vous pouvez connecter la PG ou le PC au réseau MPI ou PROFIBUS sans devoir sélectionner ces paramètres de bus. Pour des vitesses de transmission inférieures à 187,5 kbit/s, des temps d'attente allant jusqu'à une minute ne sont pas exclus.
Condition pour la détection automatique : les maîtres connectés au bus répartissent les paramètres de bus de manière cyclique ; tous les nouveaux composants MPI le font ; la répartition cyclique des paramètres de bus ne doit pas être désactivée pour les sous-réseaux PROFIBUS (paramétrage par défaut du réseau PROFIBUS).
 - Lorsque vous sélectionnez une interface **sans détection automatique des paramètres de bus**, vous pouvez afficher les paramètres afin de les adapter au sous-réseau.

Des modifications sont également indispensables en cas de conflit avec d'autres paramètres (par exemple, affectation d'interruptions ou d'adresses). Dans ce cas, effectuez les modifications requises en utilisant la fonction d'ajout de nouveau matériel et le panneau de configuration de Windows (voir ci-après).



Avertissement

Ne supprimez en aucun cas le jeu de paramètres "TCP/IP" éventuellement présent !
Vous risqueriez de perturber l'exécution des autres applications.

Contrôle de l'affectation des interruptions et adresses

Lorsque vous utilisez un PC avec carte MPI, vous devez absolument vérifier si l'interruption et la plage d'adresses prédéfinies sont libres et, si ce n'est pas le cas, choisir une interruption et une plage d'adresses libres.

Windows 2000

Sous Windows 2000, vous pouvez afficher les ressources via **Panneau de configuration > Administrative Tools > Gestion de l'ordinateur > Outils système > Informations système > Ressources matérielles**.

Windows XP

Sous Windows XP, vous pouvez afficher les ressources sous **Démarrer > Tous les programmes > Accessoires > Outils système > Programmes système > Informations système > Ressources matérielles**.

1.4.3 Désinstallation de STEP 7 Lite

Utilisez la procédure de désinstallation courante sous Windows :

1. Lancez, sous Windows, le dialogue d'installation de logiciel en effectuant un double clic sur l'icône "Ajout/Suppression de programmes" dans le "Panneau de configuration".
2. Sélectionnez l'entrée STEP 7 Lite dans la liste affichée des logiciels installés. Cliquez sur le bouton de suppression du logiciel.
3. Si des boîtes de dialogue de suppression de fichiers autorisés apparaissent, cliquez sur le bouton "Non" en cas de doute.

2 Principes de conception d'une programme

2.1 Programmes dans une CPU

Deux programmes différents s'exécutent dans une CPU :

- le système d'exploitation et
- le programme utilisateur.

Système d'exploitation

Le système d'exploitation, contenu dans chaque CPU, organise toutes les fonctions et procédures dans la CPU qui ne sont pas liées à une tâche d'automatisation spécifique. Ses tâches sont les suivantes :

- le déroulement de la mise en route,
- l'actualisation de la mémoire image des entrées et l'émission de la mémoire image des sorties,
- l'appel du programme utilisateur,
- l'enregistrement des alarmes et l'appel des OB d'alarme,
- la détection et le traitement d'erreurs,
- la gestion des zones de mémoire,
- la communication avec des consoles de programmation et d'autres partenaires de communication.

La modification des paramètres par défaut du système d'exploitation permet d'influer sur le comportement de la CPU dans des domaines précis.

Programme utilisateur

Vous devez créer votre programme utilisateur et le charger dans la CPU. Il contient toutes les fonctions nécessaires au traitement de votre tâche d'automatisation spécifique. Il doit entre autres :

- déterminer les conditions pour la mise en route de la CPU (par exemple, initialiser des signaux),
- traiter des données du processus (par exemple, combiner des signaux binaires, lire et exploiter des valeurs analogiques, définir des signaux binaires pour la sortie, écrire des valeurs analogiques),
- réagir aux alarmes,
- traiter les perturbations dans le déroulement normal du programme.

2.2 Blocs dans le programme utilisateur

Le logiciel de programmation STEP 7 Lite vous permet de structurer votre programme utilisateur, c'est-à-dire de le diviser en différentes parties autonomes. Ceci offre les avantages suivants :

- écrire des programmes importants mais clairs,
- standardiser certaines parties du programme,
- simplifier l'organisation du programme,
- modifier facilement le programme,
- simplifier le test du programme, car vous pouvez l'exécuter section par section,
- faciliter la mise en service.

Types de bloc

Vous pouvez utiliser différents types de bloc dans un programme utilisateur :

Bloc	Brève description de la fonction	Pour plus de détails, voir
Blocs d'organisation (OB)	Les OB déterminent la structure du programme utilisateur.	Blocs d'organisation et structure du programme
Blocs fonctionnels système (SFB) et fonctions système (SFC)	Les SFB et SFC sont intégrés à la CPU S7 et vous permettent de réaliser quelques fonctions systèmes importantes.	Blocs fonctionnels système (SFB) et fonctions système (SFC)
Blocs fonctionnels (FB)	Les FB sont des blocs avec "mémoire" que vous programmez vous-même.	Blocs fonctionnels (FB)
Fonctions (FC)	Les FC contiennent des sous-programmes pour les fonctions fréquemment utilisées.	Fonctions (FC)
Blocs de données d'instance (DB d'instance)	Les DB d'instance sont associés au bloc FB/SFB lors de son appel. Ils sont générés automatiquement lors de la compilation.	Blocs de données d'instance
Blocs de données (DB)	Les DB sont des zones de données pour enregistrer les données utilisateur. En plus données associées à chaque bloc fonctionnel, vous pouvez définir des données globales utilisables par tous les blocs.	Blocs de données globaux (DB)

Les OB, FB, SFB, FC et SFC contiennent des parties de programme et sont de ce fait également désignés comme blocs de code. Le nombre de blocs autorisés par type de bloc ainsi que la longueur maximale de chaque bloc dépendent de la CPU.

2.2.1 Blocs d'organisation et structure du programme

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent le traitement de programme cyclique et déclenché par alarme, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs. Vous pouvez programmer les blocs d'organisation et déterminer ainsi le comportement de la CPU.

Priorité des blocs d'organisation

Les blocs d'organisation définissent l'ordre (événements déclencheurs) dans lequel les différentes parties du programme sont traitées. L'exécution d'un OB peut être interrompue par l'appel d'un autre OB. Cette interruption se fait selon la priorité : les OB de priorité plus élevée interrompent les OB de priorité plus faible. La priorité la plus faible est celle de l'OB d'arrière-plan.

Types d'alarme et classes de priorité

On appelle alarmes les événements qui déclenchent l'appel d'un OB donné. Le tableau suivant présente les types d'alarme pour STEP 7 Lite et la priorité des blocs d'organisation associés. Tous les blocs d'organisation indiqués et toutes leurs classes de priorité ne sont pas contenus dans toutes les CPU S7 (voir le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*).

Type d'alarme	Bloc d'organisation	Classe de priorité (prédéfinie)	Pour plus de détails, voir
Cycle libre	OB1	1	Bloc d'organisation pour l'exécution cyclique du programme (OB1)
Alarmes horaires	OB10 à OB17	2	Blocs d'organisation pour l'alarme horaire (OB10 à OB17)
Alarmes temporisées	OB20	3	Blocs d'organisation pour l'alarme temporisée (OB20 à OB23)
	OB21	4	
	OB22	5	
	OB23	6	
Alarmes cycliques	OB30	7	Blocs d'organisation pour l'alarme cyclique (OB30 à OB38)
	OB31	8	
	OB32	9	
	OB33	10	
	OB34	11	
	OB35	12	
	OB36	13	
	OB37	14	
	OB38	15	
Alarmes de processus	OB40	16	Blocs d'organisation pour l'alarme de processus (OB40 à OB47)
	OB41	17	
	OB42	18	
	OB43	19	
	OB44	20	
	OB45	21	
	OB46	22	
	OB47	23	

Type d'alarme	Bloc d'organisation	Classe de priorité (prédéfinie)	Pour plus de détails, voir
Erreurs asynchrones	OB80 Erreur de temps OB82 Alarme de diagnostic OB84 Erreur matérielle CPU OB85 Erreur d'exécution du programme OB86 Défaillance d'unité OB87 Erreur de communication	26 (ou 28 si l'OB d'erreur asynchrone figure dans le programme de mise en route)	Blocs d'organisation pour le traitement d'erreurs (OB80 à OB87 / OB121 à OB122)
Mise en route	OB100 Démarrage à chaud OB102 Démarrage à froid	27 27	Blocs d'organisation pour la mise en route (OB100/OB102)
Erreurs synchrones	OB121 Erreur de programmation OB122 Erreur d'accès à la périphérie	Priorité de l'OB à l'origine de l'erreur	Blocs d'organisation pour le traitement d'erreurs (OB80 à OB87 / OB121 à OB122)
1) A la classe de priorité 29 correspond la priorité 0.29. La priorité du cycle en arrière-plan et donc inférieure à celle du cycle libre.			

Modification de la priorité

La priorité des blocs d'organisation n'est pas modifiable.

Les OB d'erreur déclenchés en cas d'erreurs synchrones sont traités dans la même classe de priorité que le bloc en cours d'exécution lors de la détection de l'erreur.

Données locales

Vous pouvez déclarer des données locales temporaires lors de la création de blocs de code (OB, FC, FB). La zone de données locales disponible dans la CPU est partagée entre les différentes classes de priorité.

Informations de déclenchement d'un OB

Chaque bloc d'organisation dispose d'informations de déclenchement de 20 octets de données locales que le système d'exploitation transmet lors du lancement d'un OB. Ces informations précisent l'événement ayant déclenché l'OB, la date et l'heure du déclenchement de l'OB, les erreurs apparues et les événements de diagnostic.

Les informations de déclenchement de l'OB40 d'alarme de processus contiennent, par exemple, l'adresse du module ayant généré l'alarme.

OB d'alarme désactivés

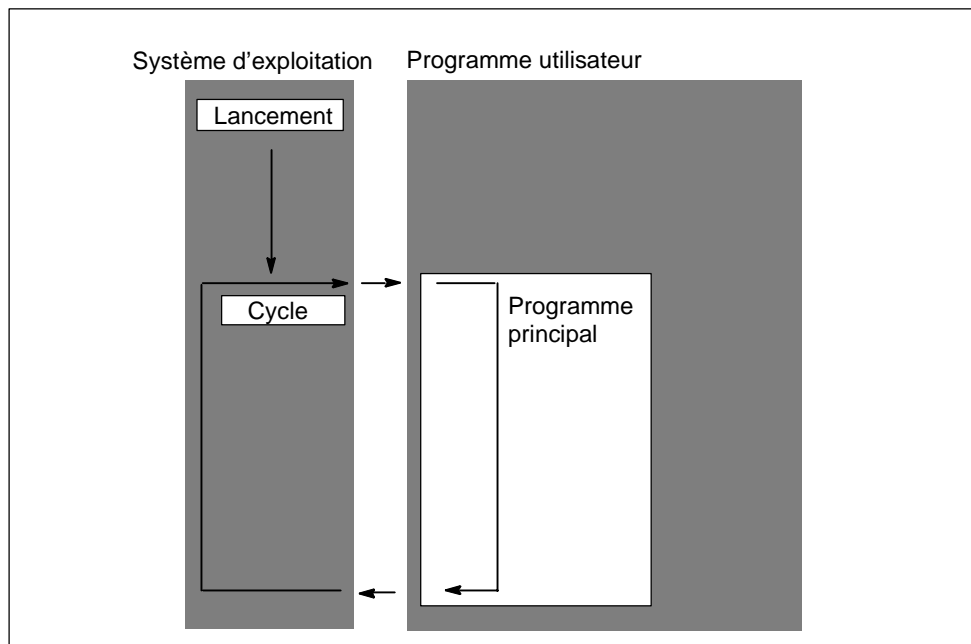
Si vous choisissez la classe de priorité 0 ou affectez moins de 20 octets de données locales à une classe de priorité, l'OB d'alarme correspondant est désactivé. Les OB d'alarme désactivés :

- ne peuvent pas être copiés ni insérés dans le programme utilisateur à l'état de fonctionnement "Marche" (RUN),
- peuvent certes être copiés et insérés dans le programme utilisateur à l'état "Arrêt" (STOP), mais entraînent lors du démarrage à chaud de la CPU l'interruption de la mise en route et génèrent une entrée dans la mémoire tampon de diagnostic.

En désactivant les OB d'alarme inutiles, vous augmenterez la zone de données locales libre disponible qui pourra ainsi servir à enregistrer des données temporaires dans d'autres classes de priorité.

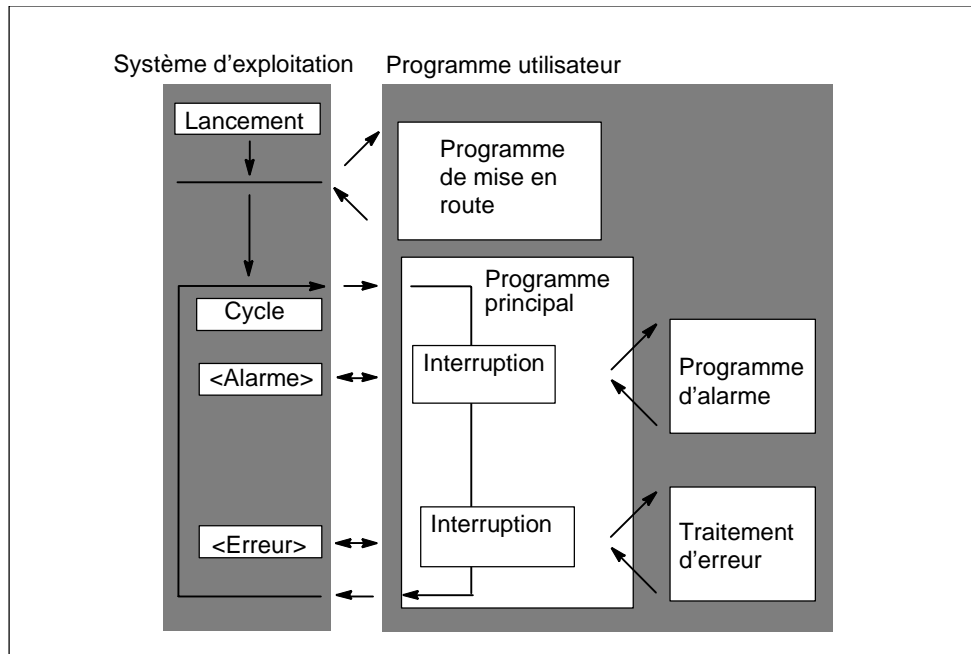
Traitement de programme cyclique

Le traitement de programme cyclique constitue le traitement normal pour les automates programmables. Ceci signifie que le système d'exploitation parcourt une boucle de programme (le cycle) et appelle le bloc d'organisation OB1 dans le programme principal une fois par boucle. Le programme utilisateur dans le bloc OB1 est donc exécuté cycliquement.



Traitement de programme déclenché par événement

Le traitement de programme cyclique peut être interrompu par des événements déclencheurs précis : les alarmes. En présence d'un tel événement, le bloc en cours d'exécution est interrompu à la fin de l'instruction et le bloc d'organisation associé à l'événement déclencheur est traité. Le traitement du programme cyclique reprend ensuite au point d'interruption.

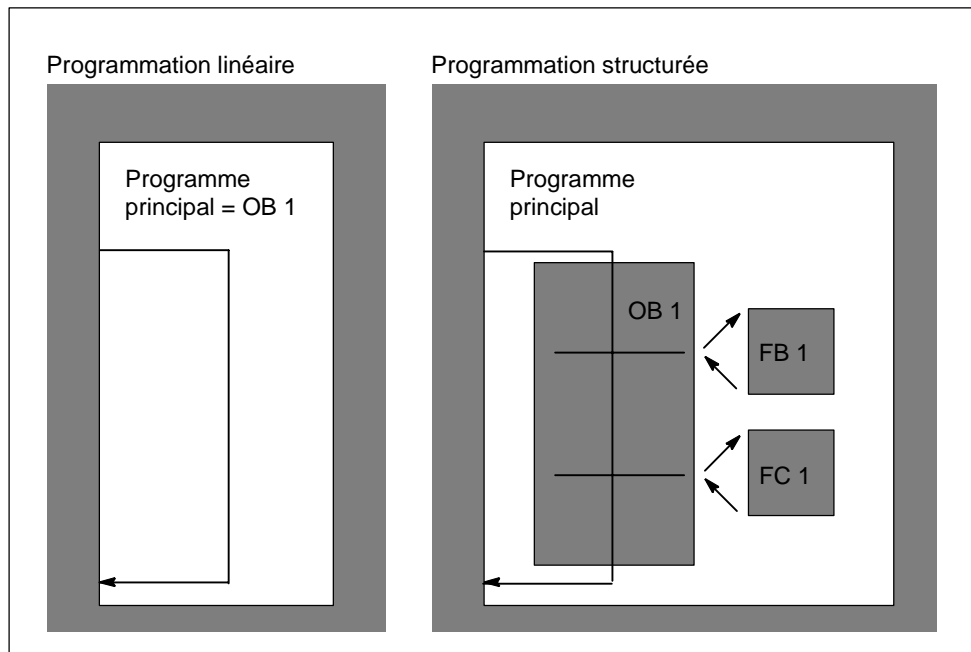


Ceci vous permet de ne traiter qu'en cas de besoin les parties du programme utilisateur qui ne doivent pas l'être cycliquement. Vous pouvez subdiviser votre programme en parties que vous répartissez dans différents blocs d'organisation. Par exemple, pour qu'il réagisse à un signal important qui ne se présente pas souvent (tel un signal de capteur indiquant qu'une cuve est pleine), vous placerez cette partie du programme dans un OB dont l'exécution sera déclenchée en événementiel.

Programmation linéaire ou structurée

Vous pouvez écrire votre programme utilisateur complet dans l'OB1 (programmation linéaire). Cela n'est toutefois recommandé que pour des programmes simples s'exécutant sur des CPU S7-300 avec une mémoire peu importante.

Les automatismes complexes seront mieux traités si vous les subdivisez en parties plus petites qui correspondent aux fonctions technologiques du processus d'automatisation ou qui peuvent être utilisées plusieurs fois. Dans le programme utilisateur, ces tâches partielles sont représentées par des parties de programme correspondantes : les blocs (programmation structurée).



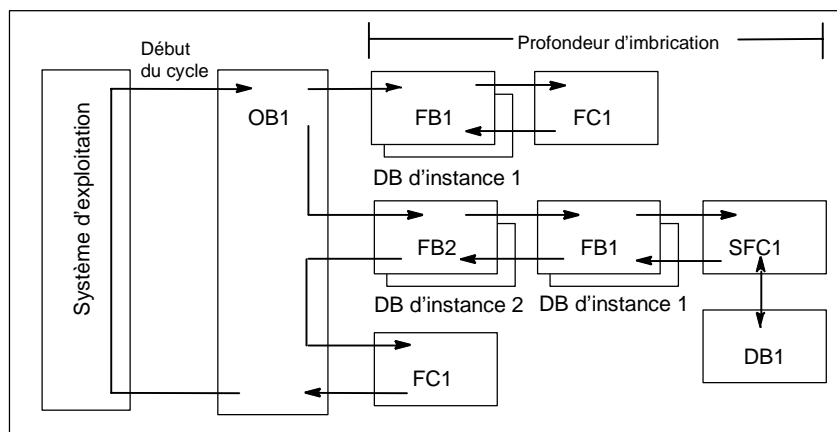
2.2.2 Hiérarchie d'appel dans le programme utilisateur

Pour faire fonctionner le programme utilisateur, vous devez appeler les blocs qui le composent. C'est ce que vous réalisez à l'aide d'opérations STEP 7 Lite spéciales, les appels de bloc que vous ne pouvez programmer et démarrer que dans des blocs de code.

Ordre et profondeur d'imbrication

On appelle hiérarchie d'appel l'ordre et l'imbrication des appels de bloc. Le niveau de profondeur autorisé pour les imbrications dépend de la CPU.

L'exemple de la figure suivante illustre l'ordre et l'imbrication des appels de bloc dans un cycle de traitement.



Règles relatives à l'ordre de création des blocs :

- Vous créez les blocs de haut en bas, ce qui signifie que vous commencez par la rangée de blocs supérieure.
- Tout bloc appelé doit déjà exister, ce qui signifie que dans une rangée de blocs, le sens de création est de droite à gauche.
- En dernier, vous créez l'OB1.

La transposition de ces règles signifie l'ordre de création suivant dans l'exemple considéré :

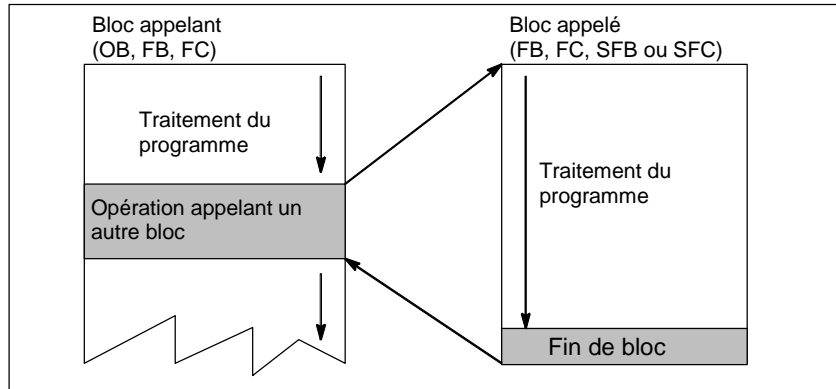
FC1 > FB1 + DB d'instance 1 > DB1 > SFC1 > FB2 + DB d'instance 2 > OB1

Attention

Lorsque la profondeur d'imbrication est trop grande, la pile des données locales risque de déborder (voir aussi Pile des données locales).

Appels de bloc

La figure suivante montre comment s'exécute un appel de bloc au sein d'un programme utilisateur : le programme appelle le deuxième bloc dont les opérations sont alors traitées dans leur intégralité. Une fois le bloc appelé achevé, le traitement se poursuit avec l'opération suivant l'appel de bloc dans le bloc appelant.



Avant de programmer un bloc, vous devez déterminer les données que le programme doit traiter : vous déclarez les variables du bloc.

Nota

Il faut écrire une valeur dans les paramètres OUT à chaque appel de bloc.

Attention

Le système d'exploitation remet à zéro les instances du SFB3 "TP" lors d'un démarrage à froid. Pour initialiser des instances de ce SFB après le démarrage à chaud, vous devez les appeler avec PT = 0 ms dans l'OB100. C'est ce que vous pouvez par exemple obtenir avec une routine d'initialisation dans les blocs contenant des instances de ce SFB.

2.2.3 Traitement de programme cyclique et paramètres de CPU

2.2.3.1 Bloc d'organisation pour l'exécution cyclique du programme (OB1)

L'exécution cyclique du programme constitue le traitement normal pour les automates programmables. Le système d'exploitation appelle l'OB1 cycliquement et déclenche ainsi le traitement cyclique du programme utilisateur.

Déroulement de l'exécution cyclique du programme

Le tableau suivant montre les phases du traitement de programme cyclique :

Phase	Déroulement (dans les CPU à partir de 10/98)
1er 1	Le système d'exploitation démarre le temps de surveillance du cycle.
2e 2	La CPU écrit les valeurs de la mémoire image des sorties dans les modules de sorties.
3e 3	La CPU lit l'état des entrées dans les modules d'entrées et met à jour la mémoire image des entrées.
4e 4	La CPU traite le programme utilisateur et exécute les opérations indiquées dans le programme.
5e 5	A la fin d'un cycle, le système d'exploitation exécute les travaux en attente, par ex. le chargement et l'effacement de blocs, la réception et l'émission de données globales.
6e 6	La CPU revient alors au début du cycle et démarre à nouveau la surveillance du temps de cycle.

Mémoires image du processus

Pour disposer d'une image cohérente des signaux du processus durant l'exécution cyclique du programme, la CPU n'accède pas directement aux modules de signaux quand elle utilise les zones d'opérandes Entrées (E) et Sorties (A), mais à une zone de mémoire interne de la CPU contenant une image des entrées et des sorties.

Programmation de l'exécution cyclique du programme

Pour programmer le traitement cyclique, vous écrivez votre programme utilisateur avec STEP 7 Lite dans l'OB1 et dans les blocs qui y sont appelés.

L'exécution cyclique du programme commence dès que le programme de mise en route s'est achevé sans erreur.

Possibilités d'interruption

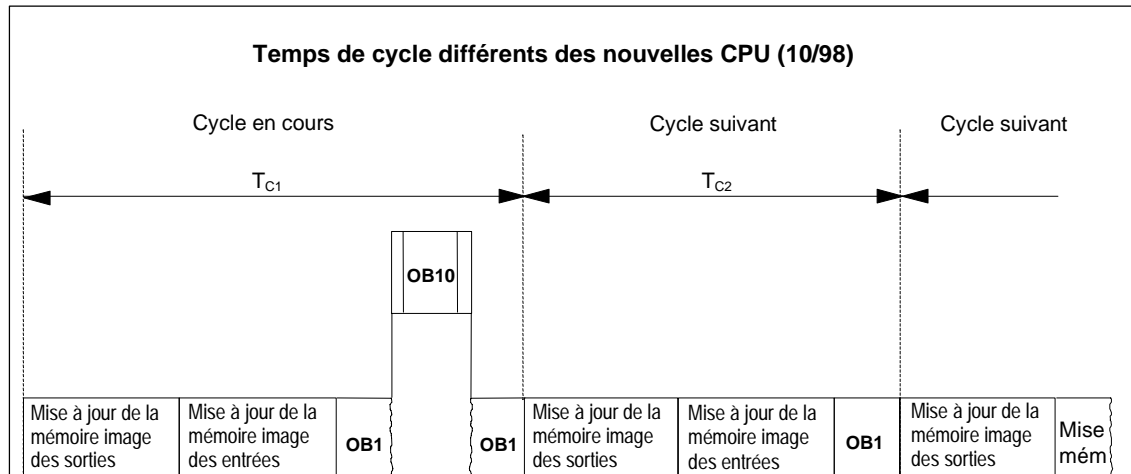
L'exécution cyclique du programme peut être interrompue par :

- une alarme,
- une commande STOP (commutateur de mode, commande de menu depuis la PG, SFC46 STP, SFB20 STOP),
- une coupure de tension secteur,
- l'apparition d'une erreur de matériel ou de programme.

Temps de cycle

Le temps de cycle est le temps dont a besoin le système d'exploitation pour traiter le programme cyclique ainsi que toutes les parties de programme interrompant ce cycle (par ex. exécution d'autres blocs d'organisation) et les activités du système (par ex. mise à jour de la mémoire image). Ce temps est surveillé.

Le temps de cycle (T_c) n'est pas le même à chaque cycle. La figure suivante montre différents temps de cycle pour des CPU, causés par l'insertion d'un OB10 d'alarme horaire (qui interrompt l'OB1).



Temps de surveillance cycle

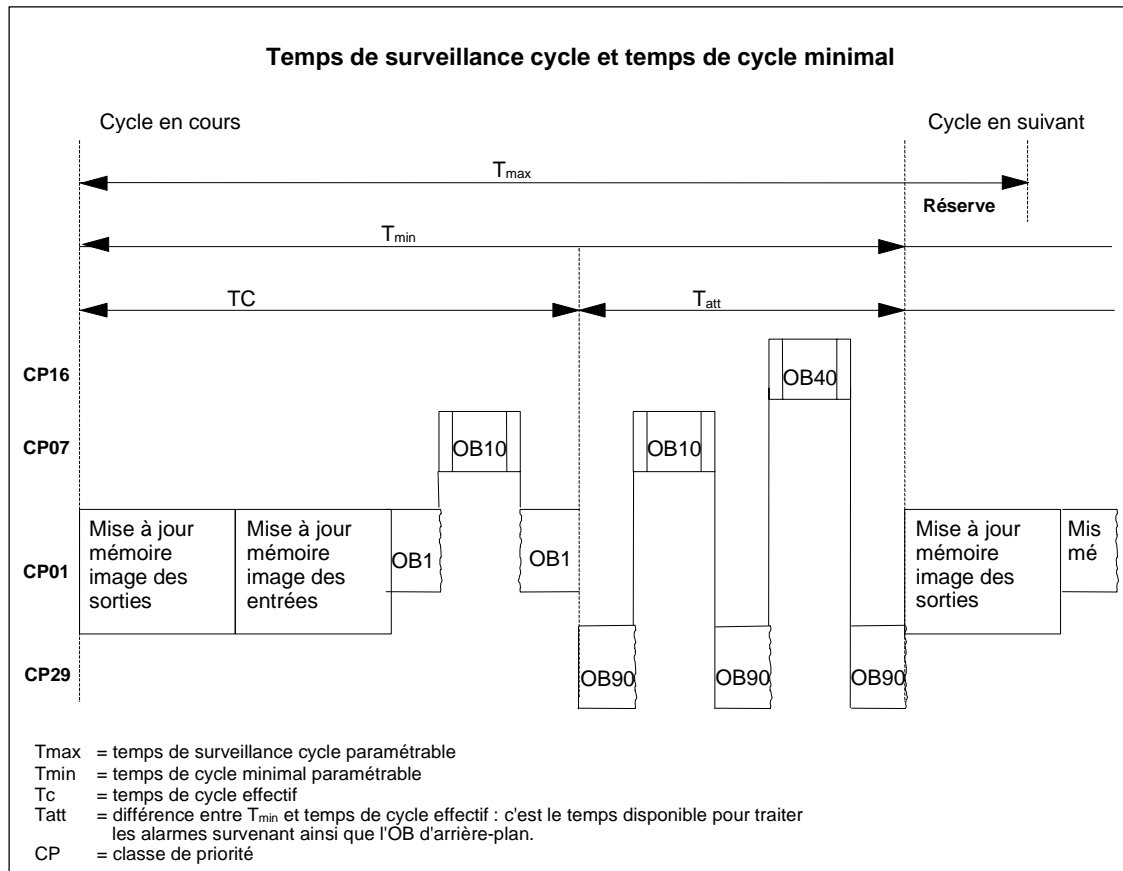
STEP 7 Lite vous permet de modifier le temps de surveillance cycle pris par défaut. L'expiration de ce temps provoque soit le passage de la CPU à l'état de fonctionnement "Arrêt", soit l'appel de l'OB80 dans lequel vous pouvez déterminer comment la CPU doit réagir à l'erreur de temps.

Temps de cycle minimal

STEP 7 Lite vous permet de définir un temps de cycle minimal pour les CPU assumant cette fonction. Ceci est recommandé :

- quand l'intervalle de temps séparant deux exécutions de l'OB1 (cycle libre) doit rester constant,
- pour éviter une actualisation trop fréquente des mémoires image lorsque le temps de cycle est très court.

La figure suivante illustre la fonction du temps de surveillance du cycle et celle du temps de cycle minimal dans le déroulement du programme.



2.2.3.2 Charge due à la communication

Le paramètre de CPU "Charge du cycle due à la communication" vous permet d'influencer dans une certaine mesure la durée des processus de communication, qui allongent toujours le temps de cycle. On entend par processus de communication, par exemple, la transmission de données à une autre CPU via MPI ou le chargement de blocs déclenché au moyen de la PG.

Les fonctions de test avec la PG, qui peuvent allonger considérablement le temps de cycle, ne sont presque pas influencées par ce paramètre. C'est dans le mode processus qu'on peut restreindre le temps disponible pour les fonctions de test.

Comment le paramètre agit-il ?

Le système d'exploitation de la CPU met continuellement à la disposition de la communication le pourcentage configuré de la puissance de traitement totale de la CPU (technique des tranches de temps). Quand cette puissance de traitement n'est pas nécessaire à la communication, elle est disponible pour le reste du traitement.

Effet sur le temps de cycle réel

Sans événements asynchrones supplémentaires, le temps de cycle de l'OB1 s'allonge d'un facteur calculable par la formule suivante :

$$\frac{100}{100 - \text{"Charge du cycle due à la communication (\%)\"}}$$

Exemple 1 (pas d'événements asynchrones en plus)

Une charge du cycle par la communication fixée à 50 % peut doubler le temps de cycle de l'OB1.

En même temps, le temps de cycle de l'OB1 est influencé aussi par des événements asynchrones (tels qu'alarmes de processus ou alarmes cycliques). Le temps de cycle étant allongé par la partie réservée à la communication, il se produira – statistiquement parlant – plus d'événements asynchrones dans un cycle d'OB1, ce qui l'allonge encore. Cet allongement dépend du nombre d'événements survenant par cycle d'OB1 et de la durée de traitement d'un événement.

Exemple 2 (compte tenu d'événements asynchrones supplémentaires)

Une durée d'exécution de l'OB1 de 500 ms avec une charge de communication de 50% peut donner un temps de cycle réel allant jusqu'à 1000 ms (à condition que la CPU ait toujours assez de tâches de communication à traiter). Une alarme cyclique intervenant toutes les 100 ms avec une durée d'exécution de 20 ms allongerait de $5 \times 20 \text{ ms} = 100 \text{ ms}$ au total un cycle sans charge de communication, ce qui donnerait un temps de cycle réel de 600 ms. Puisqu'une alarme cyclique interrompt aussi la communication, elle allongera le temps de cycle de $10 \times 20 \text{ ms}$ avec une charge de communication de 50%, c'est-à-dire que le temps de cycle réel sera dans ce cas non pas de 1000 ms mais de 1200 ms.

Nota

- Lorsque vous modifiez la valeur attribuée au paramètre "Charge du cycle due à la communication", vérifiez-en les effets sur le fonctionnement de l'installation.
- Tenez compte de la charge due à la communication lorsque vous fixez le temps de cycle minimal, pour éviter les erreurs de temps.

Recommandations

- Autant que possible, adoptez la valeur par défaut.
- Augmentez cette valeur seulement si la CPU est employée surtout à des fins de communication et que le programme utilisateur n'est pas à durée critique.
- Dans tous les autres cas, bornez-vous à réduire cette valeur !
- Passez en mode processus et limitez le temps requis à cet endroit pour les fonctions de test.

2.2.4 Traitement de programme déclenché par alarme

2.2.4.1 Blocs d'organisation pour le traitement de programme déclenché par alarme

En mettant à votre disposition des OB d'alarme, les CPU S7 vous donnent la possibilité :

- de déclencher le traitement de parties de programme par horloge,
- de réagir de manière optimale aux signaux externes du processus.

Le programme utilisateur cyclique ne doit pas tester constamment si des événements d'alarme sont apparus. En effet, en cas d'alarme, le système d'exploitation fait en sorte que soit exécutée la partie du programme utilisateur figurant dans l'OB d'alarme et qui détermine comment l'automate programmable doit réagir à cette alarme.

Types d'alarme et applications

Le tableau suivant montre comment utiliser les types d'alarme.

Type d'alarme	OB d'alarme	Exemples d'application
Alarme horaire	OB10 à OB17	Calcul du débit d'un processus de mélange à la fin de la journée de travail
Alarme temporisée	OB20 à OB23	Commande d'un ventilateur devant fonctionner encore 20 s après l'arrêt d'un moteur avant d'être lui-même arrêté
Alarme cyclique	OB30 à OB38	Echantillonnage d'un niveau de signal pour une installation de régulation
Alarme de processus	OB40 à OB47	Signaler que le niveau maximal d'une cuve est atteint

2.2.4.2 Blocs d'organisation pour l'alarme horaire (OB10 à OB17)

Les CPU S7 mettent à votre disposition des OB d'alarme horaire pouvant être traités à une date donnée ou à des intervalles de temps définis.

Les alarmes horaires peuvent être déclenchées :

- une seule fois à un moment donné (indication de temps absolue avec date),
- périodiquement avec indication du début et de la fréquence de répétition (par exemple, toutes les minutes, toutes les heures, tous les jours).

Règles d'utilisation des alarmes horaires

Les alarmes horaires ne peuvent être traitées que si une alarme horaire a été paramétrée et qu'un bloc d'organisation correspondant est contenu dans le programme utilisateur. Si ce n'est pas le cas, un message d'erreur s'inscrit dans la mémoire tampon de diagnostic et un traitement d'erreur asynchrone est exécuté (OB80, voir "Blocs d'organisation pour le traitement d'erreur (OB80 à OB87 / OB121 à OB122)").

Les alarmes horaires périodiques doivent correspondre à une date réelle. Il n'est, par exemple, pas possible de réitérer chaque mois un OB10 ayant le 31 janvier comme point de départ. Dans ce cas, l'OB ne serait déclenché que pour les mois ayant 31 jours.

Une alarme horaire activée au cours de la mise en route (démarrage à chaud ou redémarrage) n'est traitée qu'une fois la mise en route achevée.

Il n'est pas possible de déclencher les OB d'alarme horaire désactivés par paramétrage. La CPU détecte dans ce cas une erreur de programmation et passe à l'état de fonctionnement "Arrêt" (STOP).

Après un démarrage à chaud, il faut activer à nouveau les alarmes horaires réglées, par exemple à l'aide de la SFC30 ACT_TINT dans le programme de mise en route.

Cas particulier : l'instant de déclenchement unique paramétré se trouve dans le passé

Supposons le paramétrage suivant :

- Exécution : unique
- Active : oui
- Date/heure de déclenchement : se trouve dans le passé (par rapport à l'horloge temps réel de la CPU)

Comportement de la CPU : après un démarrage à froid ou à chaud, le système d'exploitation **appelle une fois** l'OB d'alarme horaire correspondant !

Déclenchement de l'alarme horaire

La CPU ne peut déclencher une alarme horaire qu'une fois que vous avez réglé puis activé cette dernière. Il y a trois façons de la déclencher :

- déclenchement automatique de l'alarme horaire par paramétrage avec STEP 7 Lite (bloc de paramètres "Alarmes horaires"),
- réglage et activation de l'alarme horaire à l'aide de SFC28 SET_TINT et de SFC30 ACT_TINT depuis le programme utilisateur,
- réglage de l'alarme horaire par paramétrage avec STEP 7 Lite et activation de l'alarme horaire à l'aide de SFC30 ACT_TINT depuis le programme utilisateur.

Interrogation de l'alarme horaire

Pour savoir si des alarmes horaires ont été réglées et à quel moment, vous pouvez :

- appeler la SFC31 QRY_TINT
- ou demander la liste partielle "Etat des alarmes" de la liste d'état système.

Désactivation de l'alarme horaire

Vous pouvez désactiver des alarmes horaires non encore traitées à l'aide de la SFC29 CAN_TINT. La SFC28 SET_TINT permet de régler à nouveau des alarmes horaires désactivées et la SFC30 ACT_TINT de les activer.

Priorité des OB d'alarme horaire

Les huit OB d'alarme horaire sont prédéfinis avec la même classe de priorité (2) et sont donc traités dans l'ordre d'apparition de leurs événements déclencheurs. Il est possible de modifier la classe de priorité par paramétrage.

Changement de l'heure réglée

Il est possible de modifier l'heure réglée comme suit :

- Un maître d'heure synchronise l'heure pour le maître et les esclaves.
- L'heure est redéfinie dans le programme utilisateur à l'aide de la SFC0 SET_CLK.

Comportement en cas de changement d'heure

Le tableau suivant montre comment les alarmes horaires se comportent après modification de l'heure.

Si...	...alors
une ou plusieurs alarmes horaires ont été sautées par suite de l'avancement de l'horloge,	l'OB80 est démarré avec inscription dans ses informations de déclenchement des alarmes horaires sautées.
vous avez désactivé dans l'OB80 les alarmes horaires sautées,	les alarmes horaires sautées ne sont pas rattrapées.
vous n'avez pas désactivé dans l'OB80 les alarmes horaires sautées,	la première alarme horaire sautée est rattrapée, mais il n'est pas tenu compte des suivantes.
des alarmes horaires déjà traitées se présentent à nouveau par suite du retardement de l'horloge,	ces alarmes horaires ne sont pas traitées une nouvelle fois.

2.2.4.3 Blocs d'organisation pour l'alarme temporisée (OB20 à OB23)

Les CPU S7 mettent à votre disposition des OB d'alarme temporisée grâce auxquels vous pouvez programmer l'exécution retardée de certaines parties de votre programme utilisateur.

Règles d'utilisation des alarmes temporisées

Les alarmes temporisées ne peuvent être traitées que si un bloc d'organisation correspondant est contenu dans le programme CPU. Si tel n'est pas le cas, un message d'erreur est inscrit dans la mémoire tampon de diagnostic et un traitement d'erreur asynchrone est réalisé (OB80, voir Blocs d'organisation pour le traitement d'erreurs (OB80 à OB87 / OB121 à OB122)).

Il n'est pas possible de déclencher les OB d'alarme temporisée désactivés par paramétrage. La CPU détecte dans ce cas une erreur de programmation et passe à l'état de fonctionnement "Arrêt" (STOP).

Les alarmes temporisées sont déclenchées lorsque le temps de retard précisé dans la SFC32 SRT_DINT a expiré.

Déclenchement de l'alarme temporisée

Pour démarrer une alarme temporisée, vous devez fixer dans la SFC32 le temps de retard à l'expiration duquel l'OB d'alarme temporisée correspondant doit être appelé. La durée maximale autorisée pour le temps de retard est donnée dans le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*.

Priorité des OB d'alarme temporisée

Par défaut, les OB d'alarme temporisée ont les classes de priorité 3 à 6. Vous pouvez les modifier par paramétrage.

2.2.4.4 Blocs d'organisation pour l'alarme cyclique (OB30 à OB38)

Les CPU S7 mettent à votre disposition des OB d'alarme cyclique qui interrompent le traitement de programme cyclique à intervalles précis.

Les alarmes cycliques sont déclenchées à des intervalles de temps précis. La période débute au passage de l'état de fonctionnement "Arrêt" (STOP) à l'état "Marche" (RUN).

Règles d'utilisation des alarmes cycliques

En choisissant la période, veillez à ce qu'il reste assez de temps pour le traitement des alarmes cycliques entre les événements déclencheurs des différentes alarmes cycliques.

Il n'est pas possible de déclencher les OB d'alarme cyclique désactivés par paramétrage. La CPU détecte dans ce cas une erreur de programmation et passe à l'état de fonctionnement "Arrêt" (STOP).

Déclenchement de l'alarme cyclique

Pour déclencher une alarme cyclique, vous devez préciser une période via STEP 7 Lite dans le bloc de paramètres "Alarmes cycliques". Cette période est toujours un multiple entier de la période de base de 1 ms.

Période = $n \times$ période de base 1 ms

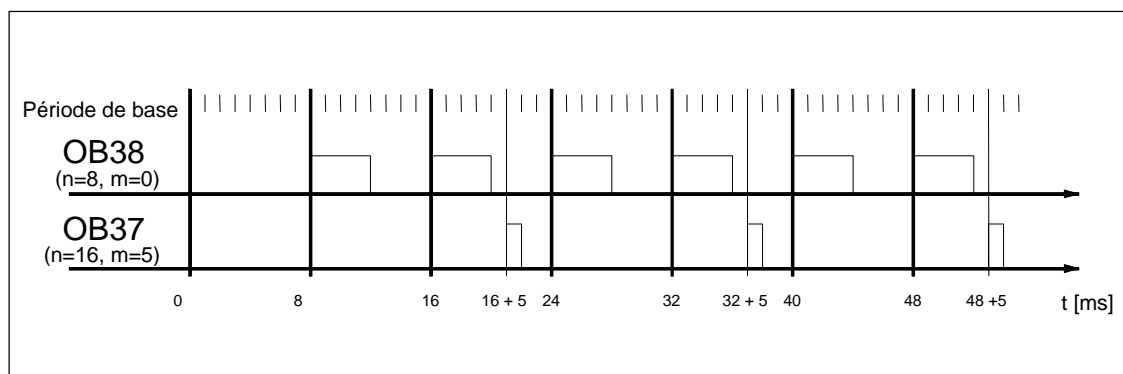
Les neuf OB d'alarme cyclique disponibles ont des périodes prédéfinies (voir tableau suivant). La période par défaut entre en vigueur quand l'OB d'alarme cyclique qui lui est associé est chargé. Vous pouvez toutefois modifier par paramétrage les valeurs prédéfinies. La limite supérieure est donnée dans le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*.

Décalage de phase pour les alarmes cycliques

Afin d'éviter que les alarmes cycliques de différents OB d'alarme cyclique reçoivent une demande de déclenchement au même moment, provoquant ainsi éventuellement une erreur de temps (dépassement du temps de cycle), vous pouvez préciser un décalage de phase. Ce décalage de phase assure que le traitement d'une alarme cyclique sera décalé d'une certaine durée après écoulement de la période.

Décalage de phase = $m \times$ période de base ($0 \leq m < n$)

La figure suivante montre le traitement d'un OB d'alarme cyclique avec décalage de phase (OB37) comparé à celui d'une alarme cyclique sans décalage de phase (OB38).



Priorité des OB d'alarme cyclique

Le tableau suivant montre les périodes et les classes de priorité par défaut des OB d'alarme cyclique. Vous pouvez les modifier par paramétrage.

OB d'alarme cyclique	Période en ms	Classe de priorité
OB30	5000	7
OB31	2000	8
OB32	1000	9
OB33	500	10
OB34	200	11
OB35	100	12
OB36	50	13
OB37	20	14
OB38	10	15

2.2.4.5 Blocs d'organisation pour l'alarme de processus (OB40 à OB47)

Les CPU S7 mettent à votre disposition des OB d'alarme de processus qui réagissent à des signaux provenant des modules (modules de signaux SM, processeurs de communication CP, modules de fonction FM). STEP 7 Lite vous permet de définir quel signal doit déclencher l'OB pour les modules TOR et analogiques paramétrables. Pour les CP et les FM, vous utiliserez à cet effet les dialogues de paramétrage correspondants.

Les alarmes de processus sont déclenchées lorsqu'un module de signaux pouvant générer des alarmes de processus, avec validation d'alarme de processus paramétrée, transmet un signal de processus reçu à la CPU ou lorsqu'un module de fonction de la CPU signale une alarme.

Règles d'utilisation d'alarmes du processus

Les alarmes de processus ne peuvent être traitées que si un bloc d'organisation correspondant est contenu dans le programme CPU. Si tel n'est pas le cas, un message d'erreur est inscrit dans la mémoire tampon de diagnostic et un traitement d'erreur asynchrone est réalisé (voir Blocs d'organisation pour le traitement d'erreurs (OB80 à OB87 / OB121 à OB122)).

Il n'est pas possible de déclencher les OB d'alarme de processus désactivés par paramétrage. La CPU détecte dans ce cas une erreur de programmation et passe à l'état de fonctionnement "Arrêt" (STOP).

Paramétrage de modules de signaux pouvant générer des alarmes de processus

Chaque voie d'un module de signaux pouvant générer des alarmes de processus peut déclencher une alarme de processus. Aussi devez-vous définir à l'aide de STEP 7 Lite dans les jeux de paramètres de ces modules :

- ce qui doit déclencher une alarme de processus,
- quel OB d'alarme de processus doit être traité (l'OB40 est prévu par défaut pour le traitement de toutes les alarmes de processus).

Vous activez avec STEP 7 Lite la génération d'alarmes de processus des modules de fonction. Vous affectez d'autres paramètres dans les dialogues de paramétrage de ces modules.

Priorité des OB d'alarme de processus

Par défaut, les OB d'alarme de processus ont les classes de priorité 16 à 23. Vous pouvez les modifier par paramétrage.

2.2.4.6 Blocs d'organisation pour la mise en route (OB100/OB102)

Modes de mise en route

On distingue entre les modes de mise en route suivants :

- démarrage à chaud,
- démarrage à froid.

Le tableau suivant indique l'OB respectivement appelé par le système d'exploitation.

Mode de mise en route	OB associé
Démarrage à chaud	OB100
Démarrage à froid	OB102

Evénements déclencheurs des OB de mise en route

La CPU exécute une mise en route

- après mise sous tension,
- lorsque vous actionnez le commutateur de mode de fonctionnement de STOP à RUN/RUN-P,
- après sollicitation par une fonction de communication,
- après synchronisation en mode multiprocesseur,
- dans un système H, après couplage (uniquement sur la CPU de réserve).

L'OB de mise en route correspondant (OB100 ou OB102) est appelé selon l'événement déclencheur, la CPU mise en œuvre ainsi que les paramètres sélectionnés pour cette dernière.

Programme de mise en route

Vous pouvez déterminer les conditions supplémentaires pour le comportement de mise en route de votre CPU (valeurs d'initialisation pour "Marche", valeurs de mise en route pour les modules de périphérie) en écrivant votre programme de mise en route dans les blocs d'organisation OB100 pour le démarrage à chaud ou OB102 pour le démarrage à froid.

La longueur du programme de mise en route est indifférente : son exécution n'est pas limitée en durée, car la surveillance du temps de cycle n'est pas active. L'exécution commandée par horloge ou par alarme n'est pas possible dans le programme de mise en route. Pendant cette dernière, toutes les sorties TOR prennent l'état de signal 0.

Mode de mise en route après une mise en route manuelle

Pour les CPU S7-300, seuls les démarrages manuel (démarrages à chaud) ou à froid (uniquement CPU 318-2) sont possibles.

Mode de mise en route après une mise en route automatique

Seul le démarrage à chaud est possible après la mise sous tension pour les CPU S7-300.

Contrôle des modules : configuration prévue / configuration sur site

Vous pouvez demander par paramétrage que soit vérifié, avant la mise en route, si tous les modules figurant dans la table de configuration sont réellement enfichés et si leur type est correct.

Lorsque le contrôle des modules est activé, la mise en route n'est pas exécutée si une différence entre la configuration prévue et la configuration réelle est mise en évidence.

Temps de surveillance

Vous pouvez paramétrer les temps de surveillance suivants pour garantir une mise en route sans erreur de l'automate programmable :

- le temps maximal autorisé pour la transmission des paramètres aux modules,
- le temps maximal autorisé pour le message Prêt des modules après la mise sous tension,

La CPU passe à l'état "Arrêt" à l'expiration des temps de surveillance ou seul un démarrage est alors possible.

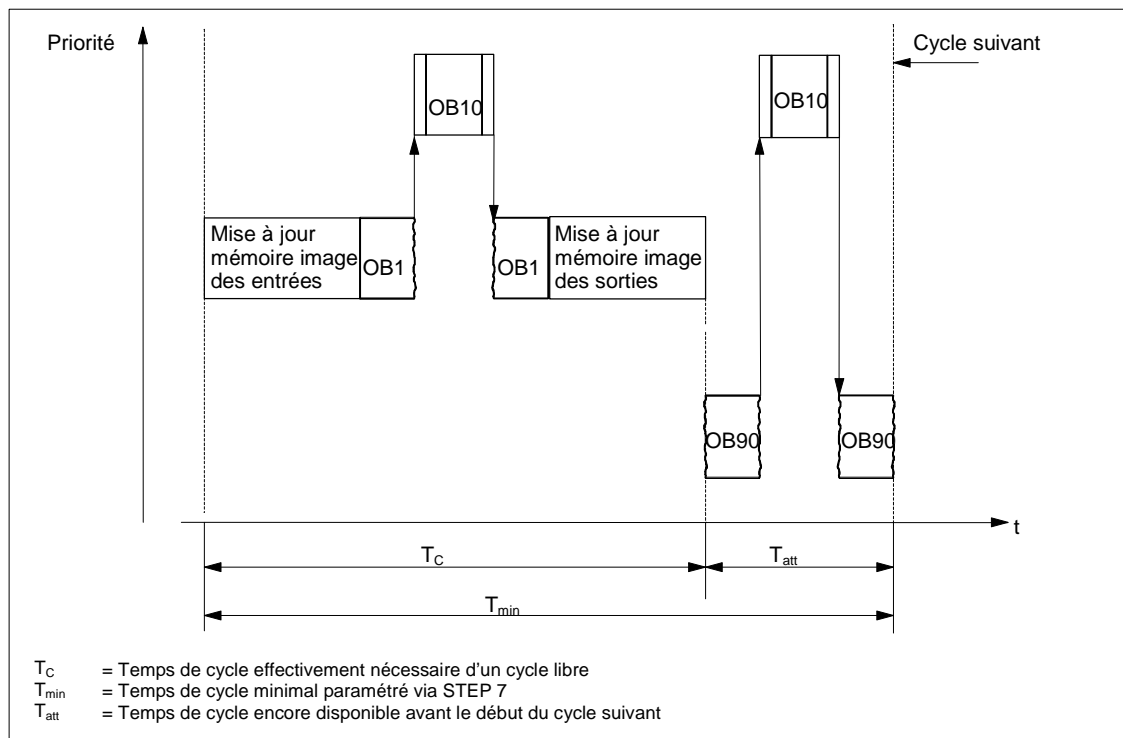
2.2.4.7 Bloc d'organisation pour l'exécution du programme en arrière-plan (OB90)

Si vous avez paramétré un temps de cycle minimal avec STEP 7 Lite et que celui-ci s'avère supérieur au temps de cycle effectif, la CPU dispose de temps d'exécution à la fin du programme cyclique. Ce temps restant sert à l'exécution de l'OB d'arrière-plan. Si l'OB90 n'existe pas dans votre CPU, cette dernière attend que le temps de cycle minimal paramétré soit expiré. L'OB90 vous permet donc d'exécuter des processus à durée non critique et ainsi d'éviter des temps d'attente.

Priorité de l'OB d'arrière-plan

L'OB d'arrière-plan a la classe de priorité 29 qui correspond à la priorité 0.29. C'est donc l'OB à la priorité la plus faible et vous ne pouvez pas modifier sa classe de priorité par paramétrage.

La figure suivante montre un exemple d'exécution du cycle en arrière-plan, du cycle libre et de l'OB10 (pour les anciennes CPU).



Programmation de l'OB90

Le système d'exploitation de la CPU ne contrôle pas le temps d'exécution de l'OB90 ; vous pouvez donc y programmer des boucles de longueur indifférente. Veillez à la cohérence des données que vous utilisez dans le programme en arrière-plan en tenant compte des points suivants lors de leur programmation :

- les événements de remise à zéro de l'OB90 (voir manuel de référence *Logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système*),
- la mise à jour de la mémoire image du processus qui est asynchrone par rapport à l'OB90.

2.2.4.8 Blocs d'organisation pour le traitement d'erreurs (OB80 à OB87 / OB121 à OB122)

Types d'erreur

Les erreurs que les CPU S7 détectent et auxquelles elles peuvent réagir à l'aide de blocs d'organisation sont classables en deux catégories :

- Erreurs synchrones : ces erreurs peuvent être associées à une partie précise du programme utilisateur. L'erreur apparaît pendant le traitement d'une opération précise. Si l'OB d'erreur synchrone correspondant n'est pas chargé, la CPU passe à l'état "Arrêt" (STOP) à l'apparition d'une telle erreur.
- Erreurs asynchrones : ces erreurs ne peuvent pas être directement associées au programme utilisateur traité. Il s'agit d'erreurs de classe de priorité, d'erreurs dans l'automate programmable (par exemple, module défaillant) ou d'erreurs de redondance. Si l'OB d'erreur asynchrone correspondant n'est pas chargé, la CPU passe à l'état "Arrêt" (STOP) à l'apparition d'une telle erreur.

Le tableau ci-après montre les types d'erreur pouvant survenir, classés selon la catégorie des OB d'erreur.

Erreurs asynchrones / erreurs de redondance	Erreurs synchrones
OB80 Erreur de temps (ex. : dépassement du temps de cycle)	OB121 Erreur de programmation (ex. : DB non chargé)
OB82 Alarme de diagnostic (ex. : court-circuit dans le module d'entrées)	OB 122 Erreur d'accès à la périphérie (ex. : accès à un module de signaux qui n'existe pas)
OB84 Erreur matérielle CPU (erreur à l'interface avec le réseau MPI)	
OB85 Erreur d'exécution du programme (ex. : OB non chargé)	
OB86 Défaillance d'unité	
OB87 Erreur de communication (ex. : mauvaise ID de télégramme pour communication par données globales)	

Utilisation des OB pour erreurs synchrones

Les erreurs synchrones apparaissent pendant le traitement d'une opération précise. A l'apparition de ces erreurs, le système d'exploitation génère une entrée dans la pile des interruptions et déclenche l'OB pour erreurs synchrones.

Les OB d'erreur appelés par des erreurs synchrones sont traités en tant que partie du programme avec la même classe de priorité que le bloc en cours d'exécution lors de la détection de l'erreur. Les OB121 et OB122 peuvent donc accéder aux valeurs enregistrées dans les accumulateurs et les autres registres au moment de l'interruption. Vous pouvez vous servir de ces valeurs pour réagir à l'erreur, puis reprendre l'exécution de votre programme (par exemple, indiquer une valeur de remplacement dans l'OB122 via la SFC44 RPL_VAL pour une erreur d'accès à un module d'entrées analogiques). Ainsi, toutefois, la pile L de cette classe de priorité doit en plus prendre en charge les données locales des OB d'erreur.

Utilisation des OB pour erreurs asynchrones

Lorsque le système d'exploitation de la CPU détecte une erreur asynchrone, il déclenche l'OB d'erreur correspondant (OB80 à OB87). Les OB pour erreurs asynchrones ont la priorité la plus élevée : ils ne peuvent pas être interrompus par d'autres OB, lorsque tous les OB d'erreurs synchrones ont la même priorité. Si plusieurs erreurs asynchrones de la même priorité apparaissent simultanément, les OB d'erreur correspondants sont traités dans l'ordre d'apparition des erreurs.

Masquage d'événements déclencheur

Des fonctions système vous permettent de masquer, d'ajourner ou d'inhiber les événements déclencheurs de quelques OB d'erreur. De plus amples informations à ce sujet et sur les blocs d'organisation en particulier sont données dans le manuel de référence *Logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système*.

Type de l'OB d'erreur	SFC	Fonction de la SFC
OB d'erreur synchrone	SFC36 MSK_FLT	Masquer des événements d'erreur synchrone individuels. Les événements d'erreur masqués ne déclenchent aucun OB d'erreur et n'entraînent pas la réaction programmée.
	SFC37 DMSK_FLT	Démasquer des événements d'erreur synchrone
OB d'erreur asynchrone	SFC39 DIS_IRT	Inhiber globalement des événements d'erreur asynchrone et d'alarme. Les événements d'erreur inhibés ne déclenchent d'OB d'erreur dans aucun des cycles de CPU suivants et n'entraînent pas la réaction programmée.
	SFC40 EN_IRT	Valider des événements d'erreur asynchrone et d'alarme
	SFC41 DIS_AIRT	Ajourner les événements d'erreur asynchrone et d'alarme prioritaires jusqu'à la fin de l'OB
	SFC42 EN_AIRT	Valider les événements d'erreur asynchrone et d'alarme prioritaires

Nota

Pour ignorer des alarmes, il est plus efficace de les inhiber au moyen de SFC à la mise en route que de charger un OB vide (contenant BE).

2.2.5 Catégorie de blocs pour le traitement de programme structuré

2.2.5.1 Fonctions (FC)

Les fonctions font partie des blocs que vous programmez vous-même. Une fonction est un bloc de code sans mémoire. Les variables temporaires d'une fonction sont enregistrées dans la pile des données locales. Ces données sont perdues à l'achèvement de la fonction. Les fonctions peuvent faire appel à des blocs de données globaux pour l'enregistrement des données.

Comme une fonction ne dispose pas de mémoire associée, vous devez toujours indiquer des paramètres effectifs pour elle. Vous ne pouvez pas affecter de valeur initiale aux données locales d'une FC.

Domaine d'application

Une fonction contient un programme qui est exécuté quand cette fonction est appelée par un autre bloc de code. Vous pouvez faire appel à des fonctions pour :

- renvoyer une valeur de fonction au bloc appelant (ex. : fonctions mathématiques),
- exécuter une fonction technologique (ex. : commande individuelle avec combinaison binaire).

Affectation de paramètres effectifs aux paramètres formels

Un paramètre formel sert de paramètre générique au paramètre "réel", le paramètre effectif. Les paramètres effectifs remplacent les paramètres formels lors de l'appel d'une FC. Vous devez toujours affecter des paramètres effectifs aux paramètres formels d'une FC (par ex., le paramètre effectif "E3.6" au paramètre formel "Démarrage"). Les paramètres d'entrée, de sortie et d'entrée/sortie utilisés par la FC sont enregistrés comme pointeurs désignant les paramètres effectifs du bloc de code qui a appelé la fonction.

2.2.5.2 Blocs fonctionnels (FB)

Les blocs fonctionnels font partie des blocs que vous programmez vous-même. Un bloc fonctionnel est un bloc avec rémanence. Un bloc de données d'instance lui est associé qui en constitue la mémoire. Les paramètres transmis au FB ainsi que les variables statiques sont enregistrés dans le bloc de données d'instance. Les variables temporaires sont rangées dans la pile des données locales.

Les données enregistrées dans le bloc de données d'instance ne sont pas perdues à l'achèvement du traitement du FB. En revanche, les données enregistrées dans la pile des données locales le sont.

Nota

Afin d'éviter des erreurs d'utilisation de FB, veuillez lire la rubrique Types de données autorisés pour la transmission de paramètres.

Domaine d'application

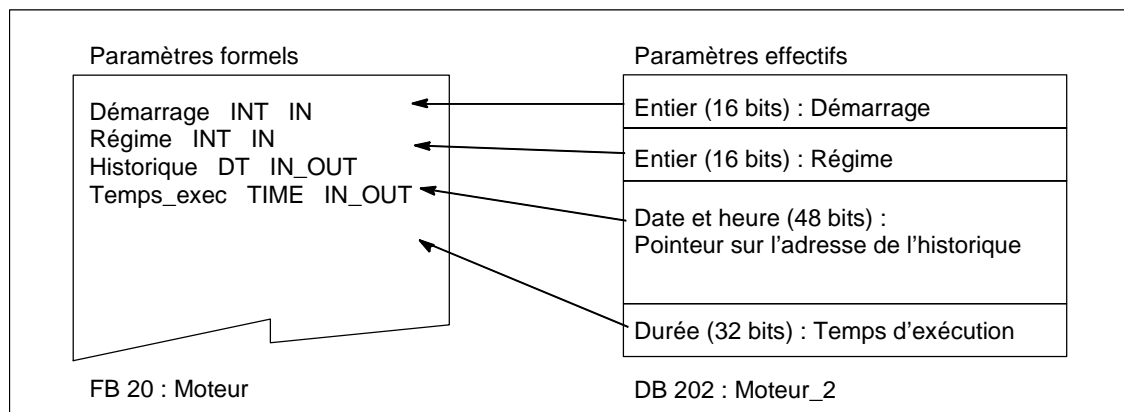
Un bloc fonctionnel contient un programme qui est exécuté quand ce bloc fonctionnel est appelé par un autre bloc de code. Les blocs fonctionnels facilitent la programmation de fonctions complexes souvent utilisées.

FB et DB d'instance

Un bloc de données d'instance est associé à chaque appel de bloc fonctionnel transmettant des paramètres.

En appelant plusieurs instances d'un FB, vous pouvez piloter plusieurs appareils avec un seul bloc fonctionnel. Un FB pour un type de moteur peut, par exemple, commander différents moteurs en utilisant des données d'instance différentes pour les différents moteurs. Il est possible de ranger les données pour chaque moteur (régime, accélération, cumul des temps de fonctionnement, etc.) dans un ou plusieurs DB d'instance.

La figure suivante montre les paramètres formels d'un FB qui utilise les paramètres effectifs enregistrés dans le DB d'instance.



Variables de type de données FB

Si votre programme utilisateur est organisé de telle manière que, dans un FB, soient appelés d'autres blocs fonctionnels existant déjà, vous pouvez déclarer les FB appelés comme variables statiques de type de données FB dans la table de déclaration des variables du FB appelant. Vous obtenez ainsi une imbrication des variables et la concentration des données d'instance dans un bloc de données d'instance (multi-instance).

Affectation de paramètres effectifs aux paramètres formels

En général, il n'est pas nécessaire dans STEP 7 Lite d'affecter des paramètres effectifs aux paramètres formels d'un FB.

Il y a pourtant des exceptions. Il faut affecter des paramètres effectifs :

- pour un paramètre d'entrée/sortie de type de données complexe (par exemple, STRING, ARRAY ou DATE_AND_TIME),
- pour tous les types de paramètre (par exemple, TIMER, COUNTER ou POINTER).

STEP 7 Lite associe les paramètres effectifs aux paramètres formels de la manière suivante :

- Lorsque vous indiquez des paramètres effectifs dans l'instruction d'appel, les opérations du FB utilisent les paramètres effectifs ainsi mis à disposition.
- Lorsque vous n'indiquez pas de paramètres effectifs dans l'instruction d'appel, les opérations du FB utilisent les valeurs contenues dans le DB d'instance.

Le tableau ci-après montre à quelles variables du FB il faut affecter des paramètres effectifs.

Variable	Type de données		
	Type de données simple	Type de données complexe	Type de paramètre
Entrée	Paramètres facultatifs	Paramètres facultatifs	Paramètres effectifs obligatoires
Sortie	Paramètres facultatifs	Paramètres facultatifs	Paramètres effectifs obligatoires
Entrée/sortie	Paramètres facultatifs	Paramètres effectifs obligatoires	–

Affectation de valeurs initiales aux paramètres formels

Vous pouvez préciser des valeurs initiales pour les paramètres formels dans la section de déclaration du FB. Ces valeurs sont reprises dans le bloc de données d'instance associé au FB.

Si vous n'affectez pas de paramètres effectifs aux paramètres formels dans l'instruction d'appel, STEP 7 Lite utilise les valeurs enregistrées dans le DB d'instance. Il peut alors s'agir de valeurs initiales que vous avez saisies dans la table de déclaration des variables du FB.

Le tableau suivant présente les variables pour lesquelles vous pouvez indiquer une valeur initiale. Comme les données temporaires ne sont pas enregistrées après le traitement du bloc, vous ne pouvez pas leur affecter de valeur.

	Type de données		
Variable	Type de données simple	Type de données complexe	Type de paramètre
Entrée	Valeur initiale autorisée	Valeur initiale autorisée	–
Sortie	Valeur initiale autorisée	Valeur initiale autorisée	–
Entrée/sortie	Valeur initiale autorisée	–	–
Statique	Valeur initiale autorisée	Valeur initiale autorisée	–
Temporaire	–	–	–

2.2.5.3 Blocs de données d'instance

Un bloc de données d'instance est associé à chaque appel de bloc fonctionnel transmettant des paramètres. Ce bloc de données d'instance contient les paramètres effectifs et les données statiques du FB. Les variables déclarées dans le FB déterminent la structure du bloc de données d'instance. On appelle instance l'appel d'un bloc fonctionnel. Si, par exemple, un bloc fonctionnel est appelé cinq fois dans le programme utilisateur S7, il existe cinq instances de ce bloc.

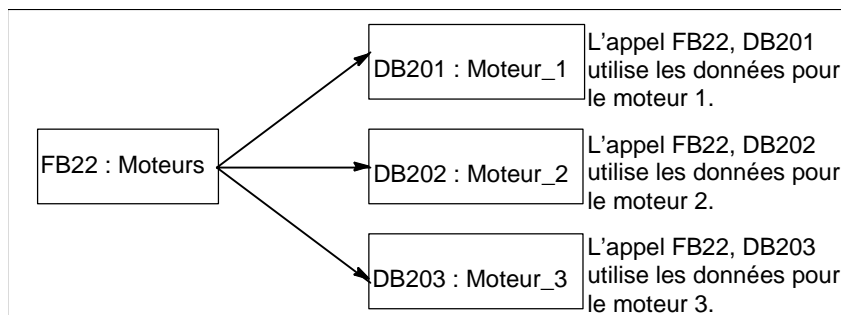
Création d'un DB d'instance

Le bloc fonctionnel correspondant à un DB d'instance doit exister avant que vous ne créiez ce DB d'instance. Vous indiquez le numéro de ce FB lors de la création du bloc de données d'instance.

Un DB d'instance pour chaque instance

Si vous associez plusieurs blocs de données d'instance à un bloc fonctionnel commandant un moteur, vous pourrez utiliser ce FB pour piloter différents moteurs.

Vous rangez les différentes données pour chaque moteur (comme, par exemple, régime, temps d'accélération, durée totale de fonctionnement) dans les différents blocs de données. Selon le DB associé au FB lors de l'appel, un autre moteur est commandé. Ainsi, un seul bloc fonctionnel est nécessaire pour plusieurs moteurs (voir la figure ci-après).

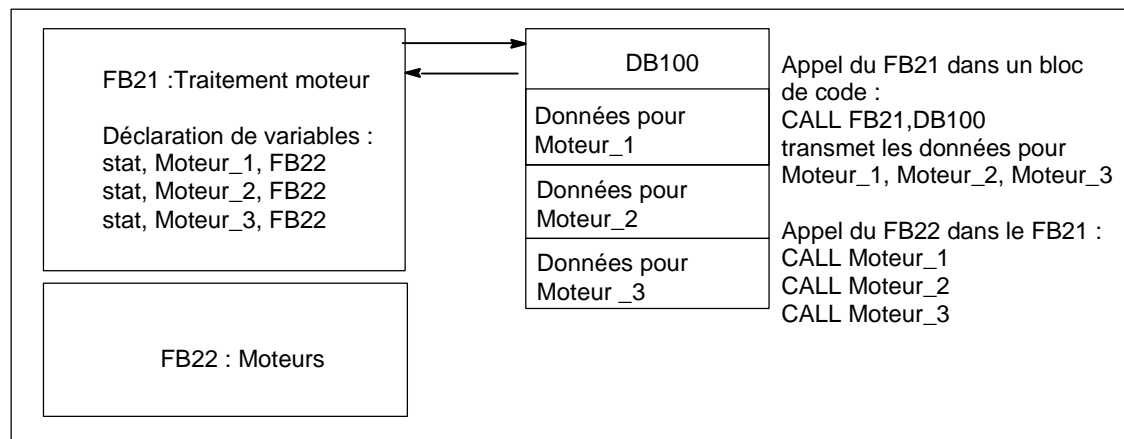


Un DB d'instance pour plusieurs instances d'un FB (multi-instances)

Vous pouvez transmettre à un bloc fonctionnel les données d'instance pour différents moteurs dans le même DB d'instance. Pour ce faire, vous devez appeler les commandes de moteur dans un autre FB dans la section de déclaration duquel vous déclarez des variables statiques de type de données FB pour les différentes instances.

Utiliser un seul DB d'instance pour plusieurs instances d'un FB vous permet de gagner de l'espace mémoire et d'optimiser l'utilisation des blocs de données.

Dans la figure suivante par exemple, le FB appelant est le FB21 "Traitement moteur", les variables sont de type de données FB22 et les instances sont identifiées par Moteur_1, Moteur_2 et Moteur_3.



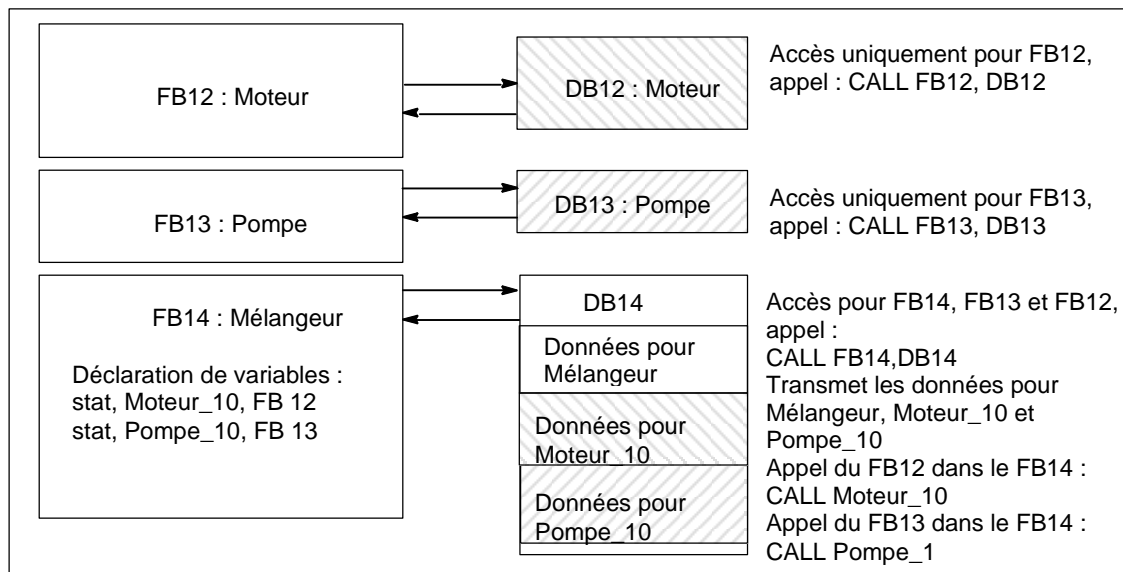
Le FB22 ne requiert pas de DB d'instance en propre dans cet exemple, car ses données d'instance sont contenues dans le DB d'instance du FB appelant.

Un DB d'instance pour plusieurs instances de différents FB (multi-instances)

Vous pouvez appeler, dans un bloc fonctionnel, des instances d'autres FB déjà créés. Vous pouvez associer les données d'instance nécessaires pour cela au bloc de données d'instance du FB appelant et n'avez ainsi pas besoin de blocs de données supplémentaires pour les FB appelés.

Pour ces multi-instances dans un DB d'instance, vous devez déclarer, dans la section de déclaration du FB appelant, des variables statiques avec le type de données du FB appelé pour les différentes instances. L'appel à l'intérieur du FB se fait alors sans indication de DB d'instance, mais uniquement via le nom de la variable.

Dans l'exemple de la figure, les données d'instance associées sont enregistrées ensemble dans un seul DB d'instance.



2.2.6 Blocs de données globaux (DB)

Contrairement aux blocs de code, les blocs de données ne contiennent pas d'instructions STEP 7 Lite. Ils servent à l'enregistrement de données utilisateur : ils contiennent des données variables que le programme utilisateur utilise. Les blocs de données globaux servent à l'enregistrement de données utilisateur pouvant être utilisées par tous les autres blocs.

La taille des DB peut varier. Vous trouverez la taille maximale autorisée dans le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*.

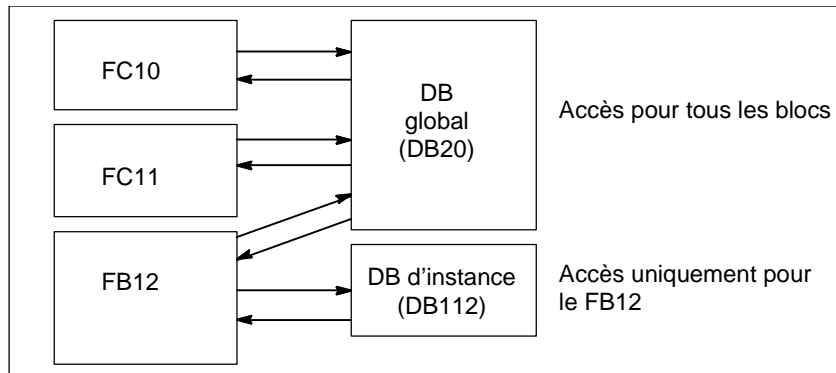
C'est vous qui définissez l'organisation des blocs de données globaux.

DB globaux dans le programme utilisateur

Lorsqu'il est appelé, un bloc de code (FC, FB ou OB) peut occuper temporairement de l'espace mémoire dans la zone des données locales (pile L). En plus de cette zone de données locales, ce bloc de code peut ouvrir une autre zone de mémoire sous la forme d'un DB. Contrairement aux données dans la zone des données locales, les données du DB ne sont pas effacées à la fermeture du DB ou à la fin du traitement du bloc de code correspondant.

Tout FB, FC ou OB peut lire les données contenues dans un DB global ou écrire des données dans un DB global. Ces données sont conservées dans le blocs de données même lorsqu'on quitte le DB.

Il est possible d'ouvrir simultanément un DB global et un DB d'instance. La figure ci-après présente les différents accès aux blocs de données.



2.2.6.1 Blocs fonctionnels système (SFB) et fonctions système (SFC)

Blocs déjà programmés

Il n'est pas nécessaire que vous programmez vous-même chaque fonction. En effet, les CPU S7 vous proposent des blocs tout prêts que vous pouvez appeler à partir du programme utilisateur.

De plus amples informations à ce sujet sont données dans l'aide de référence sur les blocs système et fonctions système (voir Sauts dans les descriptions de langage et les aides sur les blocs).

Blocs fonctionnels système

Un bloc fonctionnel système (SFB) est un bloc fonctionnel intégré à la CPU S7. Comme les SFB font partie du système d'exploitation, ils ne sont pas chargés en tant que partie du programme. Comme les FB, les SFB sont des blocs avec mémoire. Vous devez donc également créer pour les SFB des blocs de données d'instance que vous chargez dans la CPU en tant que partie du programme.

Les CPU S7 proposent des SFB :

- pour la communication via des liaisons configurées (mais non configurables dans STEP 7 Lite),
- pour des fonctions spéciales intégrées (par exemple, SFB29 HS_COUNT dans la CPU 312 IFM et la CPU 314 IFM).

Fonctions système

Une fonction système (SFC) est une fonction préprogrammée et intégrée dans la CPU S7. Vous pouvez appeler les SFC à partir de votre programme. Comme ces fonctions font partie du système d'exploitation, elles ne sont pas chargées en tant que partie du programme. Comme les FC, les SFC constituent des blocs sans mémoire.

Les CPU proposent des fonctions système pour :

- des fonctions de copie et de blocs,
- le contrôle du programme,
- la gestion de l'horloge et du compteur d'heures de fonctionnement,
- le transfert d'enregistrements logiques,
- la gestion des alarmes horaires et temporisées,
- la gestion des événements d'erreur synchrone, des événements d'alarme et des événements d'erreur asynchrone,
- l'information sur les données système statiques et dynamiques, par ex. le diagnostic,
- la mise à jour de la mémoire image du processus et le traitement de champ binaire,
- l'adressage de modules,
- la périphérie décentralisée (non configurable avec STEP 7 Lite),
- la communication par données globales (non configurable avec STEP 7 Lite),
- la communication via des liaisons non configurées,
- la création de messages relatifs aux blocs (non configurable avec STEP 7 Lite).

Informations supplémentaires

De plus amples informations sur les SFB et SFC sont données dans le manuel de référence *Logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système*. Les SFB et SFC disponibles sont précisés dans le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*.

3 Démarrage et utilisation du programme

3.1 Démarrage de STEP 7 Lite

Une fois Windows démarré, vous trouverez une icône pour STEP 7 Lite dans l'interface Windows.

La façon la plus rapide de démarrer STEP 7 Lite, c'est d'effectuer un double clic sur cette icône.

L'autre méthode consiste à lancer STEP 7 Lite via le bouton "Démarrer" dans la barre des tâches. L'entrée STEP 7 Lite figure sous "Simatic".

Nota

Vous trouverez plus d'informations sur les manipulations et options standard de Windows dans votre guide de l'utilisateur Windows ou dans l'aide en ligne de votre système d'exploitation Windows.

Pour poursuivre

Vous créez des solutions d'automatisation sous la forme de "projets". Vous vous faciliterez la tâche en vous familiarisant tout d'abord avec :

- l'interface utilisateur,
- quelques manipulations fondamentales,
- l'aide en ligne.

3.2 Appel des fonctions d'aide

Aide en ligne

L'aide en ligne vous renseigne à l'endroit où vous en avez besoin. Ceci vous évite de compulsier des manuels et représente un gain de temps et de précision. Elle comporte plusieurs parties :

- **Aide de STEP 7 Lite** : renseigne sur la marche à suivre de principe et transmet les connaissances élémentaires requises pour configurer et programmer un automate programmable.
- **Qu'est-ce que c'est ?** (touches MAJ+F1) : renseigne sur un élément sélectionné, par exemple sur une boîte de dialogue.
- **A propos de** : renseigne sur la version actuelle de l'application.

Appel de l'aide en ligne

Il y a plusieurs façons d'appeler l'aide en ligne :

- Choisissez une commande dans le menu "?" dans la barre des menus.
- Cliquez sur l'icône de pointeur d'aide dans la barre d'outils et sélectionnez l'élément pour lequel vous avez besoin d'aide.
- Appuyez sur les touches MAJ+F1 et sélectionnez l'élément pour lequel vous avez besoin d'aide avec le pointeur d'aide.
- Appuyez sur la touche F1 pour afficher l'aide de STEP 7 Lite.

Appel de la bulle d'aide

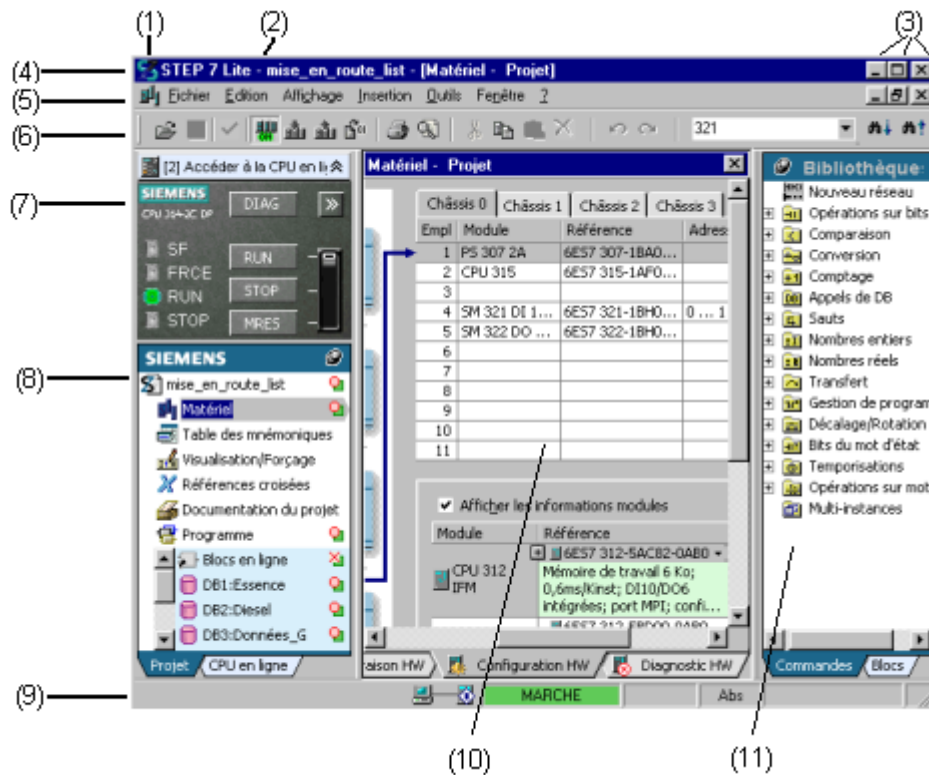
Pointez sur une icône de la barre d'outils, par exemple, ou sur un élément des bibliothèques d'opérations ou de blocs pour afficher la bulle d'aide respective contenant des informations succinctes.

Un symbole placé dans le coin inférieur droit de la bulle d'aide indique qu'il est possible d'appeler l'aide contextuelle depuis là. Cette dernière s'ouvrira automatiquement si vous attendez un petit peu, mais vous pouvez aussi l'appeler activement en cliquant sur la bulle d'aide.

3.3 Interface utilisateur et manipulation

3.3.1 Interface utilisateur

La figure suivante présente les différentes zones de l'interface utilisateur de STEP 7 Lite.



(1)	Menu système (pleine page, fermeture etc.)	(7)	Panneau de commande CPU
(2)	Titre de la fenêtre active	(8)	Projektfenster
(3)	Boutons pour icône, pleine image, fermeture	(9)	Barre d'état
(4)	Barre de titre	(10)	Zone de travail : contient des informations que vous pouvez afficher ou éditer
(5)	Barre des menus	(11)	Bibliothèques
(6)	Barre d'outils		

Barre de titre et barre des menus

La barre de titre et la barre des menus se trouvent toujours au bord supérieur de la fenêtre. La barre de titre contient le titre de la fenêtre et les boutons servant à la commander. La barre des menus présente tous les menus disponibles dans cette fenêtre.

Barre d'outils

La barre d'outils contient des icônes vous permettant d'accéder rapidement par clic de la souris aux commandes de menu disponibles qui sont le plus souvent utilisées. Pointez sur une icône pour afficher une information succincte sur sa fonction.

Les icônes non disponibles dans la configuration en cours sont estompées.

Panneau de commande CPU

Le panneau de commande de la CPU représente une unité centrale SIMATIC S7-300. En cas de liaison en ligne à une CPU, l'état de fonctionnement de cette dernière est indiqué au moyen des diodes électroluminescentes et de la position du commutateur à clé. Certains boutons, tels que "RUN" et "STOP", permettent de commander la CPU.

Bibliothèques

Dans cette zone, l'onglet "Instructions" présente les opérations CONT et LOG tandis que l'onglet "Blocs" énumère toutes les bibliothèques que le système connaît.

Fenêtre du projet

Cette zone permet de sélectionner des objets du projet pour les éditer.

Exemple : quand vous cliquez sur la ligne "Table des mnémoniques", cette dernière s'affiche dans la zone de travail où vous pouvez l'éditer.

Zone de travail


















Cette zone affiche l'objet sélectionné dans la fenêtre du projet pour que vous puissiez l'éditer.

Exemple : quand vous avez sélectionné un bloc dans la fenêtre du projet, vous pouvez l'éditer dans l'éditeur de bloc.

Barre d'état

La barre d'état affiche des informations contextuelles.

3.3.2 Icônes utilisées dans la fenêtre du projet

Icône	Signification	Icône	Signification
	Projet hors ligne		CPU en ligne
	Configuration matérielle		Table des mnémoniques
	Visualisation/ Forçage		Références croisées ("Liste des références croisées", "Opérandes utilisés", "Structure du programme")
	Documentation du projet		Programme (somme de tous les blocs de code et blocs de données)
	Blocs en ligne/hors ligne Cette entrée remplace des blocs en ligne qui ne se trouvent pas dans le projet hors ligne. Un double-clic sur l'icône ouvre une boîte de dialogue permettant de charger les blocs depuis la CPU dans le projet hors ligne.		Bloc de code L'icône change de couleur selon le type de bloc.
	Bloc de données		Carte mémoire (carte mémoire Micro MMC)
	Modifié mais pas enregistré L'étoile jaune signifie que l'objet a été modifié mais que son contenu n'a pas encore été enregistré.		L'objet dans le projet et dans la CPU en ligne sont identiques.
	Protection de bloc L'icône de cadenas signifie que le bloc est muni d'une protection ; il ne peut être ni lu ni modifié sans autorisation spéciale.		L'objet existe dans la vue actuelle de la fenêtre du projet, mais manque dans la vue non sélectionnée (vue "Projet" / vue "CPU en ligne"). Le curseur d'aide de STEP 7 Lite vous permet d'obtenir de plus amples informations.
	L'objet dans le projet ne correspond pas à celui dans la CPU en ligne. Le curseur d'aide de STEP 7 Lite vous permet d'obtenir de plus amples informations.		

3.3.3 Eléments des fenêtres et des boîtes de dialogue

Dans les fenêtres et les boîtes de dialogue, des éléments spécifiques servent à la manipulation, à la sélection et à la saisie. Ils sont expliqués ci-après à l'aide d'exemples.

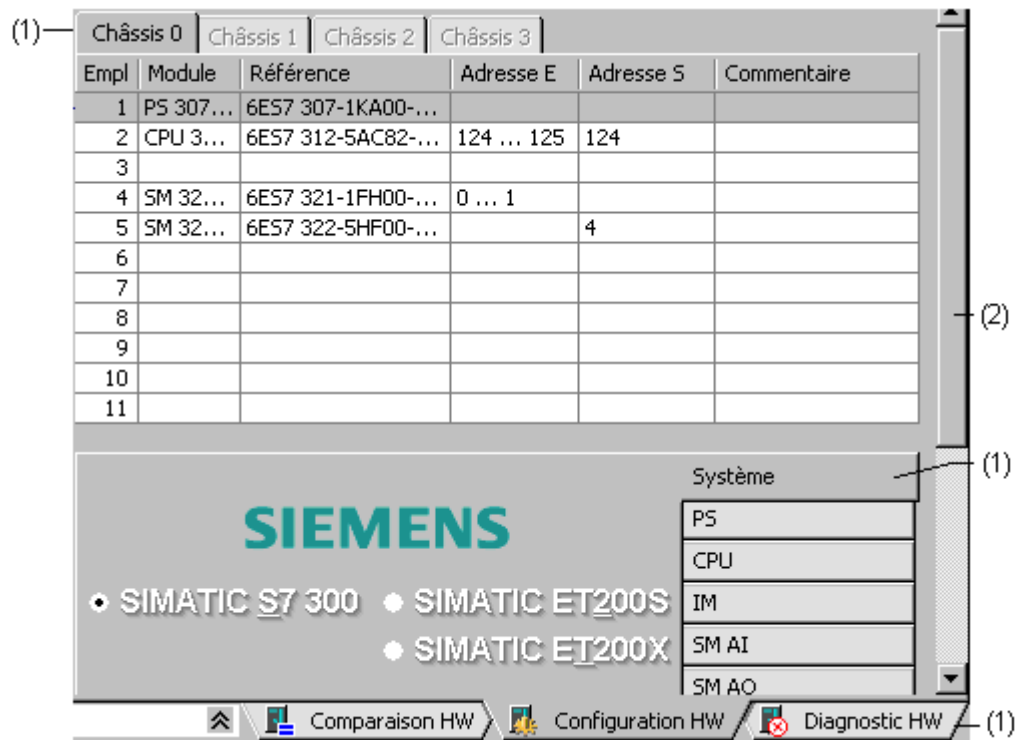
The screenshot shows a dialog box titled "Nouveau bloc" with a blue header bar containing a help icon (?) and a close icon (X). The dialog is divided into several sections:

- Section 1:** A group box containing four radio buttons: "Bloc d'organisation", "Bloc fonctionnel" (selected), "Fonction", and "Bloc de données". To the right of these are input fields for "FB" (value 2), "FC" (value 2), "DB" (value 4), and "UDT" (value 1). A dropdown menu for "Numéro" shows "OB1: CYCL_EXC".
- Section 2:** A group box labeled "Propriétés de bloc" containing two text input fields: "Mnémonique :" and "Comm. mnémonique :".
- Section 3:** A group box labeled "Propriétés de l'en-tête de bloc" containing four input fields: "Nom (en-tête) :", "Version (en-tête) :" (value 1.0), "Famille :", and "Auteur :".
- Section 4:** A group box containing a dropdown menu for "Langage de création" (value LIST) and a checked checkbox for "Multiinstance".
- Section 5:** Two buttons at the bottom: "OK" and "Annuler".

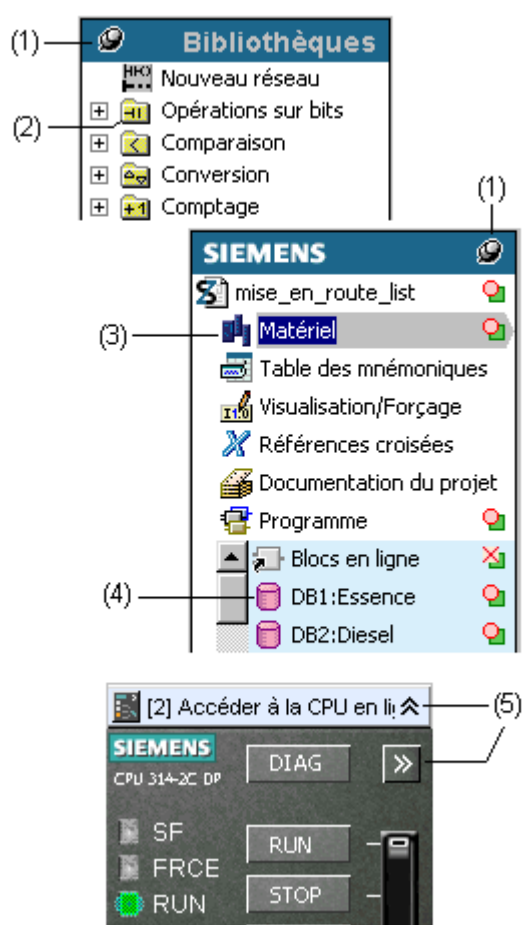
Numbered callouts (1) through (5) point to the following elements:



- (1) Points to the "Numéro" dropdown menu.
- (2) Points to the "Bloc fonctionnel" radio button.
- (3) Points to the "Mnémonique :" text input field.
- (4) Points to the "Langage de création" dropdown menu.
- (5) Points to the "OK" button.

(1)	Listes déroulantes : Une flèche pointant vers le bas indique qu'une liste peut être déroulée pour ce champ.
(2)	Les boutons radio pour la sélection d'une ou de plusieurs options.
(3)	Les champs de textes pour la saisie de texte ou de nombres via le clavier
(4)	Les cases à cocher pour la sélection d'une ou de plusieurs options.
(5)	Boutons



(1)	Onglets : Le contenu de certaines fenêtres et boîtes de dialogue est réparti pour plus de clarté sur plusieurs onglets. Pour afficher une page d'onglet particulière, il vous suffit de cliquer sur l'onglet correspondant.
(2)	Barre de défilement : Déplacez la barre de défilement ou cliquez sur la flèche pour afficher la partie de la fenêtre ou de la boîte de dialogue non visible.



(1)	<p>Punaise : La punaise permet de fixer la fenêtre de projet ou des bibliothèques. Une fenêtre non fixée passe automatiquement à l'arrière-plan pour laisser la plus grande zone de travail possible. Si le curseur de la souris se trouve sur le bord droit ou gauche de la fenêtre de STEP 7 Lite, la fenêtre de projet et la bibliothèque passent à l'avant-plan. Tant que le curseur de la souris est positionné dans une zone pouvant être masquée, la fenêtre de projet et la bibliothèque restent à l'avant-plan.</p> <p>Un clic simple sur l'icône permet de fixer la fenêtre. Un nouveau clic la libère.</p> <p> La fenêtre de projet ou les bibliothèques sont fixées.</p> <p> La fenêtre de projet ou les bibliothèques ne sont pas fixées.</p>
(2)	<p>Dossiers : Les instructions dans les bibliothèques sont réparties dans plusieurs dossiers. Vous pouvez ouvrir le dossier par double-clic et faire glisser à l'aide de la souris l'instruction souhaitée dans l'éditeur de blocs.</p>
(3)	<p>Icônes permettant d'appeler des objets et fonctions centraux (par ex. Matériel, Table des mnémoniques, Visualisation et forçage etc.)</p>
(4)	<p>Blocs du projet : un double-clic sur les blocs de données ou les blocs de code lance l'éditeur correspondant</p>
(5)	<p>Boutons Elargir et Réduire : pour l'élargissement ou la réduction de fenêtres</p> <p>Une fois la fenêtre agrandie, la dénomination du bouton change.</p>

3.3.4 Mémoire de session

STEP 7 Lite est capable de mémoriser le contenu des fenêtres, c'est-à-dire les projets ouverts, ainsi que la disposition des fenêtres. Il conserve aussi le jeu d'abréviations choisi (allemand ou anglais).

- La commande **Fenêtre > Enregistrer la disposition** enregistre le contenu et la disposition actuels des fenêtres.
- La commande **Fenêtre > Restaurer la disposition** restaure le contenu et la disposition des fenêtres précédemment enregistrés à l'aide de la commande **Fenêtre > Enregistrer la disposition**.

Avertissement

Le contenu des fenêtres en ligne n'est pas enregistré (par ex. les blocs se trouvant dans la CPU connectée).

Les mots de passe éventuellement saisis pour l'accès à des systèmes cible (par ex. S7-300) ne sont pas enregistrés au-delà de la fin de la session.

3.3.5 Modifier la disposition des fenêtres

Pour disposer toutes les fenêtres affichées de manière à ce qu'elles se chevauchent, choisissez la commande **Fenêtre > Disposition > Cascade**.

Pour disposer toutes les fenêtres les unes sous les autres, choisissez la commande **Fenêtre > Disposition > Mosaïque verticale**.

Pour disposer toutes les fenêtres les unes à côté des autres, choisissez la commande **Fenêtre > Disposition > Mosaïque horizontale**.

3.3.6 Enregistrer et restaurer la disposition des fenêtres

STEP 7 Lite offre la possibilité d'enregistrer la disposition actuelle des fenêtres afin de pouvoir la restaurer à tout moment. Ce paramétrage peut être réalisé à l'aide de la commande **Outils > Paramètres**.

Informations mémorisées

Lorsque vous enregistrez la disposition des fenêtres, les informations suivantes sont mémorisées :

- fenêtres ouvertes avec leur position,
- éventuellement, ordre de superposition des fenêtres.

Enregistrer la disposition des fenêtres

Pour enregistrer la disposition actuelle des fenêtres, choisissez la commande **Fenêtre > Enregistrer la disposition**.

Restaurer la disposition des fenêtres

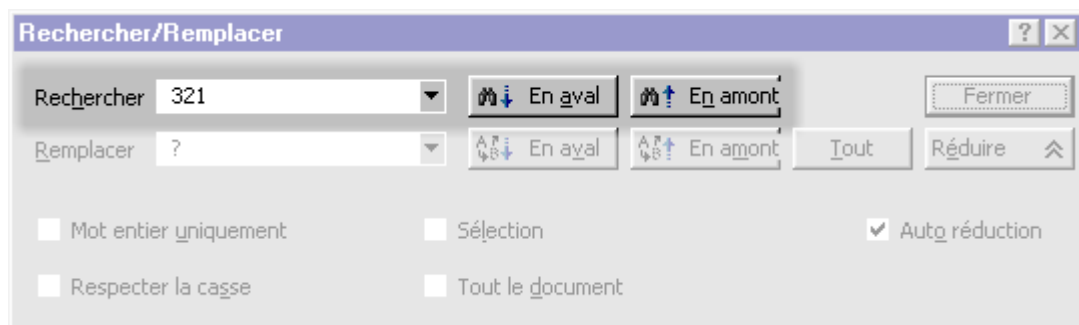
Pour restaurer une disposition des fenêtres enregistrée, choisissez la commande **Fenêtre > Restaurer la disposition**.

3.3.7 Recherche ou remplacement de termes

Les fonctions de recherche ou de recherche/remplacement de STEP 7 Lite constituent une aide précieuse lorsqu'il s'agit d'éditer des projets et d'y effectuer des recherches. Ces fonctions sont par principe disponibles dans les configurations matérielles, les tables de mnémoniques, les références croisées et l'éditeur de blocs. Les options proposées dans la boîte de dialogue dépendent de la vue sélectionnée. Ainsi, p. ex., les fonctions de remplacement sont uniquement disponibles dans les vues dans lesquelles l'édition est possible.

Recherche d'un terme

Pour rechercher un terme, utilisez la liste déroulante et les boutons de la ligne supérieure de la boîte de dialogue.

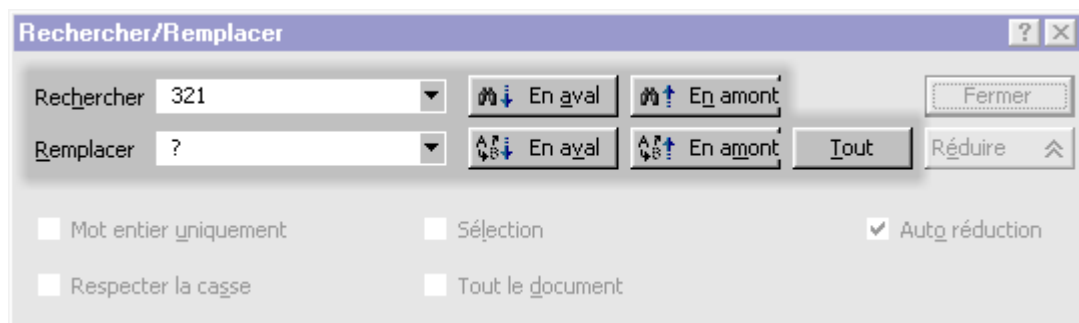


Procédez de la manière suivante :

1. Saisissez le terme à rechercher dans la zone de saisie de la liste déroulante ou choisissez un terme déjà proposé dans cette liste.
2. Démarrez la recherche en cliquant sur le bouton "Bas" ou "Haut". La recherche commence à la position du curseur dans le sens indiqué. Pour poursuivre la recherche, cliquez une nouvelle fois sur ce bouton.

Recherche et remplacement d'un terme

Pour rechercher et remplacer un terme, utilisez la liste déroulante de la fonction de recherche ainsi que la liste déroulante et les boutons de la deuxième ligne de la boîte de dialogue.

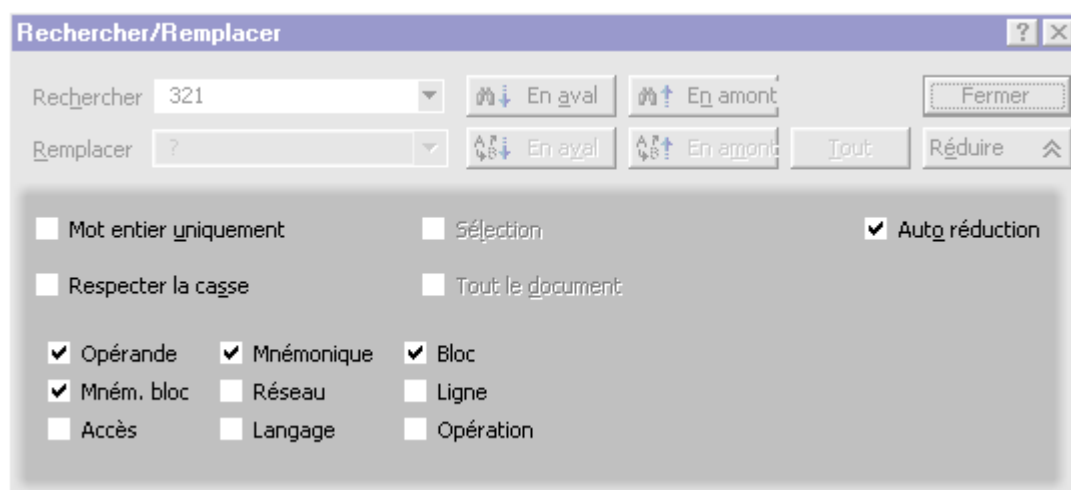


Procédez de la manière suivante :

1. Saisissez le terme à rechercher dans la zone de saisie de la liste déroulante de recherche ou choisissez un terme déjà proposé dans cette liste.
2. Saisissez le terme à remplacer dans la zone de saisie de la liste déroulante de remplacement ou choisissez un terme déjà proposé dans cette liste.
3. Démarrez la fonction de recherche et de remplacement. Vous avez les possibilités suivantes :
 - "Bas"
 - "Haut"
 - "Tout"

Extension ou réduction du nombre de résultats de recherche au moyen de critères de recherche

Vous pouvez étendre ou réduire le nombre de résultats de recherche en sélectionnant des critères de recherche supplémentaires. Ces derniers s'affichent dans la partie inférieure de la boîte de dialogue, lorsque celle-ci a été étendue avec le bouton "Etendre/Réduire". Le nombre et le type de critères de recherche affichés dépend de la vue active. De plus amples informations sur les critères de recherche sont données dans l'aide directe de STEP 7 Lite.



3.3.8 Manipulation des objets

3.3.8.1 Renommer un objet

Pour renommer un objet, procédez de la manière suivante :

1. Sélectionnez l'objet souhaité.
2. Cliquez sur le nom de l'objet sélectionné pour le rendre éditable. La zone de nom s'entoure d'un cadre et le pointeur se transforme en curseur de texte.
3. Editez le nom de l'objet. De manière générale, il faut respecter les conventions en vigueur pour l'attribution de noms sous la version de Windows utilisée.
4. Pour terminer l'édition, plusieurs possibilités s'offrent à vous :
 - Appuyez sur la touche d'entrée pour reprendre le nom modifié. Si le nouveau nom n'est pas autorisé, il est remplacé par l'ancien.
 - Appuyez sur la touche ECHAP pour interrompre l'édition et reprendre le nom précédent de l'objet.

3.3.8.2 Déplacer un objet

Vous pouvez déplacer les objets-blocs dans l'ordre de leur énumération dans la fenêtre du projet. Procédez de la manière suivante :

1. Sélectionnez le bloc que vous souhaitez déplacer et maintenez le bouton gauche de la souris enfoncé.
2. Amenez le pointeur à l'endroit où vous souhaitez placer le bloc. Une barre noire indique à quel endroit il peut être inséré.
3. Relâchez le bouton gauche de la souris pour lâcher le bloc.

3.3.8.3 Supprimer un objet

Pour supprimer un objet, procédez de la manière suivante :

1. Sélectionnez l'objet que vous souhaitez supprimer.
2. Pour le supprimer, vous avez plusieurs possibilités :
 - Choisissez la commande **Edition > Effacer**.
 - Appuyez sur la touche SUPPR.
3. Confirmez la suppression de l'objet en cliquant sur le bouton "Oui" dans le message qui s'affiche.

3.4 Utilisation du clavier

Désignation internationale des touches	Désignation française des touches
HOME	ORIGINE
END	FIN
PAGE-UP	PG.PREC
PAGE-DOWN	PG.SUIV
CTRL	CTRL
ENTER	ENTREE
DEL	SUPPR
INSERT	INSER

3.4.1 Combinaisons de touches pour les commandes de menu

Vous pouvez déclencher chaque commande en tapant la combinaison de la touche ALT avec la lettre soulignée appropriée.

Appuyez successivement sur les touches suivantes :

- touche ALT,
- lettre soulignée dans le menu désiré (par exemple, ALT, F pour le menu Fichier si le menu Fichier figure dans la barre des menus). Le menu s'ouvre.
- lettre soulignée dans la commande désirée (par exemple N pour la commande Nouveau). S'il s'agit d'une commande comportant des sous-menus, ceux-ci s'ouvriront. Continuez à taper les lettres soulignées, jusqu'à ce que la commande souhaitée soit complète.

La commande est déclenchée une fois la dernière lettre de la combinaison de touches saisie.

Exemples :

Commandes	Touches
Fichier > Ouvrir	ALT, F, O

Equivalences clavier

Fonction		Equivalence
Nouveau > Bloc	(menu Fichier)	CTRL + N
Enregistrer	(menu Fichier)	CTRL + S
Fermer	(menu Fichier)	CTRL + F4
Ouvrir projet	(menu Fichier)	CTRL + O
Charger dans CPU	(menu Fichier)	CTRL + L
Imprimer (objet)	(menu Fichier)	CTRL + P
Quitter	(menu Fichier)	ALT + F4
Annuler	(menu Edition)	CTRL + Z
Rétablir	(menu Edition)	CTRL + Y
Couper	(menu Edition)	CTRL + X
Copier	(menu Edition)	CTRL + C
Coller	(menu Edition)	CTRL + V
Effacer	(menu Edition)	SUPPR
Renommer	(menu Edition)	F2
Sélectionner tout	(menu Edition)	CTRL + A
Rechercher/remplacer	(menu Edition)	CTRL + F
Aller à > Réseau/ligne	(menu Edition)	CTRL + E
Aller à > Occurrence	(menu Edition)	CTRL + ALT + Q
Aller à > Erreur précédente	(menu Edition)	ALT + F7
Aller à > Erreur suivante	(menu Edition)	ALT + F8
Ouvrir bloc	(menu Edition)	CTRL + ALT + O
Mnémoniques	(menu Edition)	ALT + ENTREE
Réseau	(menu Insertion)	CTRL + R
Mnémonique	(menu Insertion)	CTRL + J
Visualiser	(menu Test)	CTRL + F7
CONT	(menu Affichage)	CTRL + 1
LOG	(menu Affichage)	CTRL + 3
LIST	(menu Affichage)	CTRL + 2
Agrandir	(menu Affichage)	CTRL + Num+
Réduire	(menu Affichage)	CTRL + Num-
Représentation symbolique	(menu Affichage)	CTRL + Q
Information mnémonique	(menu Affichage)	CTRL + MAJ + Q
Sélection de mnémoniques	(menu Affichage)	CTRL + 7
Commentaire	(menu Affichage)	CTRL + MAJ + K
Panneau de commande CPU	(menu Affichage)	CTRL + ALT + C
Fenêtre du projet	(menu Affichage)	CTRL + ALT + P
Bibliothèques	(menu Affichage)	CTRL + ALT + L
Actualiser la vue	(menu Affichage)	F5
Paramètres	(menu Outils)	CTRL + ALT + E
Etat du module	(menu Outils)	CTRL + D
Cascade	(menu Fenêtre)	MAJ + F5
Mosaïque horizontale	(menu Fenêtre)	MAJ + F2
Mosaïque verticale	(menu Fenêtre)	MAJ + F3
Configuration matérielle	(menu Fenêtre)	CTRL + ALT + H
Table des mnémoniques	(menu Fenêtre)	CTRL + ALT + T
Visualisation/forçage	(menu Fenêtre)	CTRL + ALT + W
Références croisées	(menu Fenêtre)	CTRL + ALT + X
Aide sur STEP 7 Lite	(menu d'aide ?)	F1

Fonction	Equivalence
Aide directe (menu d'aide ?)	MAJ + F1 (S'il y a un contexte, par ex. si une commande est sélectionnée, la rubrique d'aide correspondante s'affiche. Sinon, c'est le sommaire de l'aide qui s'affiche.)
Appel du menu contextuel	MAJ + F10

3.4.2 Combinaisons de touches pour le déplacement du pointeur

Déplacement du pointeur dans la barre des menus ou dans un menu contextuel

Fonction	Touches
Activer la barre des menus	F10
Appeler le menu contextuel	MAJ + F10
Au menu dont le nom contient le caractère souligné X	ALT + X
Choisir une commande subordonnée	Lettre soulignée dans le nom de commande
Déplacement d'un menu vers la gauche	Flèche vers la gauche
Déplacement d'un menu vers la droite	Flèche vers la droite
Déplacement d'un menu vers le haut	Flèche vers le haut
Déplacement d'un menu vers le bas	Flèche vers le bas
Activer la commande sélectionnée	ENTREE
Quitter le menu ou revenir au texte	ECHAP

Déplacement du curseur pour l'édition de texte

Fonction	Touches
Une ligne vers le haut ou un caractère vers la gauche dans un texte d'une seule ligne	Flèche vers le haut
Une ligne vers le bas ou un caractère vers la droite dans un texte d'une seule ligne	Flèche vers le bas
Un caractère vers la droite	Flèche vers la droite
Un caractère vers la gauche	Flèche vers la gauche
Un mot vers la droite	CTRL + flèche vers la droite
Un mot vers la gauche	CTRL + flèche vers la gauche
Au début de la ligne	ORIGINE
A la fin de la ligne	FIN
Une page d'écran vers le haut	PAGE PRECEDENTE
Une page d'écran vers le bas	PAGE SUIVANTE
Au début du texte	CTRL + ORIGINE
A la fin du texte	CTRL + FIN

Déplacement du curseur dans les boîtes de dialogue

Fonction	Touches
Au champ de saisie suivant (de gauche à droite et de haut en bas)	TAB
Au champ de saisie précédent	MAJ + TAB
Au champ de saisie dont le nom contient le caractère souligné X	ALT + X
Sélectionner dans une liste de sélection	Touches de direction
Ouvrir une liste de sélection	ALT + flèche vers le bas
Sélectionner un objet ou en annuler la sélection	ESPACE
Confirmer la saisie et fermer la boîte de dialogue (bouton "OK")	ENTREE
Fermer la boîte de dialogue sans enregistrer la saisie (bouton "Annuler").	ECHAP

3.4.3 Combinaisons de touches pour la sélection de texte

Fonction	Touches
Un caractère vers la droite	MAJ + flèche vers la droite
Un caractère vers la gauche	MAJ + flèche vers la gauche
Jusqu'au début de la ligne de commentaire	MAJ + ORIGINE
Jusqu'à la fin de la ligne de commentaire	MAJ + FIN
Une ligne vers le haut	MAJ + flèche vers le haut
Une ligne vers le bas	MAJ + flèche vers le bas
Une page d'écran vers le haut	MAJ + PG.PREC
Une page d'écran vers le bas	MAJ + PG.SUIV
Le texte jusqu'au début du fichier	CTRL + MAJ + ORIGINE
Le texte jusqu'à la fin du fichier	CTRL + MAJ + FIN

3.4.4 Combinaisons de touches pour accéder à l'aide en ligne

Fonction	Touches
Activer le bouton "Point d'interrogation" pour obtenir une aide directe	MAJ + F1
Ouvrir l'aide de STEP 7 Lite	F1
Fermer la fenêtre d'aide	ALT + F4

3.4.5 Combinaisons de touches pour la bascule entre les différents types de fenêtres

Fonction	Touches
Basculer d'un volet à un autre	F6
Revenir au volet précédent, en l'absence d'une fenêtre ancrée	MAJ + F6
Basculer entre la fenêtre du document et la fenêtre ancrée au document (par ex. fenêtre de déclaration des variables) En l'absence d'une fenêtre ancrée, la bascule ramène au volet précédent.	MAJ + F6
Basculer entre des fenêtres de document	CTRL + F6
Revenir à la fenêtre de document précédente	MAJ + CTRL + F6
Basculer entre des fenêtres autres que de documents (fenêtres d'applications et fenêtre qui y sont ancrées ; lors du retour à une fenêtre d'application, la bascule mène à la dernière fenêtre de document active)	ALT + F6
Basculer entre la fenêtre de projet, le panneau de commande de la CPU, la fenêtre des bibliothèques et la fenêtre du document actif	CTRL + ALT + F6
Revenir à la fenêtre autre que de document précédente	MAJ + ALT + F6
Fermer la fenêtre active	CTRL + F4

3.5 Commande d'une installation à distance grâce au Téléservice

Le logiciel optionnel Téléservice vous offre la possibilité d'établir une liaison en ligne à une installation distante depuis votre PG/PC, via le réseau téléphonique. Vous pouvez alors commander cette installation comme vous y êtes habitué dans STEP 7 Lite.

En raison des temps de réaction plus longs, ce type de liaison n'est à recommander qu'à des fins de maintenance.

Conditions préalables

Les conditions suivantes doivent être remplies pour la mise en œuvre de Téléservice :

- Le logiciel optionnel Téléservice doit être installé.
- Un modem local doit être installé et configuré sous Windows.
- L'installation distante doit être raccordée à un réseau téléphonique par l'intermédiaire d'un adaptateur TS paramétré et d'un modem.
- Avec "Paramétrage de l'interface PG/PC", vous devez régler l'entrée de l'application et ses propriétés sur l'adaptateur TS.

Appel

Une fois le logiciel optionnel installé, vous pouvez appeler Téléservice en choisissant la commande **Outils Logiciels optionnels > Téléservice**.

Nota

De plus amples informations sont données dans la documentation ainsi que dans l'aide en ligne du logiciel optionnel.

4 Création et édition du projet

4.1 Qu'est-ce qu'un projet STEP 7 Lite ?

Un projet STEP 7 Lite comprend toutes les données d'un automate SIMATIC S7-300, C7 ou d'un système DP modulaire ET 200X ou ET 200S (autonome).

Un projet sert à stocker en bon ordre les données produites par l'élaboration d'une solution d'automatisation. Dans le cas de STEP 7 Lite, il regroupe les données d'une station et en particulier :




- les données concernant la configuration matérielle et le paramétrage des modules de la station,
- la table des mnémoniques de la station,
- la table des variables permettant de visualiser et de forcer les opérandes de la station,
- la documentation du projet avec des précisions sur son contenu et sa forme,
- le programme utilisateur de la station.

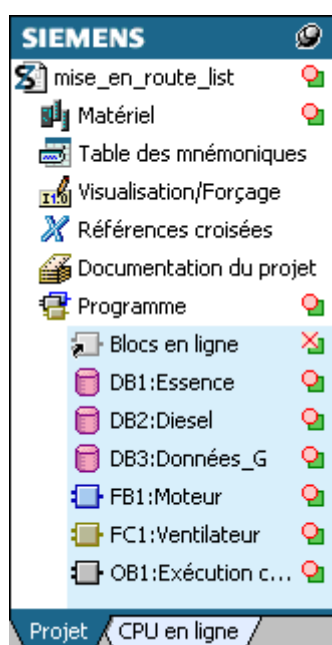
Fenêtre du projet (onglet "Projet" au premier plan)

La fenêtre du projet montre dans son onglet "Projet" :

- L'icône représentant le projet en cours avec le nom de projet actuel. Un double clic sur cette icône affiche la vue des blocs et de la configuration matérielle.
- L'icône représentant le matériel – un double clic sur cette icône ouvre la vue du paramétrage de la configuration matérielle, du diagnostic du matériel et de la comparaison du matériel.
- L'icône représentant la table des mnémoniques – un double clic sur cette icône ouvre la vue dans laquelle vous créez et gérez les mnémoniques du projet.
- L'icône représentant la "Visualisation/Forçage" – un double clic sur cette icône ouvre la vue de visualisation/forçage ou de visualisation/forçage permanent de variables.
- L'icône représentant les références croisées – un double clic sur cette icône ouvre la vue affichant les références croisées, les opérandes utilisés et la structure du programme.
- L'icône représentant la documentation du projet – un double clic sur cette icône ouvre la vue dans laquelle vous élaborez et constituez votre documentation de projet STEP 7 Lite.
- L'icône représentant le programme utilisateur, sous laquelle les blocs du projet sont énumérés. Un double clic sur cette icône ouvre la vue de tous les blocs existant dans le projet. Un double clic sur une icône représentant un bloc ouvre ce bloc dans l'éditeur associé.

Lorsqu'une liaison en ligne est établie avec une CPU, des icônes supplémentaires sont affichées à côté des icônes représentant le projet, le matériel et le programme avec ses blocs.

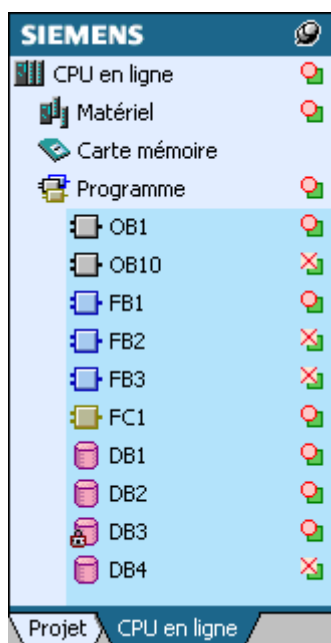
Icône	Signification
	L'objet est identique dans le projet et dans la CPU en ligne.
	L'objet dans le projet ne correspond pas à celui dans la CPU en ligne. Le curseur d'aide de STEP 7 Lite vous permet d'obtenir une aide supplémentaire.
	L'objet existe dans la vue actuelle de la fenêtre du projet, mais manque dans la vue non sélectionnée (vue "Projet" / vue "CPU en ligne"). Le curseur d'aide de STEP 7 Lite vous permet d'obtenir une aide supplémentaire.



Fenêtre du projet (onglet "CPU en ligne" au premier plan)

Dans le cas d'une liaison en ligne avec une CPU, la fenêtre du projet montre dans son onglet "CPU en ligne" :

- L'icône représentant la "CPU en ligne". Un double clic sur cette icône affiche un aperçu des blocs chargés dans la CPU, de la configuration matérielle et éventuellement une entrée groupée pour les fichiers enregistrés sur la carte mémoire Micro SIMATIC (MMC). L'entrée groupée s'affiche uniquement dans le cas d'une CPU 31xC dans laquelle une carte mémoire MMC est enfichée.
- L'icône représentant le matériel – un double clic sur cette icône ouvre la vue affichant la configuration matérielle chargée dans la CPU et le diagnostic du matériel.
- L'icône représentant une carte mémoire. Cette icône s'affiche uniquement dans le cas d'une CPU 31xC dans laquelle une carte mémoire MMC est enfichée. Un double clic sur cette icône ouvre un aperçu des fichiers enregistrés sur la carte MMC.
- L'icône représentant le programme utilisateur, sous laquelle les blocs de la CPU sont énumérés. Un double clic sur cette icône ouvre la vue de tous les blocs existant dans la CPU (selon le paramétrage effectué dans **Outils > Paramètres > Afficher les blocs système**, également avec les blocs système). Un double clic sur une icône représentant un bloc ouvre ce bloc dans l'éditeur associé, avec une protection contre l'écriture.



4.2 Création d'un projet

4.2.1 Création d'un projet

Pour résoudre votre problème d'automatisation dans le cadre d'une gestion de projet, vous commencez par créer un nouveau projet.

1. Choisissez la commande **Fichier > Nouveau**.
Un nouveau projet est alors créé avec des contenus par défaut dans la fenêtre de projet (par ex. l'icône représentant la configuration matérielle).
Lorsqu'un projet est déjà ouvert, il est d'abord fermé, puis le nouveau projet est ouvert.
2. Le nouveau projet a pour nom "Nouveau projet". Déterminez le nom définitif du projet en enregistrant (Enregistrer/Enregistrer sous).

4.2.2 Insertion d'un programme

Composants déjà créés

Si vous avez créé un projet, les composants "Configuration matérielle", "Table des mnémoniques" et "Programme" existent déjà.

Insérer des blocs

1. Choisissez la commande **Insertion > Bloc...**
2. La boîte de dialogue "Nouveau bloc" s'ouvre et vous permet de déterminer la nature du bloc (une fonction, par ex.) et ses propriétés (son mnémonique, par ex.).

Regrouper les blocs

Quand il y a un grand nombre de blocs à gérer, vous pouvez créer des catégories pour plus de clarté et amener les blocs par glisser-lâcher dans ces catégories.

1. Choisissez la commande **Insertion > Catégorie**.
2. Donnez un nom à la catégorie.
3. Faites glisser les blocs appropriés dans la nouvelle catégorie.

Utiliser des blocs tirés de bibliothèques

Pour créer vos programmes utilisateur, vous pouvez aussi utiliser des blocs tirés des bibliothèques standard faisant partie du logiciel. Vous les trouverez dans l'onglet "Blocs" de la fenêtre de bibliothèque. Si cette fenêtre n'est pas affichée, choisissez la commande **Affichage > Adapter > Bibliothèques**. Pour plus de détails sur les bibliothèques standard, consultez la rubrique Présentation des bibliothèques de blocs.

Créer une table des mnémoniques

Une table des mnémoniques vide (objet "Table des mnémoniques") est automatiquement générée lorsque vous créez le programme. La sélection de l'objet "Table des mnémoniques" affiche la fenêtre "Table des mnémoniques" qui présente la table. Pour plus de détails, consultez la rubrique Saisie de plusieurs mnémoniques globaux dans la table des mnémoniques.

4.3 Edition d'un projet

Ouverture d'un projet

Pour ouvrir un projet existant, déjà enregistré, procédez de la manière suivante :

1. Choisissez la commande **Fichier > Ouvrir....**
2. Sélectionnez un projet :
 - Les projets que vous avez déjà édités sur votre PG/PC se trouvent dans l'onglet "Traités en dernier".
 - Les projets que vous n'avez pas encore édités se trouvent dans l'onglet "Système de fichiers". Sélectionnez le chemin et le projet souhaités dans la vue de l'explorateur proposée dans cet onglet.
 - Si votre projet a été enregistré sur la carte mémoire Micro d'une CPU 31xC, il se trouve dans l'onglet "Carte mémoire". Il faut cependant qu'une liaison en ligne soit établie avec la CPU 31xC. Afin de réduire de longues durées de transfert, le projet est ouvert en tant que "copie".

Copie d'un projet


Vous copiez un projet en l'enregistrant sous un autre nom via la commande **Fichier > Enregistrer sous** .

Vous copiez les éléments de projet comme blocs dans le presse-papiers en choisissant la commande **Edition > Copier**.



La marche à suivre pour copier un projet est décrite étape par étape dans Copie d'un projet et Copie d'un élément de projet.

4.3.1 Application et enregistrement des modifications

Si vous éditez par exemple des instructions dans le programme utilisateur, une icône dans la fenêtre de projet vous signale que des modifications n'ont pas été encore enregistrées. Vous pouvez alors soit cliquer sur le bouton Appliquer, soit sur le bouton Enregistrer.

	symbolise que les modifications n'ont pas été encore appliquées, ni enregistrées (exemple)
---	--

Différence entre "Appliquer" et "Enregistrer"

	Appliquer	<p>Les modifications ont été provisoirement sauvegardées, garantissant ainsi la cohérence des données du projet et l'actualité des données affichées dans les fenêtres.</p> <p>Si vous fermez la fenêtre sans enregistrer, les modifications sont perdues car les données provisoirement sauvegardées sont effacées de la mémoire après la fermeture du projet.</p>
	Enregistrer	<p>Les modifications sont enregistrées dans le fichier du projet et sont prises en compte à l'ouverture suivante du projet.</p> <p>Lors de la création d'un nouveau filtre pour la table des mnémoniques ou pour la liste des références croisées, il est nécessaire de cliquer sur "Appliquer" avant d'enregistrer. Les filtres sont également enregistrés avec toutes les modifications du projet.</p>

Procédure recommandée

Cliquez sur "Appliquer" si l'état actuel du projet ne doit pas être écrasé (par exemple parce que les modifications effectuées ne sont pas encore définitives et peuvent être encore actualisées).

Cliquez sur "Enregistrer" si l'état enregistré du projet doit être actualisé. **Toutes** les modifications du projet sont enregistrées. Si vous sélectionnez "Enregistrer sous", l'état actuel du projet est enregistré sous un autre nom ou sous un autre chemin.

Particularités des modèles pour la documentation de projet

Les modèles pour la documentation du projet sont enregistrés séparément dans un fichier (*.k7d). Ils sont disponibles pour chaque projet via la fonction "Charger".

Les paramètres de la documentation du projet sont enregistrés en même temps que les autres données du projet.

Particularités de l'utilisation de filtres

Vous pouvez créer de nouveaux filtres pour la table des mnémoniques ou la liste des références croisées ou modifier un filtre déjà existant. Cliquez pour cela sur le bouton "Editer filtre" pour éditer dans le dialogue suivant la description et les paramètres du filtre.

Vous avez en dehors du bouton "Annuler" (La boîte de dialogue est quittée sans modification) les possibilités suivantes pour quitter la boîte de dialogue:

Clic sur le bouton "Appliquer"	Les paramètres du filtres sont enregistrés provisoirement de manière à ce que le filtre puisse être sélectionné dans la liste déroulante "Filtre" après que la boîte de dialogue a été quittée. Si vous enregistrez le projet, le filtre est automatiquement enregistré en même temps. Si vous n'avez pas sauvegardé le projet, le filtre ne sera plus disponible à la prochaine ouverture du projet.
Clic sur le bouton "Filtrer" (sans appliquer auparavant)	Les paramètres du filtre sont sauvegardés provisoirement de manière à ce que le filtre puisse être sélectionné dans la liste déroulante "Filtre" après que la boîte de dialogue a été quittée. Cependant le nom du filtre est marqué d'un astérisque (*). Cet astérisque indique que le filtre ne sera pas enregistré lors de l'enregistrement du projet ! Cette procédure est recommandée si vous n'avez plus besoin du filtre par la suite et qu'il ne doit donc plus figurer dans la liste déroulante "Filtre".

4.3.2 Détails sur l'édition

4.3.2.1 Copie d'un projet

Si vous voulez copier un projet, procédez de la manière suivante :

1. Choisissez la commande **Fichier > Enregistrer**.
2. Dans la boîte de dialogue "Enregistrer sous", saisissez le nom du nouveau projet et, le cas échéant, d'un nouveau chemin. Confirmez par "OK".

4.3.2.2 Copie d'un élément de projet

Si vous souhaitez copier dans un autre projet un élément de projet, comme par exemple des blocs, procédez de la manière suivante :

Copier avec des commandes de menu

1. Ouvrez STEP 7 Lite une seconde fois.
2. Ouvrez dans l'un des STEP 7 Lite le projet source et dans l'autre le projet cible.
3. Sélectionnez l'élément de projet à copier dans la fenêtre du projet source.
4. Dans le STEP 7 Lite de la source, choisissez la commande **Edition > Copier**.
5. Passez dans le STEP 7 Lite de la cible et choisissez-y la commande **Edition > Coller**.

Copier directement avec la souris (par glisser-lâcher)

1. Ouvrez STEP 7 Lite une seconde fois.
2. Ouvrez dans l'un des STEP 7 Lite le projet source et dans l'autre le projet cible.
3. Sélectionnez l'élément de projet à copier dans la fenêtre du projet source et maintenez le bouton gauche de la souris enfoncé.
4. Faites glisser l'objet dans la fenêtre du projet cible et lâchez-le à la position adéquate en relâchant le bouton de la souris.

Procédez dans le même ordre pour tous les éléments de projet à copier. Une fois la copie terminée, refermez le STEP 7 Lite du projet source.

4.3.2.3 Configuration du matériel (principe)

Procédez de la manière suivante:

1. Cliquez sur l'objet "Matériel" pour ouvrir la zone de travail permettant de configurer le matériel.
2. Sélectionnez le type de station (par ex. S7-300) pour configurer et paramétrer des modules dans la vue appropriée.
3. Sélectionnez des modules que vous répartissez sur les emplacements en respectant les règles relatives à ces emplacements.
4. Au besoin, donnez des valeurs aux paramètres des différents modules.

4.3.2.4 Création du logiciel dans le projet (principe)

Afin de créer le logiciel pour votre projet, procédez de la manière suivante :

1. Sélectionnez l'objet "Mnémoniques" et définissez-y les mnémoniques (cette étape peut être reportée à plus tard).
2. Insérez de nouveaux blocs (commande **Insertion > Bloc**) et éditez-les dans la zone de travail.
3. Double-cliquez sur le mnémonique pour documentation du projet dans la fenêtre de projet. Désignez dans la vue "Documentation du projet" les objets à imprimer et effectuez les paramétrages nécessaires à l'impression ou bien sélectionnez un modèle de documentation et imprimez le projet avec le bouton "Imprimer documentation".

4.4 Effacer et renommer un projet

Pour effacer ou renommer un projet, utilisez l'explorateur de fichiers de votre système d'exploitation. Votre projet se reconnaît à l'extension ".k7p".

Lorsque vous effacez ou renommez un projet, assurez-vous qu'il n'est pas ouvert dans STEP 7 Lite.

5 Configuration matérielle

5.1 Principes de la configuration matérielle avec STEP 7 Lite

5.1.1 Introduction à la configuration matérielle

Configuration

Par "configurer", on entend dans ce qui suit disposer des modules et des cartouches interface dans une vue graphique représentant la composition d'une station (par ex. S7-300).

Les profilés support sont représentés non seulement sous cette forme graphique, mais aussi par une table de configuration autorisant, comme le profilé support réel, un nombre déterminé de modules enfichables. La table de configuration donne plus de renseignements sur les modules (tels que la désignation exacte et l'adresse).

Vous pouvez copier la configuration dans un autre projet STEP 7 Lite, la modifier si besoin est et la charger dans une ou plusieurs stations existantes. A la mise en route de l'automate programmable, la CPU compare la configuration prévue, créée avec STEP 7 Lite, à la configuration sur site de l'installation. Ainsi, les erreurs éventuelles sont immédiatement détectées et signalées.

Paramétrage

Par "paramétrer", on entend dans ce qui suit définir les propriétés des modules paramétrables pour configuration centralisée.

Exemple : une CPU est un module paramétrable ; le temps de surveillance du cycle est un paramètre que vous pouvez régler.

Les paramètres sont chargés dans la CPU qui les transmet aux modules correspondants. Il est très facile de remplacer des modules, puisque les paramètres définis avec STEP 7 Lite sont automatiquement chargés dans le nouveau module à la mise en route.

Quand faut-il configurer le matériel ?

Les propriétés des automates programmables S7 et des modules étant prédéfinies, vous n'aurez pas besoin de procéder à une configuration dans bien des cas.

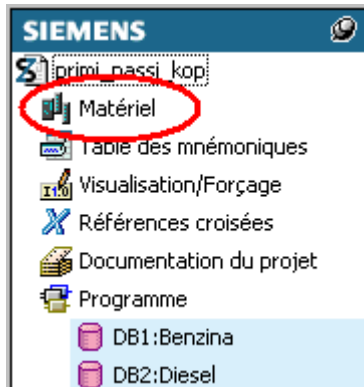
La configuration est impérativement nécessaire quand vous souhaitez modifier les paramètres prédéfinis d'un module (par ex. valider l'alarme de processus pour un module).

5.1.2 Manipulations de base pour la configuration matérielle

Démarrer la configuration matérielle

Lorsque vous avez créé un nouveau projet, ouvrez comme suit la zone de travail pour la configuration et le paramétrage de modules :

- double-clic sur l'icône "Matériel"



Zone de travail pour la configuration

La zone de travail permettant de configurer un système d'automatisation présente les parties suivantes :

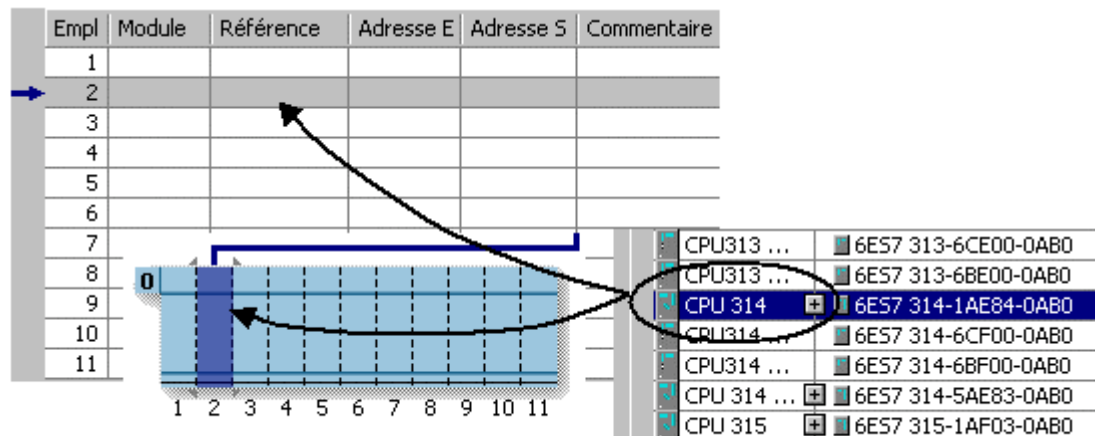
- une vue d'ensemble graphique montrant les châssis avec leurs modules,
- des tableaux représentant les différents châssis et contenant, à la différence de la vue d'ensemble graphique, plus d'informations sur les modules (n° de référence, adresses, etc.),
- le catalogue du matériel dans lequel vous allez sélectionner les composants matériels requis, tels que modules et cartouches interface.

5.1.2.1 Marche à suivre pour configurer une station

Quelle que soit la composition d'une station, il faut toujours vous en tenir aux étapes de configuration suivantes :

1. Sélectionnez un composant matériel dans le catalogue du matériel.
2. Amenez-le par glisser-lâcher
 - sur un emplacement du châssis dans la vue graphique
 - ou sur une ligne de la table de configuration représentant la composition du châssis

La figure suivante montre les manipulations de base :

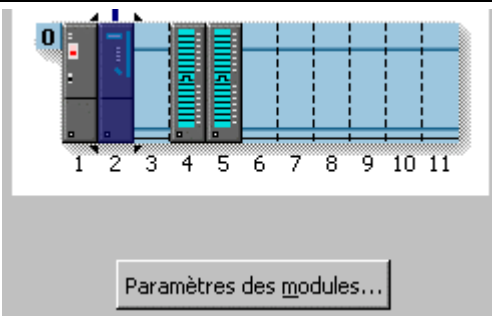


5.1.2.2 La vue 'Configuration HW'

La vue "Configuration HW" donne deux représentations de la configuration de station en cours.

- La vue graphique montre d'une manière très proche de la réalité la disposition des modules à leurs emplacements d'enchâssement.
- La table de configuration donne des renseignements détaillés sur les modules enchâssés (par ex. adresses et numéros de référence)

Cette vue montre en outre le catalogue des composants à partir duquel il faut équiper les profilés supports.

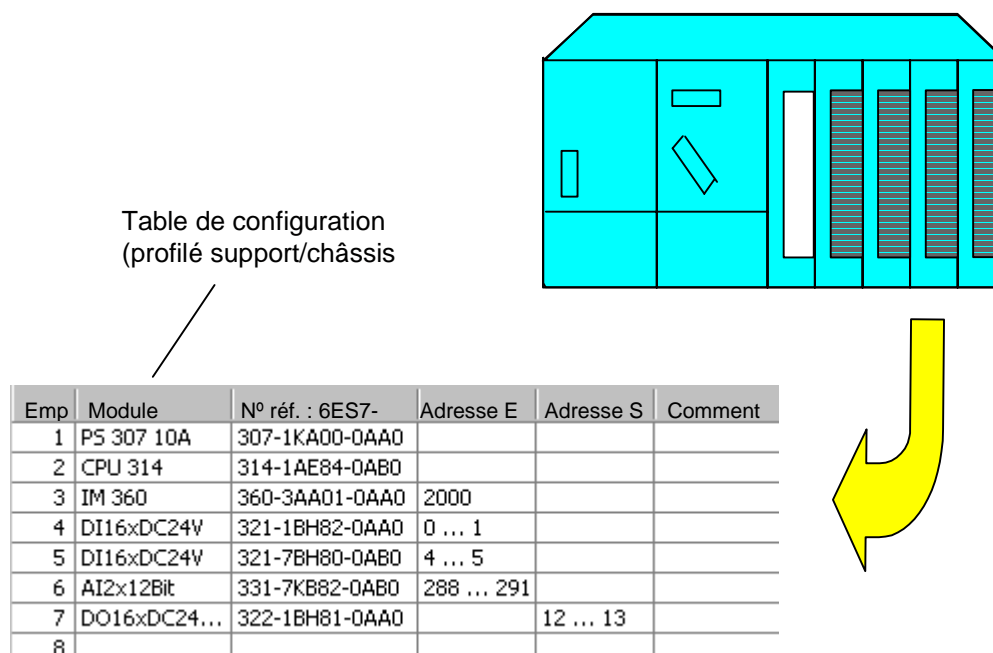
Zone de la vue 'Configuration HW'	Signification																																								
<div></div>	<p>Vue graphique de la configuration matérielle avec emplacement ou module sélectionné.</p> <p>Une flèche va du module sélectionné à l'emplacement de même valeur dans la table de configuration.</p> <p>Le bouton "Paramètres de module" sert à ouvrir la boîte de dialogue de paramétrage pour les modules paramétrables (par ex. la CPU dans le cas présent).</p>																																								
<div><table><tr><th>Châssis 0</th><th>Châssis 1</th><th>Châssis 2</th><th>Châssis 3</th></tr><tr><th>Empl</th><th>Module</th><th>Référence</th><th>Adresse E</th><th>Adresse S</th><th>Commentaire</th></tr><tr><td>1</td><td>PS 307...</td><td>6ES7 307-1BA00-...</td><td></td><td></td><td></td></tr><tr><td>2</td><td>CPU 315</td><td>6ES7 315-1AF03-...</td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>SM 32...</td><td>6ES7 321-1BH00-...</td><td>0 ... 1</td><td></td><td></td></tr><tr><td>5</td><td>SM 32...</td><td>6ES7 322-1BH00-...</td><td></td><td>4 ... 5</td><td></td></tr></table></div>	Châssis 0	Châssis 1	Châssis 2	Châssis 3	Empl	Module	Référence	Adresse E	Adresse S	Commentaire	1	PS 307...	6ES7 307-1BA00-...				2	CPU 315	6ES7 315-1AF03-...				3						4	SM 32...	6ES7 321-1BH00-...	0 ... 1			5	SM 32...	6ES7 322-1BH00-...		4 ... 5		<p>Emplacement ou module sélectionné dans la table de configuration.</p> <p>Les onglets placés au bord supérieur permettent d'accéder à différents profilés supports.</p>
Châssis 0	Châssis 1	Châssis 2	Châssis 3																																						
Empl	Module	Référence	Adresse E	Adresse S	Commentaire																																				
1	PS 307...	6ES7 307-1BA00-...																																							
2	CPU 315	6ES7 315-1AF03-...																																							
3																																									
4	SM 32...	6ES7 321-1BH00-...	0 ... 1																																						
5	SM 32...	6ES7 322-1BH00-...		4 ... 5																																					
<div><div><input type="checkbox"/> Afficher les informations modules</div><table><thead><tr><th>Module</th><th>Référence</th><th></th></tr></thead><tbody><tr><td>CPU 312 ...</td><td>6ES7 312-5AC82-0AB0</td><td>1.2</td></tr><tr><td>CPU 312 C</td><td>6ES7 312-5BD00-0AB0</td><td></td></tr><tr><td>CPU 313</td><td>6ES7 313-1AD03-0AB0</td><td>1.2</td></tr><tr><td>CPU 313 C</td><td>6ES7 313-5BE00-0AB0</td><td></td></tr><tr><td>CPU 313 ...</td><td>6ES7 313-6CE00-0AB0</td><td></td></tr><tr><td>CPU 313 ...</td><td>6ES7 313-6BE00-0AB0</td><td></td></tr><tr><td>CPU 314</td><td>6ES7 314-1AE84-0AB0</td><td>1.2</td></tr><tr><td>CPU 314 ...</td><td>6ES7 314-6CF00-0AB0</td><td></td></tr><tr><td>CPU 314 ...</td><td>6ES7 314-6BF00-0AB0</td><td></td></tr><tr><td>CPU 314</td><td>6ES7 314-6AE00-0AB0</td><td>1.2</td></tr></tbody></table><div><div>Système</div><div>PS</div><div>CPU</div><div>IM</div><div>SM AI</div><div>SM AO</div><div>SM AI/O</div><div>SM DI</div><div>SM DO</div></div></div>	Module	Référence		CPU 312 ...	6ES7 312-5AC82-0AB0	1.2	CPU 312 C	6ES7 312-5BD00-0AB0		CPU 313	6ES7 313-1AD03-0AB0	1.2	CPU 313 C	6ES7 313-5BE00-0AB0		CPU 313 ...	6ES7 313-6CE00-0AB0		CPU 313 ...	6ES7 313-6BE00-0AB0		CPU 314	6ES7 314-1AE84-0AB0	1.2	CPU 314 ...	6ES7 314-6CF00-0AB0		CPU 314 ...	6ES7 314-6BF00-0AB0		CPU 314	6ES7 314-6AE00-0AB0	1.2	<p>Catalogue avec liste de modules.</p> <p>Cliquez sur le symbole "+" placé devant le n° de référence pour afficher une liste de modules de même type, mais de n° de référence différent. Le module en cours est toujours en tête de liste.</p> <p>Quand il y a plusieurs versions du système d'exploitation (firmware) d'un module, une liste déroulante permet de sélectionner une version déterminée.</p> <p>Les onglets situés sur le côté droit du catalogue permettent d'accéder à différentes catégories de modules.</p>							
Module	Référence																																								
CPU 312 ...	6ES7 312-5AC82-0AB0	1.2																																							
CPU 312 C	6ES7 312-5BD00-0AB0																																								
CPU 313	6ES7 313-1AD03-0AB0	1.2																																							
CPU 313 C	6ES7 313-5BE00-0AB0																																								
CPU 313 ...	6ES7 313-6CE00-0AB0																																								
CPU 313 ...	6ES7 313-6BE00-0AB0																																								
CPU 314	6ES7 314-1AE84-0AB0	1.2																																							
CPU 314 ...	6ES7 314-6CF00-0AB0																																								
CPU 314 ...	6ES7 314-6BF00-0AB0																																								
CPU 314	6ES7 314-6AE00-0AB0	1.2																																							

5.1.2.3 Table de configuration comme reproduction d'un profilé support ou châssis

En configuration centralisée, vous montez les modules à côté de la CPU sur un profilé support ou un châssis, puis sur d'autres profilés support ou châssis. Le nombre des profilés support ou châssis autorisés dépend de la CPU utilisée.

Tout comme dans votre installation réelle, vous montez les modules dans des profilés support ou des châssis avec STEP 7 Lite. La différence, c'est que les profilés support ou châssis sont représentés en plus par des "tables de configuration" dont le nombre de lignes correspond au nombre de modules enfichables sur le profilé support ou châssis réel.

La figure suivante montre, à l'aide d'un exemple, comment transposer une configuration réelle dans une table de configuration. La table de configuration correspond au profilé support ou châssis utilisé.



5.1.2.4 Définir les propriétés de composants

Lorsque vous avez disposé des composants dans la vue "Configuration HW", vous pouvez en modifier les propriétés (paramètres) par défaut dans une boîte de dialogue que vous appelez toujours de la manière suivante :

- Effectuez un double clic sur le composant ou choisissez la commande **Edition > Paramètres du module**.
- Avec le bouton droit de la souris : positionnez le curseur sur le composant, cliquez avec le bouton droit de la souris et choisissez la commande **Paramètres du module** dans le menu contextuel.

Propriétés des unités centrales

Les propriétés des CPU sont de première importance pour le comportement du système. Dans la boîte de dialogue de paramétrage d'une CPU, vous pouvez définir, par exemple, le comportement à la mise en route, les zones de mémoire, le comportement de rémanence, le memento de cadence, le niveau de protection et le mot de passe, pour ne citer que quelques propriétés. STEP 7 Lite "sait" ce que vous pouvez définir et dans quelles plages de valeurs.

Autres méthodes de paramétrage

Il est possible de définir les paramètres de certains modules dans le programme utilisateur (par ex. pour les modules analogiques). Pour cela, vous appelez les fonctions système (SFC) WR_PARM, WR_DPARM et PARM_MOD dans le programme utilisateur. Ces paramètres se perdront toutefois à la mise en route (démarrage à chaud).

Vous trouverez des informations détaillées sur les fonctions système dans le manuel de référence *Logiciel système pour SIMATIC S7-300/400, Fonctions standard et fonctions système*.

5.1.2.5 Informations sur les règles d'enfichage et autres règles

STEP 7 Lite vous assiste lors de la configuration d'une station, si bien qu'en général un message vous signale immédiatement si, par exemple, un module ne peut pas être enfiché à l'emplacement souhaité.

Soyez attentif aux messages qui s'affichent, car ils renseignent sur les causes et les conséquences d'une manipulation. L'aide sur ces messages vous permet d'obtenir des informations complémentaires.

Il n'est pas tenu compte des règles supplémentaires temporaires (valables pour une version donnée), comme par exemple les restrictions quant aux emplacements d'enfichage utilisables résultant d'une restriction fonctionnelle pour certains modules. Vous devez donc toujours prendre connaissance de la documentation ou de l'information produit actuelle relatives aux modules.

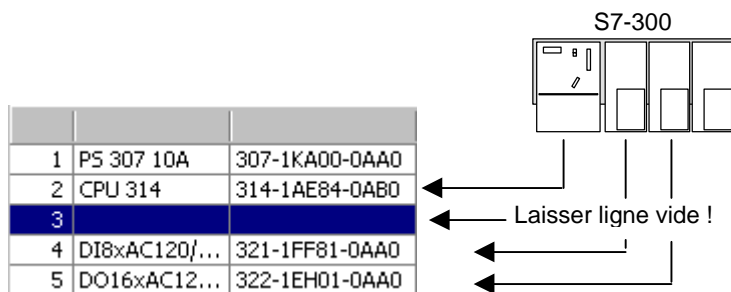
5.2 Configuration des modules

5.2.1 Règles pour disposer des modules (SIMATIC 300)

Règle de base

Vous disposez les modules dans le châssis en veillant à ne pas laisser d'emplacement vide.

Exception : dans une configuration comportant un profilé support, vous devez laisser un emplacement d'enchâssement de la table de configuration vide (réservé pour le module de couplage). Pour S7-300, il s'agit de l'emplacement d'enchâssement 3, pour M7-300 de l'emplacement venant immédiatement après le groupe de modules (CPU accompagnée de ses modules d'extension). Il pourra donc s'agir dans ce cas de l'emplacement 3, 4, 5 ou 6). Dans la configuration réelle, il ne doit pas y avoir d'emplacement vide, sans quoi le bus interne serait interrompu !



Règles d'enchâssement (S7-300)

Profilé support 0 :

- Emplacement 1 : uniquement alimentation (par ex. 6ES7 307-...) ou vide
- Emplacement 2 : uniquement unité centrale (par ex. 6ES7 314-...)
- Emplacement 3 : module de couplage (par ex. 6ES7 360-.../361-...) ou vide
- Emplacements 4 à 11 : modules de signaux ou de fonction, processeurs de communication ou vide

Profilés support 1 à 3 :

- Emplacement 1 : uniquement module d'alimentation (par ex. 6ES7 307-...) ou vide
- Emplacement 2 : vide
- Emplacement 3 : module de couplage

Emplacements 4 à 11 : modules de signaux ou de fonction, processeurs de communication (en fonction du module de couplage utilisé) ou vide

5.2.1.1 Règles particulières pour le module fictif DM 370 Dummy

Vous pouvez enficher le module fictif DM 370 Dummy pour réserver un emplacement destiné à un autre module à mettre en œuvre ultérieurement.

Selon la position de son commutateur, ce module réserve de l'espace mémoire pour un module (par ex. pour un module de sortie TOR) ou pas (par ex. pour un module de couplage).

Position du commutateur sur DM 370 Dummy	Signification	Numéro de référence
A	Espace mémoire réservable	6ES7 370-0AA01-0AA0
NA	Aucun espace mémoire réservé	Néant (le module est "invisible", il n'est pas nécessaire de le configurer)

5.2.1.2 Règles particulières pour le module de simulation TOR SIM 374 IN/OUT 16

Le module de simulation TOR SIM 374 IN/OUT 16 vous permet de simuler des entrées et sorties TOR.

Il ne vous est **pas** proposé dans la fenêtre "Catalogue du matériel" ! Dans la table de configuration, vous devez placer le module à simuler !

Position du commutateur sur le module SIM 374 IN/OUT 16	Module à disposer
16xOutput	6ES7322-1BH00-0AA0
8xOutput 8xInput	6ES7323-1BH00-0AA0
16xInput	6ES7321-1BH00-0AA0

5.2.2 Règles pour disposer des modules ET 200S et ET 200X

5.2.2.1 Règles pour disposer des modules ET 200S

La configuration maximale de l'appareil DP comporte 64 modules (IM 151/CPU comprise). Les modules doivent être enfichés sans intervalle.

L'appareil DP ET 200S débute par une IM 151/CPU.

Un Powermodul vient à la suite du module d'interface ou au début de chaque groupe de potentiel.

Un Powermodul est suivi de modules TOR ou analogiques.

L'appareil DP ET 200S se termine par le module de clôture qui n'est pas configuré.

Particularités du paramétrage de soudures froides

Respectez l'ordre suivant :

1. Placez le module électronique analogique et réglez une voie sur la plage de mesure "RTD-4L Pt 100 Kl." pour la fonction de soudure froide.
2. Cliquez deux fois sur l'IM 151/CPU et indiquez la ou les soudures froides dans la zone "Paramètres de module" de la boîte de dialogue de paramétrage : emplacement et voie du module RTD. N'oubliez pas d'activer la soudure froide.
3. Placez le module électronique analogique pour mesure de température par thermocouple (module TC) et paramétrez-le avec le numéro de la soudure froide (du module RTD).

5.2.2.2 Règles pour disposer des modules ET 200X

Une station ET 200X se compose d'un module de base (BM 147) et de 7 modules d'extension au plus (EM). Les modules doivent être enfichés sans intervalle.

Powermodule : il peut y avoir au plus 7 modules d'extension Powermodule PM 148 DO 4 x DC 24V/2A.

Départs-moteurs : 6 départs-moteurs au plus (EM 300...) peuvent être enfichés à n'importe quels emplacements de l'ET 200X.

Pneumatic-Interface-Module : 1 Pneumatic-Interface-Modul (EM 148-P DO 16 P/CPV...) au plus peut être connecté en tant que dernier module d'extension dans la configuration ET 200X.

La configuration maximale dépend de la consommation en courant des différents modules. Le manuel consacré au système DP ET 200X énumère toutes les configurations possibles avec leurs limites. Il expose aussi les moyens de repousser la limite.

5.2.3 Marche à suivre pour la configuration des modules

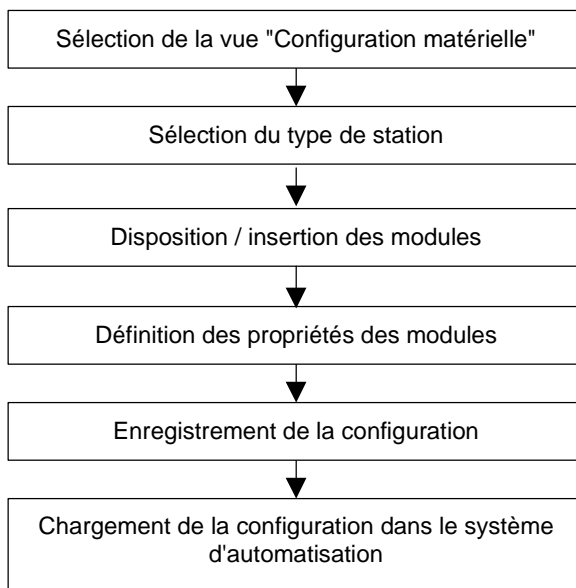
5.2.3.1 Configuration et paramétrage d'une station

Condition préalable

Vous avez ouvert ou créé un projet.

Marche à suivre

Pour configurer et paramétrer une station, vous devez procéder aux étapes suivantes :



5.2.3.2 Sélection d'un type de station

Lorsque vous avez choisi la vue "Configuration matérielle" et que vous ouvrez la configuration matérielle d'une station pour la première fois, vous devez sélectionner un type de station.

A cet effet, il faut utiliser l'onglet supérieur du catalogue : il est déjà ouvert et le type de station S7-300 y est sélectionné par défaut.


Si vous sélectionnez un autre type, la zone de travail s'adaptera automatiquement au type de station en question ; par exemple, le nombre maximal de modules pouvant être enfichés changera.

5.2.3.3 Disposer des modules dans le profilé support/châssis

La configuration matérielle est ouverte et la fenêtre est organisée de telle façon que le profilé support/châssis (vue graphique ou tableau) et le catalogue du matériel soient visibles.

Marche à suivre

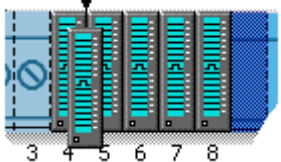
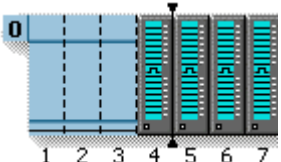
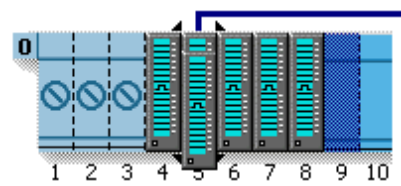
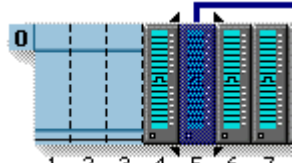
1. Sélectionnez un module (par ex. une CPU) dans le catalogue du matériel.
2. Amenez le module dans la ligne appropriée du profilé support/châssis (table de configuration) en utilisant la fonction glisser-lâcher. STEP 7 Lite vérifie le respect des règles d'enchâssement (une CPU S7-300, par exemple, ne doit être enfichée qu'à l'emplacement 2).

	Icône indiquant la violation des règles d'enchâssement
---	--

3. Répétez les étapes 1 et 2 pour enficher tous les modules souhaités dans le châssis.

Une solution de rechange consisterait à sélectionner la ou les lignes appropriées dans la table de configuration, puis à effectuer un double clic sur le module souhaité dans le catalogue du matériel. Si vous avez sélectionné plusieurs lignes, le même module sera enfiché à chacun des emplacements.

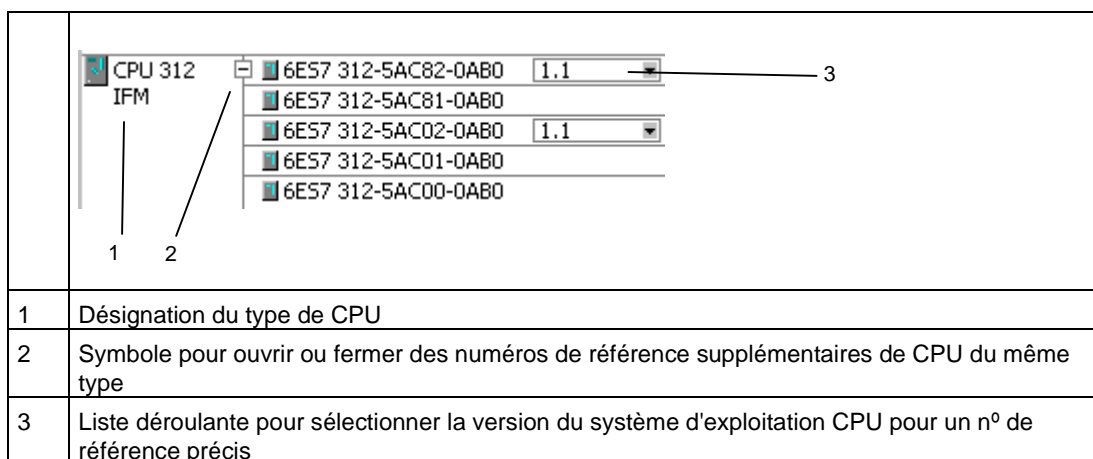
Particularités dans la vue graphique

Affichage dans la vue graphique	Signification
	Vous avez positionné un module par glisser-lâcher entre deux emplacements occupés. Si vous le lâchez maintenant, il sera enfiché à l'emplacement 5 et les modules situés à droite seront déplacés automatiquement d'un emplacement vers la droite.
	Vous avez sélectionné l'intervalle entre deux modules. Si vous effectuez maintenant un double-clic sur un module enfichable du catalogue, il sera enfiché à l'emplacement 5 et les modules situés à droite seront déplacés automatiquement d'un emplacement vers la droite.
	Vous avez positionné un module par glisser-lâcher à un emplacement occupé. Si vous le lâchez maintenant, il sera enfiché à l'emplacement 5 et celui qui se trouve "au-dessous" sera supprimé ("remplacement de module"). Le paramétrage (par ex. pour une CPU) sera repris par le "nouveau" module.
	Vous avez sélectionné un emplacement occupé. Si vous effectuez maintenant un double-clic sur un module enfichable du catalogue, il sera enfiché à l'emplacement 5 et le module enfiché à l'origine sera supprimé ("remplacement de module").

5.2.3.4 Affichage de la version du système d'exploitation CPU dans la liste des modules

Quand il existe plusieurs versions du système d'exploitation d'une CPU, vous devez en sélectionner une avec son numéro de référence dans la liste déroulante apparaissant à côté du nom de la CPU.

Vérifiez quelle est la version du système d'exploitation de la CPU que vous utilisez et sélectionnez celle-la.



5.2.3.5 Disposer des systèmes intégrés compacts C7 (particularités)

Dans un système intégré compact C7 (par ex. C7-620), les composants suivants sont intégrés dans un même boîtier :

- CPU SIMATIC 300,
- entrées et sorties (TOR et analogiques),
- module de couplage IM 360 pour le couplage d'autres modules SIMATIC 300,
- pupitre opérateur (OP) orienté ligne avec interface d'imprimante.

Pour les CPU de la série C7-621, vous avez la possibilité d'enficher en plus des modules de signaux de la série S7-300 dans le profilé support 0. Pour cela, il faut enficher le module d'extension 6ES7 621-1AD00-6AE3 à l'emplacement 3 du profilé support 0.

Marche à suivre

1. Sélectionnez un système intégré compact C7 dans le catalogue du matériel. Ces systèmes se trouvent dans l'onglet "C7" (système S7-300).
2. Amenez le système intégré compact C7 dans la fenêtre de station en utilisant la fonction glisser-lâcher.
Les entrées et sorties intégrées "se répartissent" automatiquement sur les emplacements à côté de la CPU.
3. Si vous souhaitez étendre le système complet intégré C7 :
Affectez des modules au profilé support. Important : pour que le couplage soit possible, il faut que les modules de couplage soient enfichés dans tous les profilés support/châssis !

5.2.3.6 Définir les propriétés de modules/interfaces

Introduction

Par la suite, les propriétés de composants (par ex. les modules ou les interfaces) désigneront les adresses et les paramètres. Les paragraphes suivants ne sont à lire que si vous souhaitez modifier les valeurs par défaut.

Condition préalable

Vous avez disposé le composant dont vous souhaitez modifier les propriétés dans la table de configuration.

Marche à suivre

Chaque composant (module, interface) possède des propriétés prédéfinies, par exemple les types et plages de mesure dans le cas de modules d'entrées analogiques.

Pour modifier les valeurs par défaut, procédez de la manière suivante :

1. Dans la table de configuration, effectuez un double clic sur le composant à paramétrer (par ex. un module) ou sélectionnez la ligne correspondante, puis choisissez la commande **Edition > Paramètres de module**.
Autres solutions :
avec le bouton droit de la souris : positionnez le pointeur sur le composant, cliquez avec le bouton droit de la souris et choisissez la commande **Paramètres de module** dans le menu contextuel ;
avec le bouton "Paramètres de module" : positionnez le pointeur sur le composant et cliquez sur le bouton "Paramètres de module".
2. Définissez les propriétés du composant à l'aide des pages d'onglet affichées.
La liste figurant dans la partie gauche facilite la navigation jusqu'aux paramètres individuels.

Particularités des CPU possédant des entrées et sorties intégrées

Les CPU possédant des entrées et sorties intégrées, comme p. ex. la CPU 31x C ("CPU compactes") disposent d'un bouton "Détails" dans la colonne "Adresse d'entrée". Lorsque vous cliquez sur ce bouton, des lignes supplémentaires s'affichent avec les adresses des entrées et sorties intégrées. Un double clic sur l'une de ces lignes ouvre également une boîte de dialogue de paramétrage, de même qu'un double clic sur la ligne dans laquelle la CPU est enfichée.

5.2.3.7 Attribution des adresses

Pour l'attribution des adresses, il faut distinguer les adresses de réseau et les adresses d'entrée/sortie (adresses de périphérie).

Adresses de réseau

Les adresses de réseau sont celles des interfaces de modules (adresses MPI et PROFIBUS) ; elles sont requises pour adresser les divers participants au sous-réseau, par exemple une IM 151/CPU (ET 200S) sur un sous-réseau PROFIBUS. Vous pouvez attribuer les adresses en configurant le matériel (paramètres d'interface). L'adresse de réseau de la CPU à laquelle la PG est connectée est affichée entre crochets dans la barre de titre du panneau de commande de la CPU.

L'adresse de réseau est conservée même après un effacement général de la CPU.

Remarque sur le réglage des adresses PROFIBUS des CPU d'ET 200S et d'ET 200X :

L'adresse saisie lors du paramétrage doit concorder avec celle déterminée par les commutateurs DIP de la CPU, sinon cette dernière ne se mettra pas en route !

Adresses d'entrée/sortie

Les adresses d'entrée/sortie (adresses de périphérie) sont requises pour lire des entrées ou mettre des sorties à 1 dans le programme utilisateur.

5.2.3.8 Attribution des adresses d'entrée/sortie

STEP 7 Lite attribue déjà des adresses d'entrée et de sortie lorsque vous disposez les modules dans la table de configuration. Ainsi, chaque module possède son adresse de début (adresse de la première voie) dont découlent les adresses des autres voies. Il n'est pas possible de modifier ces adresses ; à chaque emplacement d'enchâssement est attribuée une adresse de début fixe :

Emplacement 4 : adresse 0 (module TOR) ou adresse 256 (module analogique)

Emplacement 5 : adresse 4 (module TOR) ou adresse 272 (module analogique)

etc...

5.2.3.9 Conseils pour éditer une configuration de station

Déplacer des modules

Utilisez la méthode de glisser-lâcher pour amener des modules ou autres composants à un autre emplacement autorisé au sein de la station.

Remplacer des modules

Si vous avez déjà créé une configuration et que vous souhaitez remplacer un module par un autre module du catalogue du matériel, procédez de la manière suivante :

1. Amenez le nouveau module (par ex. une CPU) par glisser-lâcher sur l'emplacement du module à remplacer.
 - Lorsque le module doit être remplacé par un module "compatible" et est déjà paramétré (p. ex. CPU ou module analogique), alors le paramétrage est repris.
 - Lorsque le nouveau module n'est pas en mesure de reprendre tous les paramètres, un message vous en informe et, le cas échéant, vous pouvez annuler la procédure.
 - Lorsque le nouveau module est totalement différent du module à remplacer, ce dernier est effacé après demande de confirmation, puis remplacé par le nouveau.
2. Dans la boîte de dialogue qui s'ouvre, confirmez le remplacement du module, le cas échéant.

Le paramétrage est uniquement repris pour des modules "compatibles". Lorsque les modules ne le sont pas, l'"ancien" module est supprimé et le nouveau module est enfiché lors du remplacement. Dans ce cas, vous devez reparamétrer le nouveau module.

Exemple : vous pouvez remplacer une CPU paramétrée par une autre CPU possédant un numéro de référence différent ; le paramétrage entier (par ex. l'adresse MPI) sera repris par le nouveau module.

Les modules compatibles avec un module enfiché et sélectionné sont affichés dans l'onglet "Compatible" du catalogue du matériel.

Conseil

Vous avez toujours la possibilité d'annuler le remplacement en choisissant la commande de menu **Edition > Annuler**.

Sélectionner plusieurs lignes dans la table de configuration

Procédez de la manière suivante pour sélectionner plusieurs lignes dans une table de configuration afin de copier ou d'effacer plusieurs modules :

Sélectionner toutes les lignes	Choisissez la commande Edition > Sélectionner tout
Sélectionner un groupe de lignes	Cliquez sur la première ligne du groupe. Maintenez la touche MAJ enfoncée et cliquez sur la dernière ligne du groupe.
Sélectionner plusieurs lignes	Maintenez la touche CTRL enfoncée et cliquez sur chacune des lignes que vous souhaitez sélectionner.

5.2.4 Ce qu'il faut savoir sur démarreurs de moteur ET 200S (High Feature)

5.2.4.1 Reconnaissance des états de l'installation à l'aide des valeurs du courant

Le courant du moteur et les seuils de courant permettent de reconnaître les différents états de l'installation :

Etat de l'installation	Valeur du courant	Protection par :
Marche forcée de l'installation, par exemple à cause d'une avarie de stock	Le courant est supérieur à la normale	Seuils de courant
Marche légère de l'installation, par exemple parce que les matières premières manquent	Le courant est inférieur à la normale	Seuils de courant
L'installation est bloquée.	Un courant très fort circule.	Dispositif antibloqueur (par seuil de courant et seuil de temps)
Le moteur tourne à vide, en raison d'une avarie de l'installation ou parce que le moteur n'est pas connecté (mise en service)	Un courant très faible circule (< 18,75 % du courant d'emploi assigné)	Détection du courant résiduel

5.2.4.2 Courant à l'état bloqué

En cas de dépassement du courant à l'état bloqué, le démarreur de moteur détecte un blocage. La surveillance du temps de blocage est alors déclenchée. Sa durée est déterminée par le paramètre de même nom - quelle que soit la catégorie d'arrêt - .

Nota

Le démarreur de moteur est désactivé une fois le temps de blocage écoulé et si le blocage n'a pas disparu.

Plage de réglage : Réglage fixe de 800% du courant d'emploi assigné.

5.2.4.3 Temps de blocage

Temps que peut durer un blocage sans désactivation. Le démarreur de moteur est désactivé une fois le temps de blocage écoulé et si le blocage n'a pas disparu.

Plage de réglage : réglage fixe de 1 seconde .

Nota

Le dispositif anti-bloqueur est actif après le démarrage !

5.2.4.4 Réaction en cas de détection du courant résiduel

La détection du courant résiduel s'enclenche quand le courant du moteur est inférieur dans les trois phases à 18,75 % du courant d'emploi assigné.

Ces paramètres de l'appareil déterminent la réaction du démarreur de moteur en cas de détection de courant résiduel :

- Avertissement (se comporte comme pour "Avertissement groupé")
- Désactivation (se comporte comme "Coupure sans redémarrage")

Nota

A l'activation du moteur, la détection du courant résiduel est inhibée durant environ 1 seconde !

5.2.4.5 Déséquilibre

Les moteurs triphasés asynchrones réagissent à des déséquilibres minimes dans la tension du secteur par une consommation asymétrique plus forte. Ceci entraîne une élévation de la température des bobinages statoriques et rotoriques.

Nota

A l'activation du moteur, l'évaluation du déséquilibre est inhibée pendant environ 500 millisecondes.

5.2.4.6 Modèle de moteur thermique

Les courants moteur mesurés, le courant d'emploi assigné et la catégorie d'arrêt permettent de calculer la température du bobinage du moteur d'après le modèle de moteur thermique. Ceci permet de voir s'il y a surcharge du moteur ou si ce dernier fonctionne dans la plage normale de service.

5.2.4.7 Temps de récupération

Le temps de récupération est le temps prévu pour l'acquiescement du déclenchement de surcharge. Les chutes de tension sont également à prendre en compte dans le calcul du temps de récupération. Le temps de récupération après le déclenchement d'une surcharge est d'environ 90 secondes.

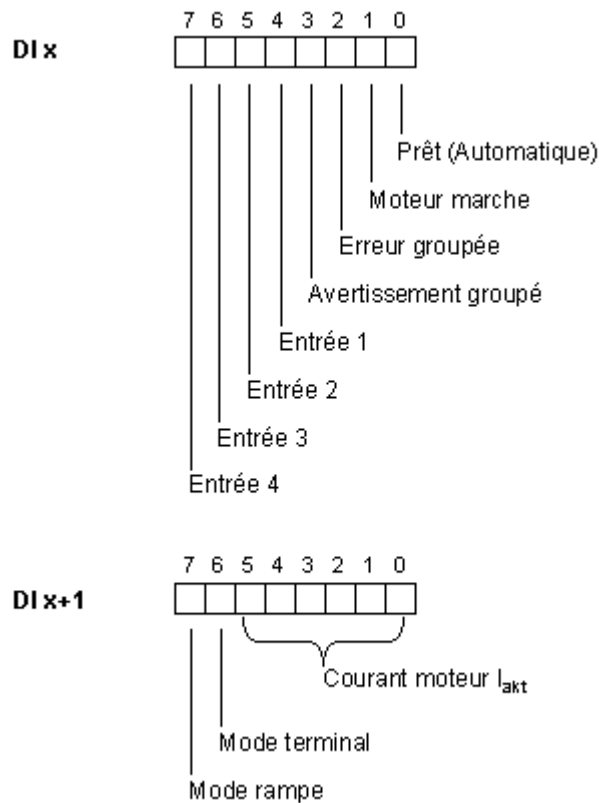
5.2.4.8 Tableau : Actions possibles du démarreur de moteur

Action	Commentaire
Aucune action	Les paramètres "Entrée n - niveau" sont activés dans la mémoire image (voir octet 1). Les paramètres "Entrée n - niveau" n'entraînent aucune réaction du démarreur de moteur.
Désactivation sans redémarrage	Entraîne la désactivation du moteur et de la sortie du frein. Le bit "Erreur groupée" est mis à 1 dans la mémoire image. Acquittement nécessaire après la suppression de la cause de l'erreur (via la mémoire image ou le commutateur rotatif sur l'appareil).
Désactivation avec redémarrage (Autoreset) Nota : Un redémarrage signifie que le démarreur est redémarré par une commande de démarrage une fois la cause de l'erreur supprimée (Autoreset).	Entraîne la désactivation du moteur et de la sortie du frein. Le bit "Erreur groupée" est mis à 1 dans la mémoire image. Acquittement automatique après la suppression de la cause de l'arrêt.
Désactivation Fin de course Marche à droite / Désactivation Fin de course Marche à gauche	Désactivation du moteur et de la sortie du frein quel que soit le sens de marche. Le bit "Erreur groupée" est mis à 1 dans la mémoire image. La sortie du frein (DO 0.2) ne peut être activée de nouveau que si Moteur-DROITE (DO 0.0) et Moteur-GAUCHE (DO 0.1) et la sortie du frein (DO 0.2) ont été remis à 0. Désactivation Fin de course Marche à droite : Le moteur ne peut être de nouveau activé qu'avec la commande inverse "Moteur-GAUCHE" (DO 0.1). Désactivation Fin de course Marche à gauche : Le moteur ne peut être de nouveau activé qu'avec la commande inverse Moteur-DROITE" (DO 0.0).
Avertissement groupé (avertissement)	Le bit "Avertissement groupé" est mis à 1 dans la mémoire image. Le démarreur de moteur et la sortie du frein ne sont pas désactivés
Mode terminal	Commande uniquement possible via "Entrée n - action : Moteur-DROITE et Moteur-GAUCHE" (voir plus bas) ! La commande via bus de terrain (mode automatique) n'est pas possible ! Le mode automatique n'est de nouveau possible que si le mode terminal a été mis à 0 et que le paramètre "Entrée n - action : Moteur-DROITE ou Moteur-GAUCHE" n'est pas activé.
Démarrage de secours	Mise en route du moteur même si la cause de l'erreur n'a pas été supprimée s'il y a une commande d'activation. Activation de la sortie du frein s'il y a une commande d'activation. Uniquement contact à fermeture autorisé
Moteur-DROITE / Moteur-GAUCHE	Pour ces actions le démarreur doit se trouver en mode terminal. Moteur-DROITE : Activation et désactivation des moteur et sortie du frein (marche à droite). Moteur-GAUCHE : Activation et désactivation des moteur et sortie du frein (marche à gauche). Uniquement contact à fermeture autorisé

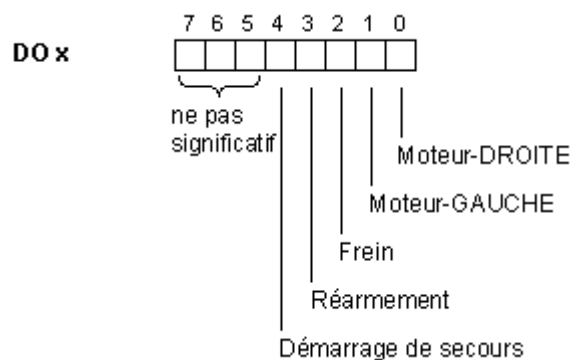
5.2.4.9 Affectation du démarreur de moteur dans la mémoire image

Les tableaux ci-après illustrent l'affectation du démarreur de moteur dans la mémoire image si l'adresse de début du démarreur de moteur est x.

Mémoire image des entrées



Mémoire image des sorties



5.3 Enregistrement d'une configuration et vérification de cohérence

Pour enregistrer une configuration avec tous les paramètres et adresses qu'elle détermine, choisissez la commande **Fichier > Enregistrer**. Même des configurations incohérentes (erronées) peuvent être enregistrées de cette façon.

Avant de la charger, il convient de contrôler la configuration avec la commande **Outils > Vérifier la cohérence** et d'y éliminer les erreurs signalées.

6 Programmation de blocs

6.1 Définition de mnémoniques

6.1.1 Adressage absolu et adressage symbolique

Dans un programme STEP 7 Lite, vous utilisez des opérandes comme des signaux d'E/S, des mémentos, des compteurs, des temporisations, des blocs de données et des blocs fonctionnels. Vous pouvez accéder à ces opérandes par adressage absolu dans votre programme. Toutefois, la lisibilité de vos programmes sera grandement améliorée si vous faites plutôt appel à des mnémoniques (par ex. Moteur_A_marche ou désignations usuelles dans le système d'identification de votre secteur d'activité). Il est alors possible d'accéder aux opérandes de votre programme utilisateur via ces mnémoniques.

Adresse absolue

Une adresse absolue est composée d'un identificateur d'opérande et d'une adresse (par ex. A 4.0, E 1.1, M 2.0, FB21).

Adressage symbolique

Vous pouvez structurer votre programme de manière plus lisible et faciliter ainsi la correction d'erreurs en affectant des noms symboliques (mnémoniques) aux adresses absolues.

STEP 7 Lite est en mesure de convertir automatiquement les mnémoniques en adresses absolues. Si vous préférez adresser des ARRAY, STRUCT, blocs de données, données locales, blocs de code et types de données utilisateur de manière symbolique, vous devez commencer par affecter des mnémoniques aux adresses absolues, avant de pouvoir utiliser l'adressage symbolique.

Vous pouvez, par exemple, affecter le mnémonique Moteur_marche à l'opérande A 4.0, puis utiliser Moteur_marche comme adresse dans une instruction du programme. Les mnémoniques vous permettent de reconnaître plus aisément la correspondance des éléments du programme aux composants de votre projet de commande du processus.

Nota

Dans un mnémonique (désignation de variable), deux caractères de soulignement successifs ne sont pas autorisés (ex. : Moteur__marche).

Assistance lors de la saisie d'un programme

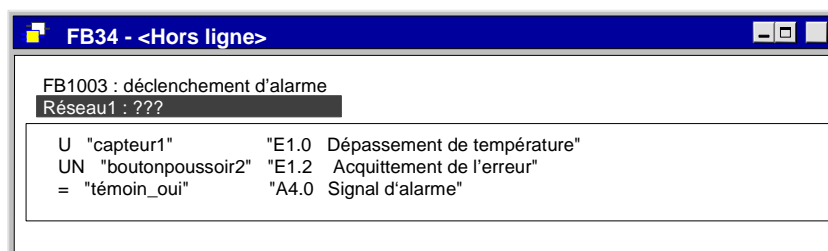
Dans les langages de programmation CONT, LOG et LIST, vous pouvez saisir les adresses, paramètres et noms de bloc de manière absolue ou symbolique.

La commande **Affichage > Afficher avec > Représentation symbolique** permet d'aller et venir entre l'affichage de l'adressage absolu et celui de l'adressage symbolique.

Pour faciliter la programmation avec adressage symbolique, vous pouvez afficher les adresses absolues et les commentaires correspondant aux mnémoniques utilisés.

Choisissez à cet effet la commande **Affichage > Afficher avec > Informations mnémonique**. La ligne de commentaire après chaque instruction LIST est alors remplacée en conséquence. Vous ne pouvez pas éditer cet affichage ; vous devez effectuer les modifications dans la table des mnémoniques ou dans la table de déclaration des variables.

La figure suivante montre des informations mnémonique dans le langage LIST.



Lorsque le bloc est imprimé, c'est la représentation en cours à l'écran qui est imprimée avec le commentaire d'instruction ou le commentaire de mnémonique.

6.1.2 Mnémoniques globaux et mnémoniques locaux

Un mnémonique (nom symbolique) vous permet d'utiliser des désignations parlantes à la place d'adresses absolues. En combinant l'usage de mnémoniques courts et de commentaires explicites, vous répondez à la fois aux besoins d'une programmation concise et d'une programmation bien documentée.

L'on distingue les mnémoniques locaux des mnémoniques globaux.

	Mnémoniques globaux	Mnémoniques locaux
Domaine de validité	<ul style="list-style-type: none"> Ils sont valables dans l'ensemble du programme utilisateur. Ils peuvent être utilisés par tous les blocs. Leur signification est la même dans tous les blocs. Leur nom doit être univoque dans l'ensemble du programme utilisateur. 	<ul style="list-style-type: none"> Ils sont connus uniquement dans le bloc dans lequel ils ont été définis. Vous pouvez utiliser le même nom dans différents blocs à des fins différentes.
Caractères autorisés	<ul style="list-style-type: none"> Lettres, chiffres, caractères spéciaux. Lettres accentuées à l'exclusion de 0x00, 0xFF et des guillemets. Lorsque vous utilisez des caractères spéciaux dans un mnémonique, ce dernier doit être placé entre guillemets. 	<ul style="list-style-type: none"> Lettres Chiffres Caractère de soulignement (_)

	Mnémoniques globaux	Mnémoniques locaux
Utilisation	<p>Vous pouvez définir des mnémoniques globaux pour :</p> <ul style="list-style-type: none"> entrées/sorties (E, EB, EW, ED, A, AB, AW, AD), entrées, sorties de périphérie (PE, PA), mémentos (M, MB, MW, MD), temporisations (T) et compteurs (Z), blocs de code (OB, FB, FC, SFB, SFC), blocs de données (DB), types de données utilisateur. 	<p>Vous pouvez définir des mnémoniques locaux pour :</p> <ul style="list-style-type: none"> paramètres de bloc (paramètres d'entrée, de sortie, d'entrée/sortie), données statiques d'un bloc, données temporaires d'un bloc.
Lieu de définition	Table des mnémoniques	Table de déclaration des variables du bloc

6.1.3 Représentation des mnémoniques globaux et des mnémoniques locaux

Dans la section des instructions d'un programme, vous pouvez distinguer les mnémoniques globaux des mnémoniques locaux de la manière suivante :

- Les mnémoniques de la table des mnémoniques (globaux) sont représentés entre guillemets (" ").
- Les mnémoniques de la table de déclaration des variables du bloc (locaux) sont précédés du signe "#".

Vous n'avez pas besoin de saisir vous-même les guillemets ou le signe #. Le mnémonique sera automatiquement complété après vérification de la syntaxe lors de la saisie du programme en CONT, LOG ou LIST.

Toutefois, lorsque la confusion est possible, par exemple parce que des mnémoniques identiques ont été utilisés dans la table des mnémoniques et dans la table de déclaration des variables, vous devez identifier de manière explicite le mnémonique global que vous souhaitez utiliser. En effet, le logiciel interprète les mnémoniques non identifiés comme étant des variables locales.

De plus, l'identification des mnémoniques globaux s'avère nécessaire lorsque ceux-ci contiennent des caractères d'espacement.

Nota

La commande **Affichage > Afficher avec > Représentation symbolique** permet d'aller et venir entre l'affichage des mnémoniques globaux déclarés et celui des adresses absolues correspondantes.

6.1.4 Paramétrage du classement d'opérandes (absolu/symbolique)

La commande de menu **Outils > Paramètres** (onglet "Général") vous permet de sélectionner le classement d'opérandes "Absolu" ou "Symbolique".

Lorsque des affectations sont modifiées ultérieurement dans la table des mnémoniques, ce paramétrage permet de déterminer si l'opérande absolu ou l'opérande symbolique sont modifiés dans le programme utilisateur. Le classement d'opérandes sélectionné n'a de signification que si vous avez ouvert puis enregistré le bloc de code.

Dans le cas du paramétrage "Classement **absolu**", l'opérande absolu reste conservé dans le programme utilisateur après une modification d'affectation dans la table des mnémoniques (le mnémonique est modifié) ; dans le cas du paramétrage "Classement **symbolique**", l'opérande symbolique reste conservé (l'opérande absolu est modifié). Dans le cas d'appels de blocs tels que CALL, CC ou UC, c'est toujours le numéro de bloc absolu qui est significatif (c'est-à-dire que seul le mnémonique est toujours modifié).

Exemple :

L'exemple suivant indique l'effet du classement d'opérandes lors d'une modification de la table des mnémoniques dans le programme utilisateur.

Instruction avant modification	U "Mnémonique_A" O "Mnémonique_B"		(Mnémonique_A = E0.1) (Mnémonique_B = E0.2)
Modification d'affectation dans la table des mnémoniques	Mnémonique_A = E0.1 Mnémonique_B = E0.2	--> -->	Mnémonique_A = E0.2 Mnémonique_B = E0.1
Bloc ouvert avec "Classement absolu"	U "Mnémonique_B" O "Mnémonique_A"		(E0.1) (E0.2)
Bloc ouvert avec "Classement symbolique"	U "Mnémonique_A" O "Mnémonique_B"		(E0.2) (E0.1)

6.1.5 Table des mnémoniques pour mnémoniques globaux

Dans la table des mnémoniques, vous définissez les mnémoniques globaux.

6.1.5.1 Structure et éléments de la table des mnémoniques

Structure de la table des mnémoniques

Filtre

Entrées

Editer le filtre...

☐
Afficher les opérandes sans mnémonique

Etat	Mnémonique	Adresse	Type de d	Commentaire
	Commutateur 2	E 0.2	BOOL	Pour la connexion en série
	Commutateur 3	E 0.3	BOOL	Pour la connexion en parallèle

Etat

- ? repère un mnémonique inutilisé ; la cellule du mnémonique est à fond bleu.
- = repère un mnémonique possédant un duplicata (absolu ou symbolique) ; les cellules concernées sont à fond rouge pastel.
- X repère les entrées dont la syntaxe est erronée ; la cellule concernée est à fond rouge.

Mnémonique

Le nom du mnémonique ne doit pas dépasser 24 caractères.

Vous ne pouvez pas affecter de mnémoniques aux opérandes de blocs de données (DBD, DBW, DBB, DBX) dans la table des mnémoniques. Les noms de ces opérandes sont définis par la déclaration dans les blocs de données.

Pour les blocs d'organisation (OB) et quelques blocs fonctionnels système (SFB) et fonctions système (SFC), il existe des entrées prédéfinies que vous importez dans la table des mnémoniques de votre programme au moment où vous l'éditez.

Opérande

Il s'agit de l'adresse d'un opérande précis.

Exemple : entrée E 12.1

La syntaxe de l'opérande est vérifiée lors de la saisie. Le logiciel contrôle également si l'affectation de cette adresse au type de données spécifié est autorisée.

Type de données

Vous pouvez choisir parmi les différents types de données que STEP 7 Lite vous propose. Une fois l'opérande saisi, un type de données par défaut correct s'inscrit dans ce champ. Quand l'opérande saisi autorise plusieurs types de données, les autres types corrects sont mis à disposition dans une liste de sélection.

Commentaire

Vous pouvez affecter des commentaires à tous les mnémoniques. La combinaison de mnémoniques courts et de commentaires détaillés permet d'assurer une bonne documentation du programme ainsi qu'une programmation efficace. Un commentaire ne doit pas dépasser 80 caractères.

6.1.5.2 Opérandes et types de données autorisés dans la table des mnémoniques

La notation employée doit être la même pour toute la table des mnémoniques. Pour passer du jeu d'abréviations françaises au jeu d'abréviations internationales, ou inversement, il faut avoir recours à la commande **Outils > Paramètres...**

Abréviation anglaise	Abréviation allemande	Désignation	Type de données	Plage d'adresses
I	E	Bit d'entrée	BOOL	0.0..65535.7
IB	EB	Octet d'entrée	BYTE, CHAR	0..65535
IW	EW	Mot d'entrée	WORD, INT, S5TIME, DATE	0..65534
ID	ED	Double mot d'entrée	DWORD, DINT, REAL, TOD, TIME	0..65532
Q	A	Bit de sortie	BOOL	0.0..65535.7
QB	AB	Octet de sortie	BYTE, CHAR	0..65535
QW	AW	Mot de sortie	WORD, INT, S5TIME, DATE	0..65534
QD	AD	Double mot de sortie	DWORD, DINT, REAL, TOD, TIME	0..65532
M	M	Bit de memento	BOOL	0.0..65535.7
MB	MB	Octet de memento	BYTE, CHAR	0..65535
MW	MW	Mot de memento	WORD, INT, S5TIME, DATE	0..65534
MD	MD	Double mot de memento	DWORD, DINT, REAL, TOD, TIME	0..65532
PIB	PEB	Octet d'entrée de périphérie	BYTE, CHAR	0..65535
PID	PEB	Double mot d'entrée de périphérie	DWORD, DINT, REAL, TOD, TIME	0..65532
PIW	PEW	Mot d'entrée de périphérie	WORD, INT, S5TIME, DATE	0..65534
PQB	PAB	Octet de sortie de périphérie	BYTE, CHAR	0..65535
PQD	PAD	Double mot de sortie de périphérie	DWORD, DINT, REAL, TOD, TIME	0..65532
PQW	PAW	Mot de sortie de périphérie	WORD, INT, S5TIME, DATE	0..65534
T	T	Temporisation	TIMER	0..65535
C	Z	Compteur	COUNTER	0..65535
FB	FB	Bloc fonctionnel	FB	0..65535
OB	OB	Bloc d'organisation	OB	1..65535
DB	DB	Bloc de données	DB, FB, SFB, UDT	1..65535
FC	FC	Fonction	FC	0..65535
SFB	SFB	Bloc fonctionnel système	SFB	0..65535
SFC	SFC	Fonction système	SFC	0..65535
UDT	UDT	Type de données utilisateur	UDT	0..65535

6.1.5.3 Mnémoniques incomplets ou non univoques dans la table des mnémoniques

Mnémoniques incomplets

Il est possible aussi d'enregistrer des mnémoniques incomplets ce qui vous permet, par exemple, de ne définir dans un premier temps que leur nom et de compléter plus tard l'indication de l'adresse (opérande). Vous pouvez, en particulier, interrompre votre travail dans la table des mnémoniques à tout moment et enregistrer l'état intermédiaire de cette dernière. Pour pouvoir utiliser le mnémonique lors de la création du logiciel sans recevoir de message d'erreur, il faut toutefois que le mnémonique, l'opérande et le type de données soient indiqués.

Formation de mnémoniques non univoques

Vous pouvez avoir des mnémoniques non univoques lorsque vous ajoutez un mnémonique à la table des mnémoniques et que le nom ou l'adresse spécifiée figure déjà dans la table pour un autre mnémonique. Le nouveau et l'ancien mnémonique ne sont donc plus univoques.

C'est ce qui se produit, par exemple, quand vous copiez et insérez un mnémonique pour modifier ensuite légèrement l'entrée dans la copie.

Repérage des mnémoniques non univoques

Les mnémoniques non univoques sont repérés, dans la table, par une mise en valeur graphique (couleur, police) afin d'attirer votre attention sur la nécessité d'une correction. Vous pouvez afficher tous les mnémoniques ou, par un filtre, seulement les mnémoniques univoques ou seulement les mnémoniques non univoques.

Correction de la non-univocité

Un mnémonique non univoque le devient lorsque vous modifiez le composant - nom et (ou) opérande - qui a engendré la non-univocité. Le mnémonique ayant auparavant la même adresse reprend automatiquement son unicité.

6.1.6 Possibilités de saisie de mnémoniques globaux

Il y a plusieurs façons de saisir les mnémoniques qui seront utilisés ultérieurement lors de la programmation :

- Saisie directement dans la table des mnémoniques
Vous pouvez inscrire directement dans une table des mnémoniques et les opérandes qui leur sont affectés. C'est le procédé recommandé quand il s'agit d'en saisir plusieurs et de créer la table, car les mnémoniques déjà définis sont affichés à l'écran, ce qui vous permet de vous y retrouver.
- Saisie dans une boîte de dialogue
Vous pouvez ouvrir une boîte de dialogue dans l'éditeur de bloc et y définir un nouveau mnémonique. Ce procédé convient à la définition de mnémoniques individuels, par exemple quand vous constatez, au cours de la programmation, qu'un mnémonique manque ou doit être corrigé. Vous évitez ainsi d'afficher la table des mnémoniques.
- Importation de tables des mnémoniques depuis d'autres tableurs
Vous pouvez créer les données pour la table des mnémoniques à l'aide de votre tableur préféré et importer ensuite le fichier créé dans la table des mnémoniques.
- Saisie des mnémoniques de bloc dans une boîte de dialogue
Dans la fenêtre du projet, vous pouvez ouvrir la boîte de dialogue "Nouveau bloc" soit par la commande **Fichier > Nouveau > Bloc...**, soit par celle du menu contextuel **Nouveau > Bloc....** Vous afficherez le menu contextuel en cliquant avec le bouton droit de la souris sur l'objet sélectionné. Vous pouvez alors définir le mnémonique de bloc dans cette boîte de dialogue.
- Saisie des mnémoniques de bloc dans l'éditeur de bloc
Vous pouvez choisir la vue "Propriétés" dans l'éditeur de bloc et y éditer ou modifier le nom symbolique du bloc. La modification que vous apportez au champ "Mnémonique" ou "Commentaire de mnémonique" est aussitôt enregistrée. Même si vous quittez la vue des propriétés sans enregistrer, les modifications apportées à ces deux champs seront acceptées et reportées à toutes les vues.
- Saisie des mnémoniques de bloc dans la fenêtre du projet
Dans la fenêtre du projet, vous pouvez modifier le nom d'un bloc soit par double-clic sur le bloc, soit par la commande **Renommer** du menu contextuel. Vous afficherez le menu contextuel en cliquant avec le bouton droit de la souris sur l'objet sélectionné.

6.1.6.1 Remarques générales sur la saisie de mnémoniques

Pour inscrire de nouveaux mnémoniques dans la table des mnémoniques, vous vous positionnez dans la première ligne vide de la table et en complétez les champs. Vous pouvez insérer de nouvelles lignes avant ou après la ligne en cours au moyen des commandes **Insertion > Ligne > Avant sélection/Après sélection**. Quand la ligne précédant la position du pointeur contient déjà un opérande, l'insertion d'un nouveau mnémonique vous est facilitée par des valeurs par défaut s'inscrivant automatiquement dans les colonnes "Opérande" et "Type de données" : un opérande dérivé de celui de la ligne précédente et le type de données par défaut.

Les commandes du menu "Edition" permettent de copier, puis de modifier des entrées existantes. Les mnémoniques qui ne sont pas encore entièrement définis peuvent également être enregistrés.

En saisissant les mnémoniques dans la table, vous devez tenir compte des particularités suivantes :

Colonne	Nota
Mnémonique	Ce nom doit être univoque dans l'ensemble de la table des mnémoniques. Quand vous quittez ce champ, un repère est placé devant un mnémonique non univoque. Un mnémonique ne doit pas dépasser 24 caractères. Les guillemets " ne sont pas autorisés.
Opérande	Quand vous quittez ce champ, le programme vérifie si l'opérande indiqué est autorisé.
Type de données	Une fois l'opérande saisi, une valeur par défaut correcte s'inscrit dans ce champ. Quand l'opérande saisi autorise plusieurs types de données, les autres types corrects sont mis à disposition dans une liste de sélection.
Commentaire	Ce champ vous permet de saisir des remarques (80 caractères au maximum) décrivant la fonction du mnémonique. La saisie d'un commentaire est facultative.

6.1.6.2 Saisie de mnémoniques globaux individuels dans des boîtes de dialogue

La procédure suivante vous montre comment modifier ou créer, lors de la programmation de blocs, des mnémoniques via des boîtes de dialogue sans devoir afficher la table des mnémoniques.

Cette méthode est utile lorsque vous ne désirez éditer qu'un seul mnémonique. Nous vous conseillons, pour la modification de plusieurs mnémoniques, d'ouvrir la table des mnémoniques et d'y travailler directement.

Activer l'affichage des mnémoniques dans le bloc

Lorsqu'un bloc est ouvert, vous pouvez activer l'affichage des mnémoniques dans l'éditeur de bloc avec la commande **Affichage > Afficher avec > Représentation symbolique**. Une coche devant le nom de la commande indique que la représentation symbolique est active.

Définir des mnémoniques lors de la saisie du programme

1. Sélectionnez, dans la section des instructions de votre programme, l'adresse absolue à laquelle vous voulez affecter un mnémonique.
2. Choisissez soit la commande **Edition > Mnémoniques...**, soit celle du menu contextuel **Editer les mnémoniques....** Vous afficherez le menu contextuel en cliquant sur l'objet sélectionné avec le bouton droit de la souris.
3. Complétez la boîte de dialogue affichée en y inscrivant en particulier un mnémonique, puis fermez-la.

Le mnémonique défini s'inscrit dans la table des mnémoniques.

Nota

Les indications créant des mnémoniques non univoques sont repérées par un signe d'égalité = dans la boîte de dialogue, quand vous avez quitté le champ de saisie du mnémonique, ainsi que dans la colonne d'état de la table des mnémoniques. Les lignes correspondantes apparaissent sur fond rouge pastel.

Tant que vous n'avez pas enregistré l'instruction dont vous avez sélectionné l'opérande pour appeler la boîte de dialogue, le mnémonique est repéré comme mnémonique non utilisé par un "?" et un fond bleu, dans la boîte de dialogue "Editer les mnémoniques" et dans la table de mnémoniques elle-même.

Un opérande à mnémonique non univoque s'affiche sous forme absolue dans CONT, LOG et LIST. Vous ne pourrez pas appeler la boîte de dialogue "Editer les mnémoniques" pour cet opérande.

6.1.6.3 Saisie de plusieurs mnémoniques globaux dans la table des mnémoniques

Ouverture d'une table des mnémoniques

Il y a plusieurs façons d'ouvrir la table des mnémoniques :

- Faites un double clic sur la table des mnémoniques dans la fenêtre du projet.
- Sélectionnez la table des mnémoniques dans la fenêtre du projet, appelez le menu contextuel avec le bouton droit de la souris et choisissez-y la commande **Ouvrir**.

La table des mnémoniques du programme en cours s'affiche dans sa propre fenêtre. Vous pouvez alors créer ou modifier des mnémoniques. La table est vide lorsque vous l'ouvrez pour la première fois.

Saisie de mnémoniques

Pour entrer de nouveaux mnémoniques dans la table, vous vous positionnez dans la première ligne vide de la table et en complétez les champs. Vous pouvez insérer de nouvelles lignes vides avant ou après la ligne en cours au moyen de la commande **Insertion > Ligne avant la sélection/Ligne après la sélection**. Les commandes du menu **Edition** permettent de copier, puis de modifier des entrées existantes. Ensuite, vous enregistrez la table des mnémoniques et la fermez. Vous pouvez aussi enregistrer des mnémoniques qui ne sont pas encore entièrement définis.

Tri des mnémoniques

Il est possible de trier les enregistrements logiques de la table des mnémoniques dans l'ordre alphabétique des mnémoniques, des opérandes, des types de données ou des commentaires.

Cliquez sur le titre de la colonne concernée pour démarrer le tri. Une petite flèche bleue verticale s'affiche à droite du titre pour indiquer le sens de tri.

La table est alors triée suivant les enregistrements de cette colonne. Cliquez à nouveau sur la flèche pour inverser le sens de tri.

Filtrage des mnémoniques

Les filtres vous permettent de choisir des sous-ensembles parmi tous les enregistrements de la table.

Un clic sur le bouton "Editer le filtre..." ouvre la boîte de dialogue "Editer le filtre".

Vous y définissez les critères auxquels les enregistrements logiques devront satisfaire pour être affichés. Il peut s'agir de :

- mnémoniques, opérandes, types de données, commentaires ;
- mnémoniques avec l'état "valide", "invalide (non univoque, incomplet)".

Les différents critères sont combinés par ET. Les enregistrements logiques affichés commencent par les chaînes de caractères indiquées.

Pour en savoir plus sur les possibilités offertes par la boîte de dialogue "Editer le filtre", consultez la rubrique Filtrage de la table des mnémoniques.

6.1.6.4 Exportation et importation de tables de mnémoniques

Vous pouvez exporter dans un fichier de texte la table des mnémoniques affichée, pour la traiter avec un éditeur de texte de votre choix, par exemple.

Vous pouvez importer, dans votre table des mnémoniques, des tables créées avec une autre application et poursuivre leur traitement dans la table des mnémoniques. Cette fonction vous servira, par exemple, à adopter dans la table des mnémoniques après leur conversion des listes d'assignation créées sous STEP 5/ST.

Vous disposez du format de fichier *.SDF.

Vous pouvez exporter la table des mnémoniques entière, un sous-ensemble de cette table défini par filtre ou des lignes sélectionnées dans la représentation de la table.

6.1.7 Edition de la table des mnémoniques

6.1.7.1 Ouverture d'une table des mnémoniques

La "table des mnémoniques" est automatiquement créée sous un nouveau projet. Pour pouvoir remplacer des données globales par des mnémoniques dans un bloc, vous devez les affecter dans la table des mnémoniques.

Ouvrez la "table des mnémoniques" en cliquant deux fois dessus dans la fenêtre du projet ou affichez un menu contextuel en cliquant avec le bouton droit de la souris. La commande **Ouvrir** permet alors d'ouvrir la table.

6.1.7.2 Définition de mnémoniques individuels

1. Dans l'éditeur de bloc, activez la représentation symbolique à l'aide de la commande **Affichage> Afficher avec > Représentation symbolique**. Une coche devant le nom de la commande indique que la représentation symbolique est active.
2. Cliquez dans le réseau sur l'opérande pour lequel vous désirez définir un mnémonique.
3. Choisissez soit la commande **Edition > Mnémoniques...**, soit celle du menu contextuel **Editer les mnémoniques....** Vous afficherez le menu contextuel en cliquant sur l'objet sélectionné avec le bouton droit de la souris.
4. Inscrivez, dans la boîte de dialogue qui apparaît alors, le mnémonique, le type de données de l'opérande et le commentaire éventuel.
Le mnémonique doit être univoque dans l'ensemble de la table et comporter au maximum 24 caractères. Les guillemets ne sont pas autorisés.
5. Confirmez par "OK". Le mnémonique défini est inscrit dans la table des mnémoniques et remplace l'opérande sélectionné dans la section des instructions.

6.1.7.3 Insertion de lignes de mnémonique

Pour insérer une ligne de mnémonique avant la position du pointeur, choisissez la commande **Insertion > Ligne > Avant la sélection/Après la sélection**.

Il y a plusieurs façons d'insérer une ou plusieurs lignes de mnémonique provenant du presse-papiers :

- avec le bouton gauche de la souris, cliquez sur la ligne (et non pas sur l'en-tête de ligne) à partir de laquelle vous souhaitez insérer la ou les lignes de mnémoniques,
- choisissez la commande **Edition > Coller**,
- cliquez sur l'icône correspondante de la barre d'outils ou
- appuyez sur les touches CTRL + V.

6.1.7.4 Suppression de lignes de mnémonique

Pour supprimer les lignes de mnémonique sélectionnées et en conserver une copie dans le presse-papiers, vous avez les possibilités suivantes :

- choisissez la commande **Edition > Couper**,
- cliquez sur l'icône correspondante de la barre d'outils ou
- appuyez sur les touches CTRL + X.

Pour effacer les lignes de mnémonique sélectionnées sans en conserver de copie, vous avez les possibilités suivantes :

- choisissez la commande **Edition > Effacer** ou
- appuyez sur la touche SUPPR.

Sachez que lorsque vous coupez et supprimez les propriétés spécifiques de l'objet, il n'est plus possible d'annuler cette commande.

6.1.7.5 Filtrage de la table des mnémoniques

Pour définir un filtre pour l'affichage dans la fenêtre active, procédez de la manière suivante :

1. Cliquez sur le bouton "Editer le filtre".
2. Dans la boîte de dialogue "Editer le filtre", sélectionnez un filtre existant par son numéro ou définissez-en un nouveau.
3. Cliquez à cet effet sur le bouton "Nouveau filtre".
4. Donnez un nom univoque au nouveau filtre.
5. Cochez les options souhaitées.
6. Cliquez sur le bouton "Appliquer". Le filtre créé peut à présent être sélectionné dans la liste déroulante.

Seuls s'afficheront les mnémoniques qui répondent aux critères de filtre en vigueur. Vous pouvez utiliser plusieurs critères à la fois. Les critères que vous indiquez sont combinés entre eux.

Vous pouvez aussi sélectionner un filtre existant dans la liste de sélection déroulante.

Vous disposez des filtres prédéfinis suivants :


- Afficher tout : affiche tous les mnémoniques (valeur par défaut)
- Erronés : affiche tous les mnémoniques en double ou à syntaxe erronée
- Inutilisés : affiche tous les mnémoniques inutilisés
- Entrées : affiche tous les mnémoniques représentant des entrées
- Sorties : affiche tous les mnémoniques représentant des sorties
- Mémentos : affiche tous les mnémoniques représentant des mémentos
- Blocs : affiche tous les mnémoniques représentant des blocs
- Tempos et compteurs : affiche tous les mnémos représentant temporisations et compteurs

Ces filtres prédéfinis ne peuvent être ni modifiés ni supprimés. Mais vous pouvez les dupliquer ou les enregistrer sous un nouveau nom.

6.1.7.6 Opérandes libres

En sélectionnant le filtre prédéfini "Inutilisés" dans la table des mnémoniques, vous obtenez la liste de tous ceux présentant la caractéristique suivante :

- ils sont définis dans la table,
- mais ils ne sont pas utilisés dans les différentes parties du programme utilisateur.

Un opérande libre (mnémonique inutilisé) est repéré par un point d'interrogation . En plus, la ligne du mnémonique apparaît sur fond bleu.

6.1.7.7 Mnémoniques manquants

Activez la case d'option "Afficher les mnémoniques manquants" dans la table des mnémoniques pour obtenir la liste de tous les opérandes utilisés dans le programme et correspondant au filtre en vigueur, et pas seulement ceux qui sont munis d'un mnémonique.

6.1.7.8 Tri de la table des mnémoniques

Pour choisir un critère de tri pour l'affichage de la table en cours, procédez de la manière suivante :

1. Cliquez sur le titre de la colonne pour lancer le tri. Une flèche bleue verticale s'affiche du côté droit.
2. Sélectionnez le sens de tri qui vous convient (il est indiqué par le sens de la flèche).

La table des mnémoniques sera triée d'après les entrées dans cette colonne. En répétant cette action, vous pouvez inverser l'ordre de tri.

6.1.7.9 Sélection de lignes de mnémonique

Pour sélectionner la ligne du mnémonique dans laquelle se trouve le pointeur, vous avez les possibilités suivantes :

- cliquez sur l'en-tête de ligne, à gauche de la ligne de mnémonique souhaitée ou
- appuyez sur les touches MAJ + ESPACE.

Pour sélectionner toutes les lignes de la table des mnémoniques en cours, vous avez les possibilités suivantes :

- choisissez la commande **Edition > Sélectionner > Tout** ou
- appuyez sur les touches CTRL + A.

Pour annuler une sélection, choisissez la commande **Edition > Annuler la sélection**.

6.1.7.10 Copie de lignes de mnémonique dans le presse-papiers

Pour copier une ou plusieurs lignes de mnémonique sélectionnées, vous avez les possibilités suivantes :

- choisissez la commande **Edition > Copier**,
- cliquez sur l'icône correspondante de la barre d'outils ou
- appuyez sur les touches CTRL + C.

L'ancien contenu du presse-papiers sera écrasé.

6.1.7.11 Enregistrement d'une table des mnémoniques

Il n'est pas nécessaire d'enregistrer explicitement la table des mnémoniques. Quand vous quittez l'un de ses champs, le contenu en est automatiquement enregistré et se trouve aussitôt disponible, par exemple lors de la programmation. La table des mnémoniques est enregistrée en même temps que le projet (Commande de menu **Fichier > Enregistrer** ou **Fichier > Enregistrer sous**).

6.1.8 Modifier les paramètres de fenêtre

6.1.8.1 Afficher ou masquer la barre d'outils

Pour afficher ou masquer la barre d'outils, choisissez la commande **Affichage > Barre d'outils**.

Quand la barre d'outils est affichée, la commande est cochée.

6.1.8.2 Afficher ou masquer la barre d'état

Pour afficher ou masquer la barre d'état, choisissez la commande **Affichage > Barre d'état**.

Quand la barre d'état est affichée, la commande est cochée.

6.1.8.3 Positionner la barre d'outils

Pour changer la position de la barre d'outils affichée, procédez de la manière suivante :

1. Amenez le pointeur sur une zone vide de la barre en question.
2. Tout en maintenant le bouton gauche de la souris enfoncé, faites glisser la barre à la position souhaitée.
3. Relâchez le bouton gauche de la souris.

6.1.8.4 Taille de l'affichage dans une fenêtre de travail

Pour agrandir **progressivement** la taille de l'affichage dans les vues Editeur de bloc, Détails du projet et Détails du programme, vous avez les possibilités suivantes :

- choisissez la commande **Affichage > Agrandir** ou
- appuyez sur les touches CTRL + NUM+.

Pour réduire progressivement la taille de l'affichage dans la fenêtre active, vous avez les possibilités suivantes :

- choisissez la commande **Affichage > Réduire** ou
- appuyez sur les touches CTRL + NUM-.

Pour modifier la taille de l'affichage selon le **facteur choisi**, vous avez les possibilités suivantes :

1. choisissez la commande Affichage > Facteur d'agrandissement ou
2. dans la boîte de dialogue "Facteur d'agrandissement", sélectionnez le facteur d'agrandissement souhaité.
3. Confirmez par "OK".

6.2 Utilisation de blocs

6.2.1 Editeur de bloc

L'éditeur de bloc sert à créer et à tester des blocs pour les CPU de SIMATIC S7-300 dans les langages de programmation CONT, LOG et LIST.

Par saisie incrémentale, ces langages permettent de créer les blocs un par un et de les éditer pareillement.

L'éditeur ne sert pas seulement à élaborer le programme, c'est-à-dire à créer, puis à éditer les blocs de code, blocs de données et types de données utilisateur. Il vous offre aussi des fonctions supplémentaires pour la programmation, le test et la mise en service du programme :

- programmation avec des mnémoniques ;
- lecture des informations d'état et des données de fonctionnement de la CPU au moyen de la commande Etat du module (menu Outils) ;
- affichage et changement du mode de fonctionnement de la CPU (menu Outils) ;
- effacement général de la CPU ;
- affichage et réglage de la date et de l'heure de la CPU au moyen de la commande Paramètres de la CPU (menu Outils) ;
- visualisation d'un bloc de code déterminé (test de programmes CONT/LOG/LIST dans la visualisation d'état du programme) ;
- possibilité de multi-instances, c'est-à-dire qu'un même DB d'instance peut contenir les données de plusieurs FB :
 - informations sur les blocs de données d'instance,
 - déclaration de multi-instances ;
- table de déclaration des variables :
 - elle permet d'éditer plusieurs déclarations à la fois (copier, couper, coller),
 - pour les blocs de code, la table de déclaration des variables et la section des instructions sont affichées dans la même fenêtre de travail.

Il faut d'abord qu'un projet existe pour pouvoir générer des blocs et les éditer.

6.2.2 Choix du langage de programmation

6.2.2.1 Langages de programmation de l'éditeur de bloc

Lors de la création d'un bloc ou d'une source, vous déterminez dans les propriétés de l'objet avec quel langage de programmation et quel éditeur vous voulez écrire ce bloc ou cette source. Vous avez également la possibilité de changer de langage de programmation dans l'onglet "Propriétés" lorsqu'un bloc est ouvert.

Démarrer l'éditeur de bloc

Vous démarrez l'éditeur de bloc par un double clic sur le bloc concerné ou par la commande **Ouvrir** du menu contextuel qui s'affiche lorsque vous cliquez avec le bouton droit de la souris sur l'objet sélectionné.

Vous disposez des langages de programmation indiqués dans le tableau pour créer le programme.

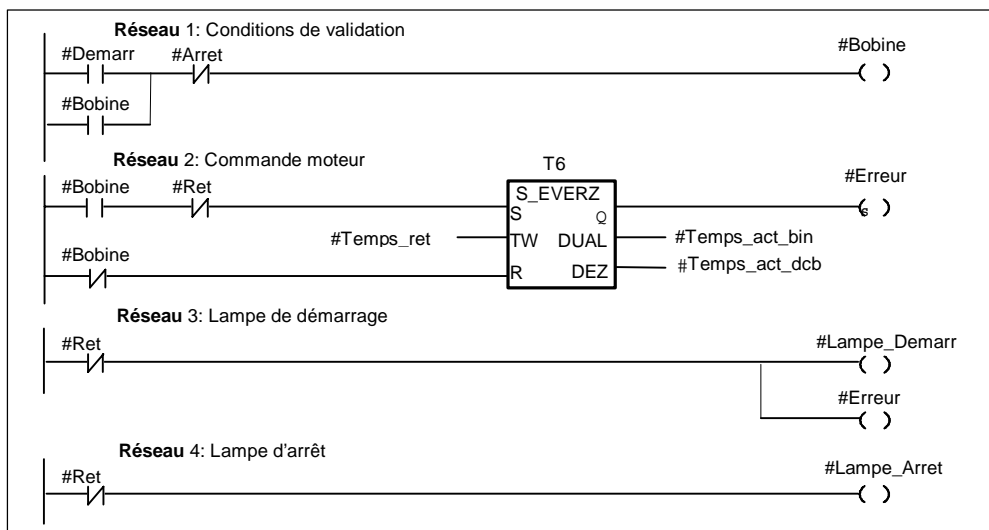
Langage de programmation	Groupe d'utilisateurs	Application	Saisie incrémentale	Possibilité de redocumenter le bloc de la CPU
Liste d'instructions LIST	Utilisateurs souhaitant une programmation proche de la machine	Programmes optimisés en temps d'exécution et en espace mémoire	•	•
Schéma à contacts CONT	Utilisateurs habitués aux schémas de circuits	Programmation de commandes combinatoires	•	•
Logigramme LOG	Utilisateurs habitués aux boîtes logiques de l'algèbre booléenne	Programmation de commandes combinatoires	•	•

Pour un bloc exempt d'erreur, vous pouvez passer d'une représentation à l'autre dans les langages CONT, LOG et LIST. Les parties de programme ne pouvant pas être représentées dans le langage cible sont représentées en LIST.

6.2.2.2 Langage de programmation CONT (schéma à contacts)

La représentation en langage de programmation CONT (schéma à contacts) s'inspire des schémas de circuits. Les éléments d'un schéma de circuit, tels que contacts à fermeture et contacts à ouverture, sont rassemblés dans des réseaux. Un ou plusieurs réseaux forment la section des instructions complète d'un bloc de code.

Exemple de réseaux en CONT

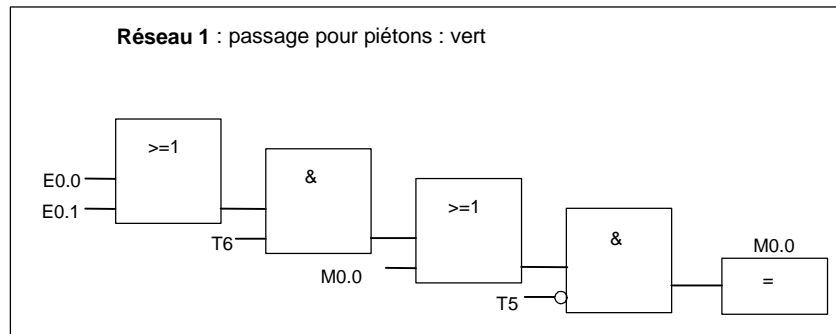


Le langage de programmation CONT fait partie de STEP 7 Lite.

6.2.2.3 Langage de programmation LOG (logigramme)

Le langage de programmation LOG (logigramme) utilise les boîtes fonctionnelles graphiques de l'algèbre booléenne pour représenter des éléments logiques. Il permet en outre de représenter des fonctions complexes, telles que les fonctions mathématiques en les mettant directement en liaison avec ces boîtes logiques.

Exemple de réseau en LOG



Le langage de programmation LOG fait partie de STEP 7 Lite.

6.2.2.4 Langage de programmation LIST (liste d'instructions)

Le langage de programmation LIST (liste d'instructions) est un langage textuel proche du langage machine. Chaque instruction correspond à une étape de l'exécution du programme par la CPU. Vous pouvez regrouper plusieurs instructions en réseaux.

Exemple de réseaux en LIST

```

Réseau 1 : Commande soupape de vidange
U(
O
O #Bobine
)
UN #Fermer
= #Bobine

Réseau 2 : Indication "Soupape ouverte"
U #Bobine
= #Indic_Ouverte

Réseau 3 : Indication "Soupape fermée"
UN #Bobine
= #Indic_Fermee
    
```

Le langage de programmation LIST fait partie de STEP 7 Lite.

6.2.3 Création de blocs

6.2.3.1 Types de données utilisateur (UDT)

Les types de données utilisateur (user data type, UDT) sont des structures de données particulières, créées par vous. Vous pouvez les utiliser, une fois définis, dans l'ensemble du programme utilisateur.

- Vous pouvez utiliser les UDT comme types de données simples ou comme types de données complexes dans la déclaration des variables de blocs de code (FC, FB, OB) ou encore comme types de données pour des variables dans un bloc de données (DB). L'avantage réside dans le fait que vous ne définissez qu'une seule fois une structure de données spéciale, que vous allez utiliser plusieurs fois en l'affectant à un nombre illimité de variables.
- Les UDT peuvent servir de modèle afin de créer des blocs de données de même structure. Cela signifie que vous ne définissez qu'une seule fois la structure et créez ensuite les blocs de données requis par simple affectation du type de données utilisateur (exemple d'une recette : la structure du DB est toujours la même, seules les quantités varient).

Structure d'un UDT

Après l'ouverture de l'UDT, une nouvelle fenêtre de travail s'ouvre montrant la table - et, plus précisément, la vue des déclarations - pour ce type de données utilisateur.

- La première et la dernière ligne contiennent déjà les déclarations `STRUCT` et `END_STRUCT` pour début et fin de type de données utilisateur ; vous ne pouvez pas modifier ces lignes.
- Pour éditer un type de données utilisateur, vous saisissez vos données dans les colonnes correspondantes à partir de la deuxième ligne de la table de déclaration. Vous pouvez modifier ou écraser la variable "udt_bool" écrite par défaut par le programme.
- Vous pouvez structurer les types de données utilisateur à partir de :
 - types de données simples,
 - types de données complexes,
 - types de données utilisateur existant.

Les types de données utilisateur du programme utilisateur ne sont pas chargés dans la CPU S7. Ils sont créés et édités directement dans l'éditeur de bloc.

6.2.3.2 Propriétés de bloc

Les propriétés d'un bloc vous permettent de l'identifier avec certitude (par ex. pour mettre à jour la version).

Lorsque vous avez sélectionné un bloc, vous pouvez en éditer les propriétés dans la vue "Propriétés". Cette vue affiche aussi, à titre informatif, des données que vous ne pouvez pas éditer.

Avertissement

La modification que vous apportez au champ "Mnémonique" ou "Commentaire" est aussitôt enregistrée. Même si vous quittez la vue des propriétés sans enregistrer, les modifications apportées à ces deux champs seront acceptées et reportées à toutes les vues.

Nota

C'est avec la commande **Outils > Paramètres...** que vous choisissez le jeu d'abréviations à utiliser pour programmer vos blocs.

Tableau des propriétés de bloc

Propriété	Signification	Editable ou sélectionnable	Exemple
Propriétés spécifiques au projet			
Nom	Nom du bloc (type et numéro)	Non	FB10
Langage de création	Langage de création ou d'édition en cours	Oui	LIST
Mnémonique	Nom symbolique du bloc	Oui	Regulateur_PID
Commentaire	Commentaire du nom symbolique attribué au bloc	Oui	Commande de régulateur
Horodatage			
Code créé le	Date et heure de création du bloc	Non	24.08.2000 09:13:16
Code modifié le	Date et heure du dernier enregistrement du bloc avec code modifié	Non	24.08.2000 09:13:16
Interface modifiée le	Date et heure du dernier enregistrement du bloc avec interface modifiée	Non	24.08.2000 09:13:16
Commentaire	Commentaire du bloc : 1er champ : titre 2ème champ : commentaire	Oui	Bloc fonctionnel pour commande de régulateur
Propriétés dans l'en-tête du bloc			
Nom (en-tête)	Nom du bloc (8 caractères au plus, attribué par l'auteur)	Oui	PID
Version	Numéro de version du bloc (les 2 nombres étant compris entre 0 et 15, c.-à-d. 0.0 à 15.15)	Oui	3.10
Famille	Nom de la famille du bloc (8 caractères au plus sans espace)	Oui	Régulateur

Propriété	Signification	Editable ou sélectionnable	Exemple
Auteur	Nom de l'auteur, de la société, du service (8 caractères au plus sans espace)	Oui	Siemens
Longueurs			
Données locales	Taille des données locales en octets	Non	10
Mémoire de chargement requise	Taille en octets de la mémoire de chargement requise	Non	142
MC7	Taille du code MC7 en octets	Non	38
Mémoire de travail requise	Taille en octets de la mémoire de travail requise	Non	74
Attributs			
Le DB de l'AP est protégé en écriture	Protection en écriture pour blocs de données : les données peuvent uniquement être lues par l'AP mais pas modifiées par le programme utilisateur.	Oui	
Protection KNOW HOW	Un bloc compilé avec cette option ne plus être affiché ou modifié par aucun outil de développement (STEP 7 Lite, STEP 7).	Non	
Bloc standard	C'est un bloc standard SIEMENS avec protection KNOW HOW. Les champs de saisie du nom, de la famille, de la version et de l'auteur sont estompés et il n'est pas possible de les éditer.	Non	
UNLINKED	Quand un bloc de données possède la propriété UNLINKED, il n'est pas intégré au programme.	Oui	
Multiinstance	Seuls les blocs fonctionnels et les blocs fonctionnels système peuvent être générés avec une propriété de multiinstance. Ils peuvent alors utiliser les DB d'instance d'autres FB/SFB.	Non	
Non-Retain	Les blocs de données ayant cet attribut retrouvent leurs valeurs initiales après chaque arrêt et redémarrage et après chaque passage de la CPU de STOP en RUN.	Oui	

La protection KNOW HOW a les conséquences suivantes :

- Lorsque vous voudrez plus tard examiner un bloc compilé dans l'éditeur de bloc, vous ne pourrez pas consulter sa section des instructions.
- La table de déclaration des variables du bloc n'affiche que celles dont le type de déclaration est var_in, var_out ou var_in_out. Celles dont le type de déclaration est var_stat ou var_temp restent cachées.

Propriétés suivant les types de bloc

Le tableau ci-dessous montre quelles propriétés peuvent être déclarées pour quels types de bloc.

Propriété	OB	FB	FC	DB	UDT
Protection KNOW HOW	•	•	•	•	–
Auteur	•	•	•	•	–
Famille	•	•	•	•	–
Nom	•	•	•	•	–
Version	•	•	•	•	–
UNLINKED	–	–	–	•	–
Le DB de l'AP est protégé en écriture	–	–	–	•	–
Non-Retain	–	–	–	•	–

6.2.3.3 Protection du bloc

Lorsque vous souhaitez protéger vos connaissances et expériences ou encore empêcher des manipulations malencontreuses, vous pouvez paramétrer une protection du bloc.

Lorsque la protection du bloc est activée :

- Le bloc protégé ne peut être modifié ni dans le projet, ni dans la CPU.
- Lorsqu'un bloc est protégé, seuls sa section de déclaration et son nom sont visibles ; la section d'instructions ou la section de données sont cachées. Dans la table de déclaration des variables du bloc, seules les variables du type var_in, var_out et var_in_out sont affichées. Les variables du type var_stat et var_temp sont cachées.

Lorsque vous activez la "Protection du bloc", une copie non protégée du bloc est enregistrée dans un fichier d'exportation. Si nécessaire, vous pouvez à nouveau importer la copie non protégée du bloc.

Pour activer la protection du bloc, procédez de la manière suivante :

1. Dans la fenêtre du projet, sélectionnez les blocs à protéger (assurez-vous que les blocs à protéger ne soient pas ouverts).
2. Choisissez la commande de menu **Outils > Protection du bloc**.
3. Dans la boîte de dialogue qui s'ouvre, entrez le nom et le chemin du fichier d'exportation.
4. Démarrez la procédure de "Protection du bloc" en cliquant sur le bouton "Exporter". Lorsque la procédure est terminée, les blocs sont caractérisés par un petit cadenas dans la fenêtre du projet.

Nota

Si vous souhaitez protéger d'autres blocs ultérieurement, choisissez respectivement un nouveau fichier d'exportation.

6.2.3.4 Propriétés de bloc autorisées pour chaque type de bloc

Le tableau suivant présente les propriétés que vous pouvez déclarer pour les différents types de blocs.

Propriété	OB	FB	FC	DB	UDT
Protection KNOW HOW	•	•	•	•	–
Auteur	•	•	•	•	–
Famille	•	•	•	•	–
Nom	•	•	•	•	–
Version	•	•	•	•	–
Unlinked	–	–	–	•	–
Le DB de l'AP est protégé en écriture	–	–	–	•	–

Instaurer une protection en écriture des blocs de données

Vous pouvez protéger les blocs de données en écriture en activant la case d'option "Le DB de l'AP est protégé en écriture". Dans ce cas, le contenu des blocs ne pourra pas être écrasé au cours de l'exécution du programme.

6.2.3.5 Affichage des longueurs de bloc

Les longueurs de bloc sont indiquées en octets.

Affichage dans la structure du programme

Cette vue indique les longueurs suivantes :

- total de toutes les longueurs de bloc sans données système dans la mémoire de chargement de la CPU,
- total de toutes les longueurs de bloc sans données système dans la mémoire de travail de la CPU.

Affichage dans les propriétés d'un bloc

Cette vue indique :

- le nombre de données locales requises : taille des données locales en octets ;
- MC7 : taille du code MC7 en octets ou taille des données utiles du DB ;
- taille dans la mémoire de chargement de la CPU (mémoire de chargement requise) ;
- taille dans la mémoire de travail de la CPU (mémoire de travail requise).

Les indications sont les mêmes, que le bloc soit ouvert dans la vue en ligne ou dans une vue hors ligne.

6.2.3.6 Comparaison de blocs

Procédez de la manière suivante :

1. Sélectionnez le ou les blocs que vous souhaitez comparer.
2. Choisissez la commande **Outils > Comparer > Bloc**.
3. Les résultats de la comparaison (ONLINE/offline) s'affichent dans une boîte de dialogue "Comparaison de blocs - Résultats".
4. Sélectionnez un bloc dans la liste de comparaison.
5. Actionnez le bouton "Détails" pour obtenir les informations sur le bloc.

Création de blocs

Procédez de la manière suivante :

1. Choisissez la commande **Fichier > Nouveau > Bloc**.
2. Dans la boîte de dialogue "Nouveau bloc", indiquez les paramètres souhaités pour le bloc à créer.
3. Confirmez par "OK".

Le bloc est alors créé et ouvert dans l'éditeur de bloc. La partie supérieure de la fenêtre sert à éditer la table de déclaration des variables, la partie inférieure à programmer la section des instructions.

Nota

Lorsque vous créez un bloc fonctionnel (FB), vous déterminez dès le départ si vous pourrez déclarer des multi-instances dans ce FB.

Création de blocs de données (DB)

Vous créez les blocs de données comme tous les autres blocs.

1. Choisissez la commande **Fichier > Nouveau > Bloc** ou cliquez sur l'icône correspondante de la barre d'outils.
2. Indiquez le bloc de données à créer dans la boîte de dialogue. Il n'est pas permis d'attribuer DB0 comme numéro de DB, car il est réservé pour le système.
3. Dans la boîte de dialogue "Nouveau bloc", sélectionnez le type de bloc de données que vous souhaitez générer :
 - bloc de données global
 - bloc de données avec UDT associé (bloc de données global)
 - bloc de données avec FB associé (bloc de données d'instance)

Dans le troisième cas, il faut sélectionner en outre le FB auquel le bloc de données doit appartenir.

Nota

STEP 7 Lite offre la possibilité, dans certaines conditions, de stocker les données de différents FB dans un seul bloc de données (bloc de données multiinstance, voir Saisie de la multi-instance dans la table de déclaration des variables).

Paramétrage des propriétés de bloc

1. Ouvrez le bloc par un double-clic ou affichez le menu contextuel en cliquant avec le bouton droit de la souris. La commande **Ouvrir** permet alors d'ouvrir le bloc sélectionné.
2. Une fois le bloc ouvert, sélectionnez l'onglet "Propriétés".
3. Tapez le nom, la famille, le mnémonique, la version et l'auteur du bloc. L'indication du nom et de la famille vous aidera à programmer des appels de bloc dans CONT. Les modifications que vous apportez au champ "Mnémonique" sont immédiatement reportées à toutes les vues. L'onglet contient en outre les indications suivantes :
 - Longueur du bloc, code MC7 et données locales.
 - Le DB de l'AP est protégé en écriture.
 - Protection KNOW HOW : un bloc présentant cette propriété ne peut être édité.
 - Bloc standard
 - Unlinked

6.2.4 Utilisation de bibliothèques

6.2.4.1 Présentation des bibliothèques de blocs

Les bibliothèques servent à stocker des éléments de programme réutilisables pour SIMATIC S7. STEP 7 Lite comprend les bibliothèques standard qui contiennent, par exemple, les fonctions standard et les fonctions système de la famille S7-300.

Les bibliothèques standard s'affichent automatiquement du côté droit de la fenêtre avec leur onglet "Blocs" si vous les appelez au moyen de la commande **Affichage > Bibliothèques**. Vous pouvez passer de l'onglet "Blocs" à l'onglet "Opérations" et vice-versa.

STEP 7 Lite contient les bibliothèques de blocs suivantes :

- **IEC Function Blocks** : blocs pour fonctions CEI, par exemple pour traiter les indications de date et d'heure, pour les opérations de comparaison, le traitement de chaînes et le choix de maxima et minima ;
- **Organization Blocks** : blocs d'organisation (OB) standard ;
- **PID Control Blocks** : blocs fonctionnels (FB) pour la régulation PID ;
- **S5-S7 Converting Blocks** : blocs servant à convertir d'anciens programmes de STEP 5 ;
- **Sytem Function Blocks** : fonctions système (SFC) et blocs fonctionnels système (SFB) ;
- **TI-S7 Converting Blocks** : fonctions standard à usage général.

6.3 Création des blocs de code

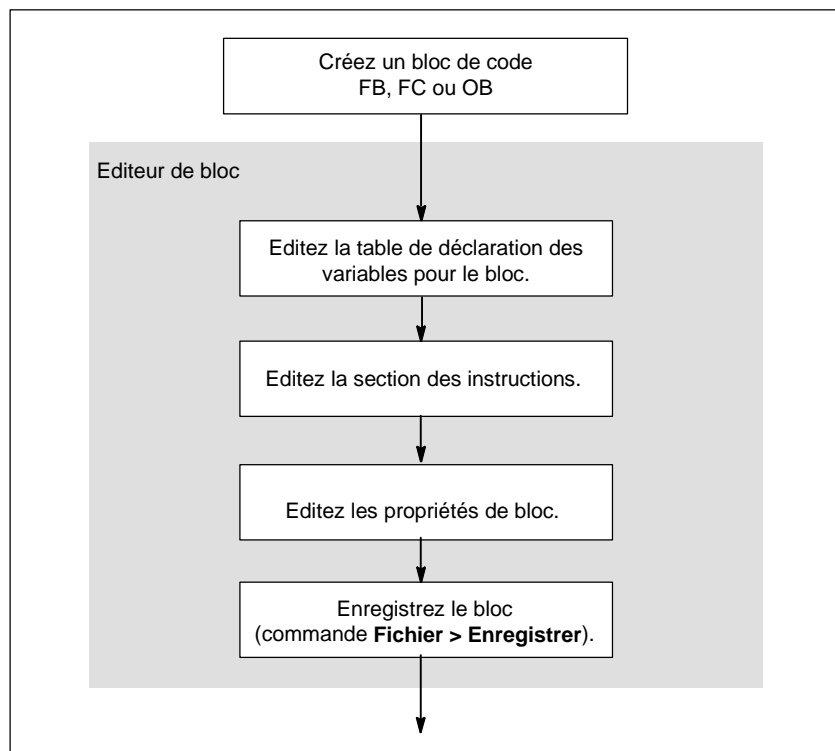
6.3.1 Principes de la création de blocs de code

6.3.1.1 Marche à suivre pour créer des blocs de code

Les blocs de code (OB, FB, FC) se composent d'une section de déclaration de variables et d'une section d'instructions et possèdent en plus des propriétés. Pour les programmer, vous devez donc éditer les trois parties suivantes :

- **table de déclaration des variables** : vous y définissez les paramètres ainsi que les variables locales du bloc ;
- **section des instructions** : vous y programmez le code du bloc que l'automate programmable devra exécuter ; il se compose d'un ou de plusieurs réseaux ; pour créer ces réseaux, vous disposez par exemple des langages de programmation LIST (liste d'instructions), CONT (schémas à contacts) et LOG (logigramme) ;
- **propriétés de bloc** : elles contiennent des informations supplémentaires qui sont inscrites par le système, comme l'horodatage ou l'indication du chemin ; vous pouvez saisir vous-même des indications concernant le nom, la famille, la version et l'auteur.

En principe, vous pouvez éditer les parties d'un bloc de code dans n'importe quel ordre. Vous avez bien entendu aussi la possibilité de les corriger ou de les compléter.



Nota

Si vous souhaitez reprendre des mnémoniques de la table des mnémoniques, vous devriez d'abord vérifier qu'ils sont complets et les compléter au besoin.

6.3.1.2 Présélections pour l'éditeur de bloc CONT/LOG/LIST

Avant de commencer à programmer, vous devriez vous familiariser avec les possibilités de présélection qui vous faciliteront le travail.

La commande **Outils > Paramètres** ouvre une boîte de dialogue dans laquelle vous pouvez choisir à l'avance certaines caractéristiques de la programmation des blocs, par exemple :

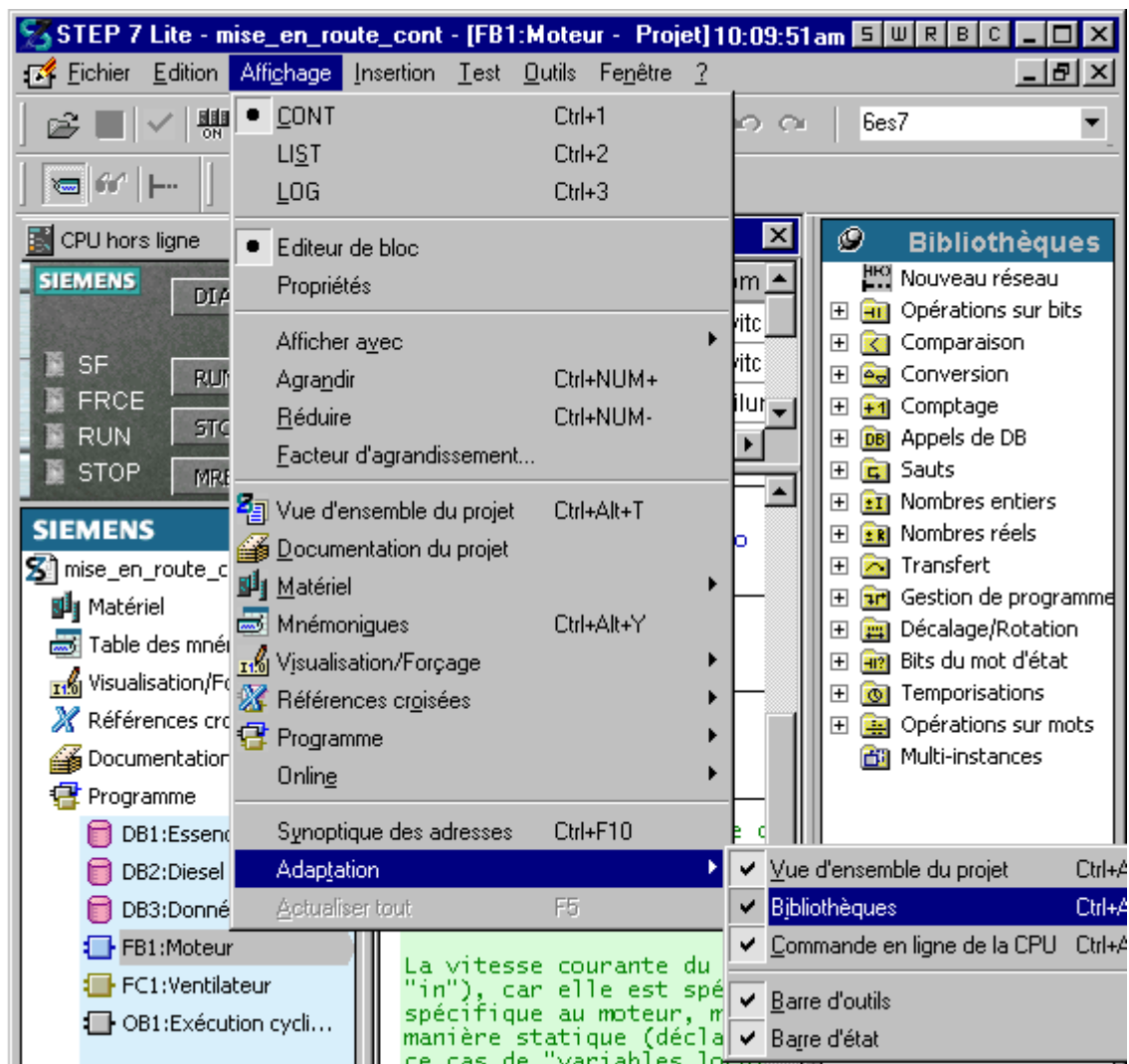
- la police (type et taille) utilisée pour le texte et les tables,
- si vous souhaitez voir s'afficher d'abord les mnémoniques et le commentaire pour un nouveau bloc,
- la couleur employée pour marquer les réseaux ou les lignes d'instruction sélectionnés.

Vous pouvez modifier les paramètres concernant le langage, le commentaire et les mnémoniques durant l'édition à l'aide des commandes **Affichage >**

6.3.1.3 Instructions tirées des bibliothèques d'opérations

Les bibliothèques d'opérations mettent à votre disposition des opérations CONT et LOG ainsi que des multi-instances déjà déclarées. Vous les appellerez en choisissant la commande **Affichage > Adapter > Bibliothèques**. Sélectionnez l'élément qui vous convient dans l'onglet "Opérations" et insérez-le à l'endroit sélectionné dans la section d'instructions soit avec la commande "Coller" du menu contextuel, soit par double-clic, soit par glisser-lâcher.

Exemple : bibliothèque d'opérations CONT



6.3.1.4 Choix de l'affichage dans la fenêtre d'édition

Réduction/agrandissement de l'affichage

Pour chaque fenêtre de travail (bloc de données ou bloc de code), vous pouvez réduire ou agrandir par paliers l'affichage, police de caractères comprise.

Procédez de la manière suivante :

1. Activez la fenêtre dont vous voulez réduire ou agrandir le contenu d'un palier.
2. Choisissez la commande **Affichage > Réduire** ou **Affichage > Agrandir**. L'affichage en cours est réduit ou agrandi d'un palier jusqu'à la taille minimale ou maximale.

Zoom

Pour chaque fenêtre de travail (bloc de données ou bloc de code), vous pouvez agrandir ou réduire l'affichage - y compris la police de caractères - ou lui rendre sa taille normale en indiquant le facteur voulu.

Procédez de la manière suivante :

1. Activez la fenêtre dans laquelle vous voulez modifier la taille du contenu représenté.
2. Choisissez la commande **Affichage > Facteur d'agrandissement...**
3. Indiquez la valeur qui vous convient dans la boîte de dialogue et confirmez par "OK".

6.3.1.5 Fractionnement de la fenêtre de travail

Les blocs de code s'affichent dans des fenêtres de travail divisées en deux parties. La moitié supérieure de la fenêtre contient la table de déclaration des variables et la moitié inférieure la section des instructions. Vous pouvez modifier la taille des deux parties.

Procédez de la manière suivante :

- Cliquez sur la barre de fractionnement et faites glisser le pointeur, en maintenant le bouton gauche de la souris enfoncé, dans la direction dans laquelle vous voulez positionner la barre de fractionnement.

6.3.1.6 Régler la largeur des colonnes

Vous pouvez régler la largeur des différentes colonnes de la table de déclaration des variables.

Procédez de la manière suivante :

- Positionnez le pointeur dans le titre d'une colonne sur la ligne verticale séparant deux colonnes afin qu'il prenne l'aspect d'une double flèche.
- Faites-le glisser horizontalement tout en maintenant le bouton gauche de la souris enfoncé, jusqu'à ce que la largeur de la colonne vous convienne.

Nota

Faites un double-clic sur le titre de la colonne pour en optimiser la largeur.

Passage d'un langage de programmation à un autre

Vous disposez des langages de programmation "liste d'instructions" (LIST), "logigramme" (LOG) et "schéma à contacts" (CONT) pour programmer vos blocs.

1. Le bloc étant ouvert, choisissez la commande **Affichage > CONT/LOG/LIST...** ou l'onglet "Propriétés".
2. Sélectionnez un langage de programmation dans la zone "Langage de création".

Nota

- Il est toujours possible de passer du langage de programmation CONT au langage de programmation LOG, et inversement.
- Il n'est possible de passer du langage de programmation LIST au langage de programmation CONT/LOG que pour les instructions LIST qui assurent la transmission intégrale des paramètres des éléments CONT/LOG correspondants, et ce dans l'ordre requis. Il faut affecter "NOP 0" aux paramètres non utilisés en LIST.

6.3.2 Edition de la table de déclaration des variables

6.3.2.1 Utilisation de la déclaration des variables dans les blocs de code

Lorsque vous ouvrez un bloc de code, la fenêtre qui s'affiche présente sa table de déclaration des variables dans la partie supérieure et la section des instructions, dans laquelle vous éditez le code du bloc proprement dit, dans la partie inférieure.

Exemple : table de déclaration des variables et section des instructions en LIST

Adresse	Décl.	Nom	Type	Valeur initiale	Commentaire
0.0	in	Switch_On	BOOL	FALSE	Mise en marche du moteur
0.1	in	Switch_Off	BOOL	FALSE	Arrêt du moteur
0.2	in	Failure	BOOL	FALSE	Défaillance du moteur provoquant
2.0	in	Actual_Speed	INT	0	Vitesse réelle du moteur
4.0	out	Engine_On	BOOL	FALSE	Le moteur se met en marche
4.1	out	Preset_Speed_Reached	BOOL	FALSE	Vitesse prescrite atteinte
	in_out				
6.0	stat	Preset_Speed	INT	1500	Vitesse de moteur prescrite


```

U      #Switch_On
UN     "Mode automatique"
S      #Engine_On
O      #Switch_Off
ON     #Failure
R      #Engine_On

```

Dans la table de déclaration des variables, vous définissez les variables locales y compris les paramètres formels du bloc. Ceci a, entre autres, les effets suivants :

- La déclaration sert à réserver l'espace mémoire approprié dans la pile des données locales pour les variables temporaires, dans le cas de blocs fonctionnels, et dans le DB d'instance associé ultérieurement pour les variables statiques.
- En définissant les paramètres d'entrée, de sortie, d'entrée/sortie, vous déterminez également "l'interface" pour l'appel du bloc dans le programme.
- Lorsque vous déclarez les variables dans un bloc fonctionnel, celles-ci (à l'exception des variables temporaires) déterminent également la structure de données pour chaque DB d'instance que vous affecterez au FB.

6.3.2.2 Interaction entre la table de déclaration des variables et la section des instructions

La table de déclaration des variables et la section des instructions dépendent étroitement l'une de l'autre pour les blocs de code, puisque la section des instructions utilise les noms de la table de déclaration des variables. Les modifications apportées à la table de déclaration se répercutent donc dans l'ensemble de la section des instructions.

Action dans la déclaration des variables	Réaction dans la section des instructions
Nouvelle entrée correcte	Dans le cas d'un code erroné, la variable non déclarée auparavant devient valide.
Changement de nom correct sans changement de type	Le mnémonique est immédiatement représenté partout avec le nouveau nom.
Nom correct changé en nom invalide	Le code n'est pas modifié.
Nom invalide changé en nom correct	Dans le cas d'un code erroné, celui-ci devient valide.
Changement de type	Un code erroné devient valide et un code correct peut devenir invalide le cas échéant.
Suppression d'une variable (mnémonique) utilisée dans le code	Un code correct devient invalide.

Les modifications de commentaire, la saisie erronée d'une nouvelle variable, la modification d'une valeur initiale ou la suppression d'une variable non utilisée n'ont aucun effet sur la section des instructions.

6.3.2.3 Structure de la table de déclaration des variables

La table de déclaration des variables contient des entrées pour l'adresse, le type de déclaration, le nom, le type de données, la valeur initiale et le commentaire des variables. Chaque ligne de la table correspond à une déclaration de variable. Les variables de type ARRAY ou STRUCT nécessitent plusieurs lignes.

Vous trouverez la liste des types de données autorisés pour les données locales des différents types de bloc dans Affectation de types de données aux données locales de blocs de code.

Colonne	Signification	Observation	Edition
Adresse	Adresse dans le format BYTE.BIT.	Pour les types de données nécessitant plus d'un octet, l'adresse indique l'affectation par un saut à l'adresse d'octet suivante. Légende des caractères : * : taille d'un élément de tableau en octets. + : adresse de début par rapport au début de la structure = : place mémoire totale d'une structure	Entrée système : le système affecte l'adresse et l'affiche lorsque vous clôturez la saisie d'une déclaration.
Nom	Mnémonique de la variable	Le nom doit commencer par une lettre. Les mots-clés réservés ne sont pas autorisés.	Obligatoire
Déclaration	Type de déclaration, "utilité" de la variable	Possibilités selon le type de bloc : paramètre d'entrée "in", paramètre de sortie "out", paramètre d'entrée/sortie "in_out", variable statique "stat", variable temporaire "temp".	Indiquée par le système selon le type de bloc
Type	Type de données de la variable (BOOL, INT, WORD, ARRAY etc.).	Vous pouvez sélectionner les types de données dans le menu appelé par le bouton droit de la souris.	Obligatoire
Valeur initiale	Valeur initiale si le logiciel ne doit pas utiliser la valeur par défaut	Doit être compatible avec le type de données. Elle est prise comme valeur actuelle de la variable lors du premier enregistrement du DB si vous n'affectez pas explicitement une valeur actuelle à cette variable.	Facultative
Commentaire	Commentaire pour la documentation		Facultative

Valeurs par défaut

Quand vous ouvrez un bloc de code que vous venez de créer, une table de déclaration des variables prédéfinie s'affiche. Elle contient la liste des seuls types de déclaration autorisés pour le type de bloc sélectionné et ce, dans l'ordre prescrit (in, out, in_out, stat, temp).

Quand vous créez un bloc d'organisation, la table de déclaration des variables prédéfinie contient les données locales prévues pour l'OB en question. Ces 20 octets renferment des informations de déclenchement standard fournies par le système d'exploitation. On y trouve non seulement des paramètres pour le comportement des blocs d'organisation, mais aussi des informations telles que la classe de priorité, le numéro de l'OB, des identificateurs pour les événements déclencheurs, etc. Elles sont inscrites durant l'exécution de l'OB et peuvent être lues, par exemple à des fins de diagnostic.

Colonnes non éditables dans la table de déclaration des variables

Colonne	Entrée
Adresse	C'est le système qui affecte l'adresse et l'affiche lorsque vous clôturez la saisie d'une déclaration.
Type de déclaration	Le type de déclaration est déterminé par la position de la déclaration dans la table. Ceci garantit que les variables ne peuvent être saisies que dans l'ordre correct des types de déclaration. Pour changer le type d'une déclaration, il faut d'abord couper la déclaration, puis la coller sous le nouveau type de déclaration.

6.3.2.4 Remarques générales sur les tables de déclaration de variables

Pour l'édition de la table, vous disposez des fonctions habituelles du menu **Edition**. L'édition est facilitée par le menu contextuel que vous appelez en cliquant avec le bouton droit de la souris. Pour la saisie du type de données, vous êtes également assisté par le menu contextuel.

Sélection dans la table de déclaration des variables

Pour sélectionner une ligne individuelle, cliquez sur la zone d'adresse correspondante qui est protégée en écriture. La sélection de lignes supplémentaires du même type de données s'effectue en maintenant la touche MAJ enfoncée. Les lignes sélectionnées apparaissent sur fond noir.

La sélection des ARRAY se fait par clic sur la zone d'adresse de la ligne concernée.

Pour **sélectionner une structure**, cliquez sur la zone d'adresse de la première ou de la dernière ligne (c'est-à-dire celle contenant les mots-clés STRUCT ou END_STRUCT). Pour sélectionner une déclaration individuelle dans une structure, cliquez sur la zone d'adresse correspondante de la ligne.

Lorsque vous imbriquez des structures, les noms de variable s'inscrivent en retrait pour représenter la hiérarchie.

Annulation d'actions

La commande **Edition > Annuler** vous permet d'annuler la dernière opération Couper ou Effacer effectuée dans la table de déclaration des variables.

6.3.2.5 Utilisation de la table de déclaration des variables

Insertion de lignes vides dans une table de déclaration de variables

Avant une ligne

1. Positionnez le pointeur dans cette ligne de la table.
2. Choisissez la commande **Insertion > Ligne avant sélection**.

Après une ligne

- Positionnez le pointeur dans la zone de commentaire de cette ligne et appuyez sur ENTREE ou
- choisissez la commande **Insertion > Ligne après sélection**.

6.3.2.6 Saisie de types de données simples dans la table de déclaration

Pour saisir une nouvelle déclaration, procédez de la manière suivante :

1. Entrez le nom de la variable derrière le type de déclaration souhaité.
2. Après la saisie, déplacez le curseur dans le champ adjacent à l'aide de la touche TAB.
3. Entrez de la même façon :
 - le type de données,
 - la valeur initiale (facultative),
 - le commentaire (facultatif).

Quand vous clôturez une ligne, une adresse est affectée à la variable.

Toute édition d'un champ du tableau est suivie d'une vérification de syntaxe qui repère en rouge les erreurs éventuelles. Vous n'êtes pas obligé de les corriger aussitôt, vous pouvez poursuivre l'édition et effectuer les corrections ultérieurement.

6.3.2.7 Saisie d'éléments de données du type STRUCT

1. Pour déclarer le type de données, vous avez plusieurs possibilités :
 - Positionnez le pointeur dans la cellule de la colonne "Type" et choisissez la commande **Insertion > Type de données > Complexe > STRUCT**.
 - Sélectionnez la cellule de la colonne "Type" et appuyez sur le bouton droit de la souris. Choisissez alors le type de données en question dans le menu contextuel.
 - Tapez le mot-clé STRUCT dans la cellule de la colonne "Type".
2. Tapez un nom dans la colonne "Nom" et clôturez la saisie dans cette ligne de la table avec la touche TAB ou ENTREE. Une ligne vide et la ligne de fin de la déclaration (END_STRUCT) s'affichent alors.

3. Saisissez les éléments de la structure en indiquant leur nom, type de données, valeur initiale (facultative) et, éventuellement, un commentaire. Vous obtenez d'autres lignes par les commandes du menu "Insertion" ou en appuyant sur la touche ENTREE ; vous pouvez aussi copier ou effacer des variables via le menu "Edition".

Adresse	Décl.	Nom	Type	Valeur initiale	Commentaire
0.0	in	New_Struct	STRUCT		
+0.0	in	var1	BOOL		
+2.0	in	var2	INT		
+4.0	in	var3	WORD		
=6.0	in		END_STRUCT		

Saisie du type de données ARRAY dans la table de déclaration

1. Positionnez le pointeur dans la cellule de la colonne "Type" dans la table de déclaration des variables.
2. Choisissez la commande **Insertion > Type de données > Complexe > ARRAY**. ARRAY s'inscrit alors dans la cellule sélectionnée. Vous pouvez aussi taper le mot ARRAY directement au clavier.
3. Tapez directement derrière ARRAY un crochet ouvrant, les limites inférieures de l'indice, deux points, les limites supérieures de l'indice et un crochet fermant, par ex. ARRAY[1..14] pour un tableau à une dimension ou ARRAY[1..20,1..24] pour un tableau à deux dimensions.
4. Dans la cellule de la colonne "Valeur initiale", vous pouvez entrer les valeurs initiales pour les différents éléments du tableau (voir exemples ci-après).
5. Dans la cellule de la colonne "Commentaire", vous pouvez noter des remarques sur le tableau.
6. Clôturez la saisie dans cette ligne de la table avec la touche TAB ou ENTREE.
7. Dans la seconde ligne automatiquement créée, tapez le type de données des éléments ARRAY.

Exemples de saisie de valeurs initiales dans les tableaux

- Attribution individuelle :
Vous donnez une valeur initiale propre à chaque élément. Les valeurs sont séparées par des virgules dans la liste.
- Facteur de répétition :
Vous donnez la même valeur initiale à plusieurs éléments. La valeur est indiquée entre parenthèses et précédée du facteur de répétition correspondant au nombre d'éléments.

Type	Valeur initiale	Explication
ARRAY[1..14]	1234	La valeur initiale 1234 est affectée au premier élément du tableau uniquement.
ARRAY[1..14]	1234, 56, 78, 90	Les valeurs initiales 1234, 56, 78 et 90 sont affectées dans cet ordre aux quatre premiers éléments du tableau.
ARRAY[1..14]	14 (9876)	La valeur initiale 9876 est affectée aux quatorze éléments du tableau.

6.3.2.8 Copie de variables dans une table de déclaration

1. Sélectionnez les variables à copier :
 - en cliquant sur la zone "Adresse" (sélection d'une variable),
 - en cliquant à nouveau sur la zone "Adresse" d'une autre variable tout en maintenant la touche MAJ enfoncée : toutes les lignes situées entre la première sélection et la seconde sont ainsi sélectionnées (sélection de plusieurs variables).
2. Choisissez la commande **Edition > Copier** ou l'icône correspondante dans la barre d'outils.
3. Positionnez le pointeur à l'endroit après lequel vous voulez insérer la variable copiée et choisissez la commande **Edition > Coller** ou l'icône correspondante de la barre d'outils.

Les variables copiées sont alors insérées. Pour que les noms des variables restent univoques, un numéro d'ordre est automatiquement ajouté à ceux des variables copiées.

6.3.2.9 Suppression de variables dans une table de déclaration

1. Sélectionnez les variables à supprimer :
 - en cliquant sur la zone "Adresse" (sélection d'une variable),
 - en cliquant à nouveau sur la zone "Adresse" d'une autre variable tout en maintenant la touche MAJ enfoncée : toutes les lignes situées entre la première sélection et la seconde sont ainsi sélectionnées (sélection de plusieurs variables).
2. Choisissez la commande **Edition > Couper** ou la commande **Edition > Effacer** ou les icônes correspondantes dans la barre d'outils.

Nota

Important lors de la suppression de ARRAY et STRUCT :

- Quand vous sélectionnez la première ligne d'un ARRAY pour l'effacer, la deuxième ligne correspondante se trouve sélectionnée elle aussi.
 - Quand vous sélectionnez la première ligne d'une STRUCT pour l'effacer, toutes les lignes jusqu'à END_STRUCT sont sélectionnées elles aussi.
-

Largeur variable des colonnes

Vous pouvez donner des largeurs variées aux colonnes de la table. Procédez de la manière suivante :

- Positionnez le pointeur dans le titre d'une colonne sur la ligne verticale séparant deux colonnes afin qu'il prenne l'aspect d'une double flèche.
- Faites-le glisser horizontalement tout en maintenant le bouton gauche de la souris enfoncé, jusqu'à ce que la largeur de la colonne vous convienne.

Si vous ne comptez pas saisir les commentaires ou les valeurs initiales (qui sont facultatifs), vous pouvez modifier ces colonnes de cette manière et vous concentrer entièrement sur les autres.

6.3.3 Multi-instances dans la table de déclaration des variables

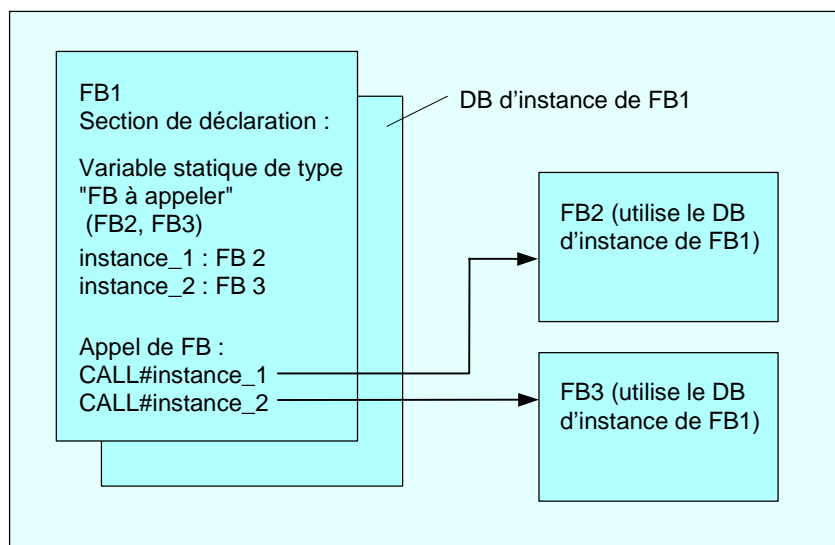
6.3.3.1 Utilisation de multi-instances

Il est probable qu'en raison des performances des CPU S7 utilisées (par ex. espace mémoire), vous ne pourrez ou ne souhaitez attribuer aux données d'instance qu'un nombre restreint de blocs de données. Lorsque vous appelez dans un FB de votre programme utilisateur d'autres blocs fonctionnels déjà existants (hiérarchie d'appel des FB), vous pouvez le faire sans leur associer un DB d'instance en propre (c'est-à-dire sans générer de DB d'instance supplémentaires).

Procédez pour cela de la manière suivante :

- Entrez les FB à appeler comme variables statiques dans la déclaration des variables du FB appelant.
- Dans ce bloc fonctionnel, appelez d'autre blocs fonctionnels sans DB d'instance en propre (c'est-à-dire en plus).
- Vous regroupez ainsi les données d'instance dans un seul DB d'instance, ce qui revient à mieux utiliser le nombre de DB disponibles.

L'exemple suivant illustre la solution décrite : FB2 et FB3 utilisent le DB d'instance du bloc fonctionnel FB1 qui les appelle.



Seule condition : vous devez "faire savoir" au bloc fonctionnel appelant quelles instances vous appelez et de quel type (de FB) elles sont. Ces indications doivent être faites dans la section de déclaration du FB appelant. Le FB à utiliser doit posséder au moins une variable ou un paramètre de la zone de données (c'est-à-dire pas VAR_TEMP).

N'utilisez pas de blocs de données multi-instance tant qu'il faut compter avec des modifications en ligne quand la CPU est en marche. Un chargement a posteriori sans à-coups n'est garanti que si vous utilisez des blocs de données d'instance.

6.3.3.2 Règles de formation des multi-instances

Les règles suivantes s'appliquent à la déclaration des multi-instances :

- Pour déclarer des multi-instances, il faut que le bloc fonctionnel ait été créé comme FB admettant les multi-instances.
- Il faut associer un DB d'instance au bloc fonctionnel dans lequel une multi-instance est déclarée.
- Une multi-instance ne peut être déclarée que comme variable statique (type de déclaration "stat").

Nota

Vous pouvez également créer des multi-instances pour les blocs fonctionnels système.

6.3.3.3 Saisie de la multi-instance dans la table de déclaration des variables

1. Ouvrez le FB à partir duquel les FB imbriqués doivent être appelés.
2. Définissez, dans la table de déclaration des variables du FB appelant, une variable statique pour chaque appel d'un bloc fonctionnel pour l'instance duquel vous ne voulez pas préciser de DB.
 - Positionnez-vous dans la deuxième colonne d'une ligne vide avec la déclaration "stat".
 - Dans la colonne "Nom", tapez une désignation pour l'appel du FB derrière le type de déclaration "stat".
 - Dans la colonne "Type", tapez le FB à appeler sous forme d'adresse absolue ou sous forme de mnémonique.
 - Vous pouvez saisir d'éventuelles explications dans la colonne de commentaire.

Appels dans la section des instructions

Si vous avez déclaré des multi-instances, vous pouvez utiliser des appels de FB sans indication d'un DB d'instance.

Exemple : la variable statique "Nom : Moteur_1, Type de données : FB20" étant définie, l'instance peut être appelée de la manière suivante :

```
CALL Moteur_1      // Appel du FB20 sans DB d'instance
```

6.3.4 Remarques générales sur la saisie d'instructions et de commentaires

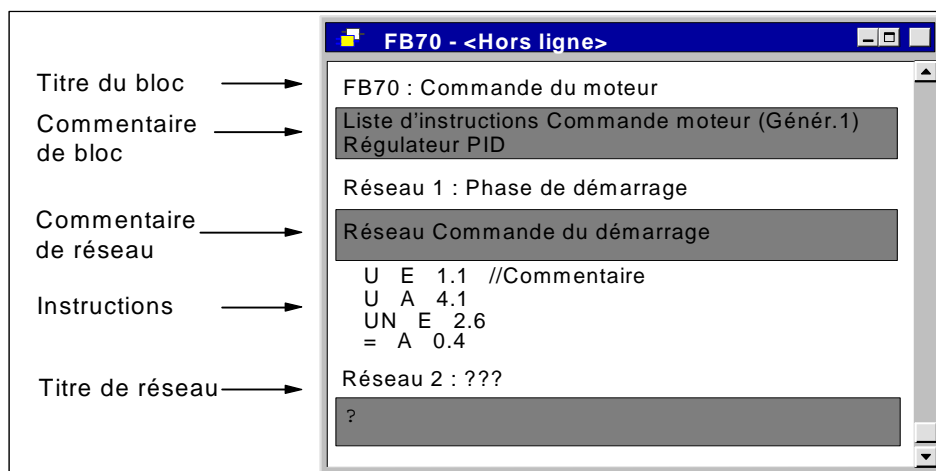
6.3.4.1 Structure de la section des instructions

Vous décrivez l'exécution du programme de votre bloc de code dans la section des instructions. Vous écrivez à cet effet les instructions correspondantes dans des réseaux, en fonction du langage de programmation activé. L'éditeur de bloc vérifie la syntaxe immédiatement après la saisie d'une instruction et vous signale les erreurs par des caractères rouges en italique.

Dans la plupart des cas, la section des instructions d'un bloc de code comporte plusieurs réseaux, eux-mêmes composés d'une liste d'instructions.

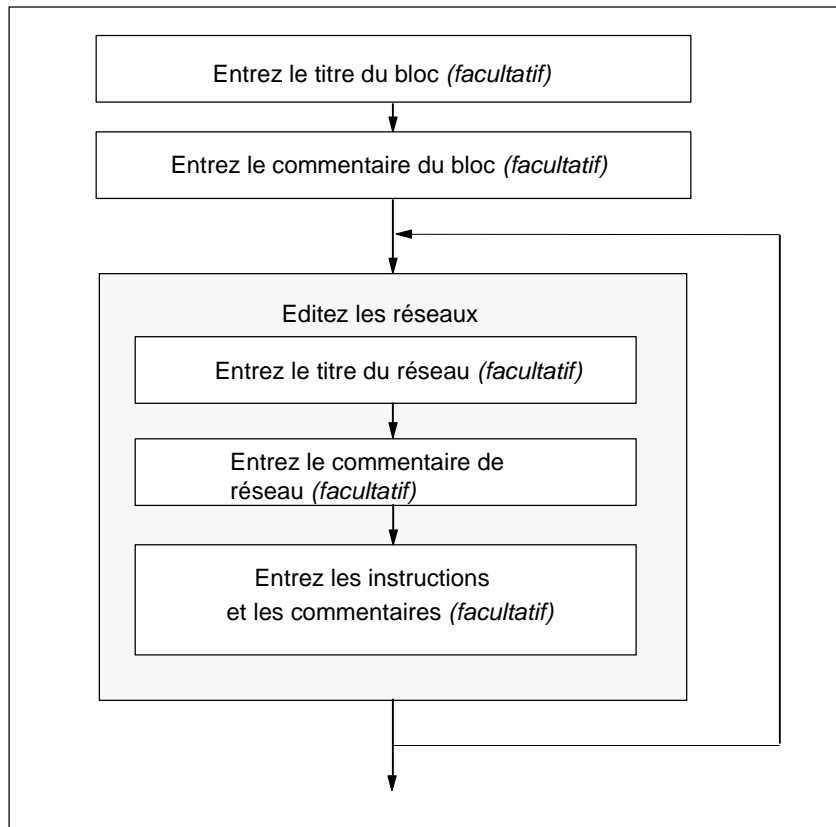
Vous pouvez éditer le titre de bloc, les commentaires de bloc, les titres de réseaux, les commentaires de réseaux ainsi que les lignes d'instructions dans les réseaux d'une section d'instructions.

Exemple en langage de programmation LIST montrant la structure de la section des instructions



6.3.4.2 Marche à suivre pour saisir des instructions

De manière générale, vous pouvez éditer les éléments de la section des instructions dans un ordre quelconque. Quand vous programmez un bloc pour la première fois, il est recommandé de procéder dans l'ordre suivant :



Vous pouvez effectuer les modifications en mode d'insertion et de substitution. Vous passez d'un mode à l'autre avec la touche INSERT.

6.3.4.3 Saisie de mnémoniques globaux dans un programme

Avec la commande **Insertion > Mnémonique**, vous pouvez insérer des mnémoniques dans la section des instructions de votre programme. Lorsque le curseur est positionné au début, à la fin ou à l'intérieur d'une chaîne de caractères, le mnémonique commençant par cette chaîne de caractères est déjà sélectionné - si un tel mnémonique existe. Si vous modifiez la chaîne de caractères, la sélection est reprise dans la liste.

Les séparateurs de début et de fin d'une chaîne de caractères sont par exemple le caractère d'espacement, le point et le double point. Les séparateurs ne sont pas interprétés dans les mnémoniques globaux.

Pour insérer des mnémoniques, vous pouvez procéder de la manière suivante :

1. Saisissez les caractères de début du mnémonique souhaité dans le programme.
2. Appuyez simultanément sur la touche CTRL et sur la touche J, pour afficher la liste des mnémoniques. Le premier mnémonique commençant par les caractères de début saisis est déjà sélectionné.
3. Validez le mnémonique en appuyant sur la touche ENTREE ou sélectionnez un autre mnémonique.

Le mnémonique entre guillemets remplace alors les caractères de début.

De manière générale : lorsque, lors de l'insertion d'un mnémonique, le curseur se trouve au début, à la fin ou à l'intérieur d'une chaîne de caractères, cette chaîne de caractères est remplacée par le mnémonique entre guillemets.

6.3.4.4 Titres et commentaires de blocs et de réseaux

Les commentaires améliorent la lisibilité de votre programme utilisateur et facilitent ainsi la mise en service et la recherche des erreurs éventuelles. Vous devez absolument y faire appel, car ils constituent une part importante de la documentation du programme.

Commentaires pour les programmes CONT, LOG et LIST

Vous disposez des commentaires suivants :

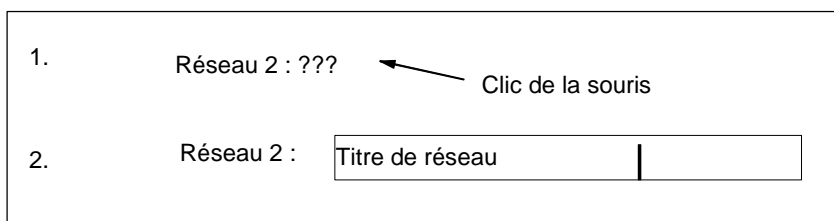
- Titre de bloc : titre d'un bloc de 64 caractères au maximum.
 - Commentaire de bloc : documentation du bloc de code entier, indique par exemple la fonction du bloc.
 - Titre de réseau : titre d'un réseau de 64 caractères au maximum.
 - Commentaire de réseau : documentation de la fonction des différents réseaux.
 - Colonne de commentaire de la table de déclaration des variables : commentaires pour les données locales déclarées.
 - Commentaire de mnémonique : commentaires ayant été saisis pour un opérande lors de la définition du nom dans la table des mnémoniques.
- Vous pouvez afficher ces commentaires en choisissant la commande **Affichage > Afficher avec > Commentaire**.

Vous pouvez saisir le titre de bloc, les titres de réseaux ainsi que les commentaires de bloc et les commentaires de réseaux dans la section des instructions d'un bloc de code.

Titre de bloc ou de réseau

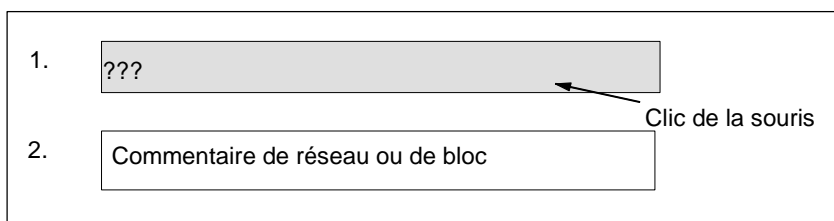
Pour saisir un titre de bloc ou de réseau, positionnez le curseur sur les trois points d'interrogation à droite du nom de bloc ou de réseau (par ex. Réseau 1 : ???). Une zone de texte dans laquelle vous pouvez entrer le titre s'ouvre. Ce titre peut comporter 64 caractères au maximum.

Le commentaire d'un bloc se rapporte au bloc de code entier : vous pouvez en décrire la fonction. Quant aux commentaires de réseaux, ils font référence aux réseaux individuels et décrivent donc les caractéristiques de chacun.



Commentaires de blocs et de réseaux

Vous pouvez activer/désactiver l'affichage des zones de commentaires grises en choisissant la commande **Affichage > Afficher avec > Commentaire**. Un double clic sur une telle zone de commentaires ouvre une zone de texte dans laquelle vous pouvez saisir vos explications. Vous disposez de 64 Ko par bloc pour les commentaires de blocs et de réseaux.



6.3.4.5 Saisie de commentaires de blocs et de réseaux

1. Activez - commande cochée - l'affichage des commentaires via la commande **Affichage > Afficher avec > Commentaire**.
2. Positionnez le pointeur par un clic de la souris dans la zone grise sous le nom de bloc ou de réseau. Un cadre délimite maintenant la zone de commentaire qui est passée du gris au blanc.
3. Saisissez le commentaire dans la zone de texte ouverte. Vous disposez de 64 Ko en tout par bloc pour les commentaires de blocs et de réseaux.
4. Quittez la zone de texte par un clic de la souris en dehors de la zone de texte, via la touche TAB ou via la combinaison de touches MAJ + TAB.
5. Désactivez éventuellement l'affichage des commentaires via la commande **Affichage > Afficher avec > Commentaire**.

6.3.4.6 Fonction de recherche d'erreurs dans la section des instructions

Puisqu'elles sont représentées en rouge, les erreurs peuvent être localisées aisément dans la section des instructions. Pour passer facilement aux erreurs se trouvant hors de la zone visible, l'éditeur de bloc propose les deux fonctions de recherche **Edition > Aller à > Erreur précédente/suivante**.

Cette recherche se fait par-delà les réseaux, c'est-à-dire dans toute la section des instructions et pas uniquement dans le réseau en cours ou dans la zone actuellement visible.

La barre des résultats affiche des informations sur les erreurs.

Vous avez également la possibilité de corriger des erreurs et d'effectuer des modifications en mode de substitution. Vous passez du mode d'insertion au mode de substitution et inversement avec la touche INSERT.

6.3.5 Edition d'instructions CONT dans la section des instructions

6.3.5.1 Paramètres pour le langage de programmation CONT

Paramétrage de la mise en page pour CONT

Vous pouvez définir la mise en page pour la programmation en langage CONT. Le format choisi (format vertical A4, format horizontal A4, taille maximale) a une influence sur le nombre d'éléments CONT représentables dans une branche.

1. Choisissez la commande **Outils > Paramètres**.
2. Dans la boîte de dialogue qui s'affiche, sélectionnez le format qui vous convient dans la zone de liste "Mise en page" de la partie "CONT/LOG". Indiquez la taille de format désirée.

Paramètres pour l'impression

Si vous désirez imprimer la section des instructions CONT, nous vous conseillons de choisir le format de page approprié avant même d'écrire la section des instructions.

Paramètres de base sous Outils > Paramètres...

La commande **Outils > Paramètres** vous permet de définir des paramètres généraux tels que le format de la mise en page ou la largeur du champ de l'opérande.

6.3.5.2 Règles de saisie des instructions CONT

La description du langage "CONT" figure dans le manuel *CONT pour S7-300/400 – Programmation de blocs* ou dans l'aide en ligne sur CONT.

Un réseau CONT peut être composé de plusieurs éléments dans plusieurs branches. Tous les éléments et branches doivent être reliés entre eux, la barre conductrice gauche n'étant pas considérée comme une liaison (CEI 1131-3).

Vous devez observer quelques règles lors de la programmation en CONT. Des messages vous signaleront d'éventuelles erreurs.

Terminaison d'un réseau CONT

Tout réseau CONT doit posséder une terminaison sous forme de bobine ou de boîte. Vous ne pouvez cependant pas utiliser les éléments CONT suivants comme terminaison de réseau :

- boîtes de comparaison
- bobines pour connecteurs `_(#)_/`
- bobines pour le traitement de fronts positifs `_(P)_/` ou négatifs `_(N)_/`

Placement de boîtes

Le point de départ de la branche pour la connexion d'une boîte doit toujours être la barre conductrice gauche. La branche précédant la boîte peut cependant contenir des fonctions logiques ou d'autres boîtes.

Placement de bobines

Les bobines sont automatiquement placées à l'extrémité droite du réseau, où elles forment la terminaison d'une branche.

Exceptions : les bobines pour connecteurs `_(#)_/` et les traitements de front positifs `_(P)_/` ou négatifs `_(N)_/` ne peuvent être placés ni complètement à gauche, ni complètement à droite dans la branche. Ils ne sont pas non plus autorisés dans les branches parallèles.

Il existe des bobines pour lesquelles une fonction logique booléenne est requise et d'autres qui l'interdisent.

- Bobines nécessitant une fonction logique :
 - sortie `_()`, mise à 1 sortie `_(S)`, remise à 0 sortie `_(R)`
 - connecteur `_(#)_/`, front positif `_(P)_/`, front négatif `_(N)_/`
 - toutes les bobines pour compteurs et temporisations
 - sauts si 0 `_(JMPN)`
 - activation relais de masquage `_(MCR<)`
 - chargement du résultat logique dans le registre RB `_(SAVE)`
 - retour saut `_(RET)`
- Bobines interdisant une fonction logique :
 - début de relais de masquage `_(MCRA)`
 - fin de relais de masquage `_(MCRD)`

- ouverture bloc de données _/(OPN)
- désactivation relais de masquage _/(MCR>)

Pour toutes les autres bobines, les fonctions logiques sont autorisées sans être requises.

Vous **ne devez pas** utiliser les bobines suivantes comme sortie parallèle :

- sauts si 0 _/(JMPN)
- sauts si 1 _/(JMP)
- appel de bloc _/(CALL)
- retour _/(RET)

Entrée de validation / sortie de validation

La connexion de l'entrée de validation "EN" ou de la sortie de validation "ENO" de boîtes est possible mais pas impérativement requise.

Suppression et modification

Lorsqu'une branche n'est composée que d'un élément, la suppression de cet élément supprime la branche entière.

La suppression d'une boîte entraîne également la suppression de toutes les branches reliées avec les entrées booléennes de cette boîte, à l'exception de la branche principale.

Pour le simple échange d'éléments du même type, vous disposez du mode de substitution.

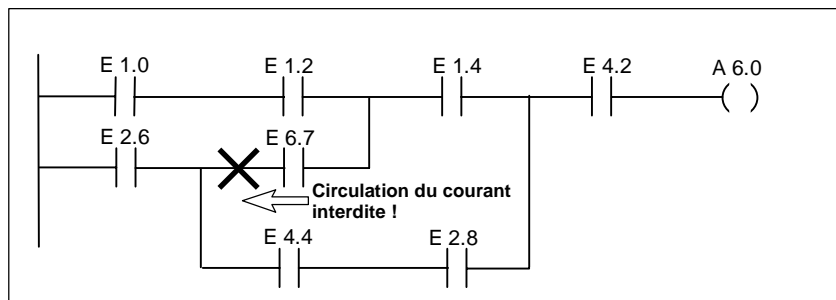
Branches parallèles

- Insérez les branches OU de la gauche vers la droite.
- Les branches parallèles s'ouvrent vers le bas et se ferment vers le haut.
- Les branches parallèles s'ouvrent toujours après l'élément CONT sélectionné.
- Les branches parallèles se ferment toujours après l'élément CONT sélectionné.
- Pour effacer une branche parallèle, vous devez effacer tous les éléments CONT qu'elle contient. La suppression du dernier élément CONT de la branche entraîne également l'effacement du reste de celle-ci.

6.3.5.3 Branchements interdits en CONT

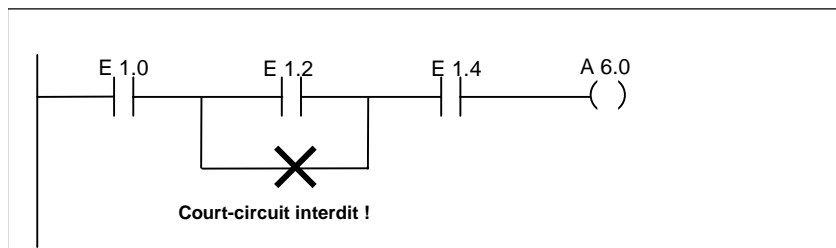
Circulation du courant de la droite vers la gauche

Vous ne pouvez pas éditer de branches qui provoqueraient une circulation inverse du courant. La figure suivante en montre un exemple. Si E 1.4 a l'état de signal "0", E 6.7 entraînerait une circulation du courant de la droite vers la gauche. Ceci est interdit.



Court-circuit

Vous ne pouvez pas éditer de branches qui entraînent un court-circuit. La figure suivante en montre un exemple :



6.3.5.4 Saisie d'éléments CONT

Comment saisir des éléments CONT

1. Sélectionnez, dans un réseau, l'endroit derrière lequel un élément CONT doit être inséré.
2. Pour insérer l'élément désiré :
 - cliquez sur l'icône correspondante de la barre d'outils (pour contact à fermeture, contact à ouverture),
 - appuyez sur les touches de fonction correspondantes F2 ou F3 (pour contact à fermeture, contact à ouverture),
 - faites un double-clic sur l'élément sélectionné dans la bibliothèque d'opérations ou amenez-le par glisser-lâcher dans l'éditeur de bloc.

L'élément CONT sélectionné est alors inséré, avec "???" ou "..." pour représenter les adresses et les paramètres.

Nota

Vous pouvez également éditer la section des instructions en sélectionnant des éléments CONT y figurant déjà et en choisissant les commandes **Edition > Couper**, **Edition > Copier** ou **Edition > Coller**.

Saisie et édition d'adresses ou de paramètres dans des éléments CONT

Lors de l'insertion d'un élément CONT, les chaînes de caractères "???" et "..." sont utilisées pour réserver l'emplacement des adresses ou des paramètres.

La chaîne de caractères "???" en rouge signale les adresses et paramètres devant être définis.

La chaîne de caractères "..." en noir signale les adresses et paramètres pouvant être définis.

1. Positionnez le pointeur sur l'adresse ou le paramètre non défini avec un clic de la souris ou avec la touche TAB.
2. Remplacez les points ou les points d'interrogation par l'adresse ou le paramètre (adressage direct ou indirect). Si l'affichage avec sélection des mnémoniques est activé (commande **Affichage > Afficher avec > Sélection de mnémoniques**), la liste des mnémoniques définis s'affiche. Le mnémonique commençant par les caractères saisis est sélectionné et peut être repris via la touche ENTREE.
3. Appuyez sur la touche ENTREE.
 - Si la syntaxe est correcte, l'adresse s'affiche formatée en noir et l'éditeur de bloc ouvre automatiquement la zone de texte suivante qui ne contient pas encore d'adresse ou de paramètre.
 - Si la syntaxe est erronée, l'éditeur ne quitte pas le champ de saisie et le message d'erreur correspondant s'affiche dans la barre d'état. Si vous appuyez encore une fois sur la touche ENTREE, le champ de saisie se ferme et le texte erroné apparaît en rouge et en italique.

Remplacement d'adresses ou de paramètres dans les éléments CONT

1. Appuyez sur la touche INSERT pour activer le mode de substitution. Le mode en vigueur est indiqué dans la barre d'état en bas à droite.
2. Positionnez le pointeur sur la zone de texte de l'adresse ou du paramètre avec un clic de la souris ou avec la touche TAB.
3. Remplacez l'adresse ou le paramètre.
4. Appuyez sur la touche ENTREE.
 - Si la syntaxe est correcte, l'adresse s'affiche formatée en noir et l'éditeur de bloc ouvre automatiquement la zone de texte suivante qui ne contient pas encore d'adresse ou de paramètre.
 - Si la syntaxe est erronée, l'éditeur ne quitte pas le champ de saisie et le message d'erreur correspondant s'affiche dans la barre d'état. Si vous appuyez encore une fois sur la touche ENTREE, le champ de saisie se ferme et le texte erroné apparaît en rouge et en italique.

Remplacement d'éléments CONT

Le mode de substitution vous permet de permuter facilement des éléments CONT de même type. L'avantage est que vous n'avez pas besoin de saisir à nouveau les adresses et paramètres. Il n'est possible de substituer à un élément CONT qu'un élément CONT de même type. Vous pouvez, par exemple, remplacer un contact à fermeture par un contact à ouverture, une bascule "mise à 0, mise à 1" par une bascule "mise à 1, mise à 0", une temporisation ou un compteur par une autre temporisation ou compteur.

1. Appuyez sur la touche INSERT pour activer le mode de substitution. Le mode en vigueur est indiqué dans la barre d'état en bas à droite.
2. Sélectionnez l'élément CONT à remplacer.
3. Pour insérer l'élément désiré :
 - cliquez sur l'icône correspondante dans la barre d'outils (pour contact à fermeture, contact à ouverture, bobine),
 - appuyez sur les touches de fonction correspondantes F2, F3 ou F4 (pour contact à fermeture, contact à ouverture),
 - faites un double-clic sur l'élément sélectionné dans la bibliothèque d'opérations ou amenez-le par glisser-lâcher dans l'éditeur de bloc.

L'élément CONT sélectionné auparavant est remplacé par le nouveau.

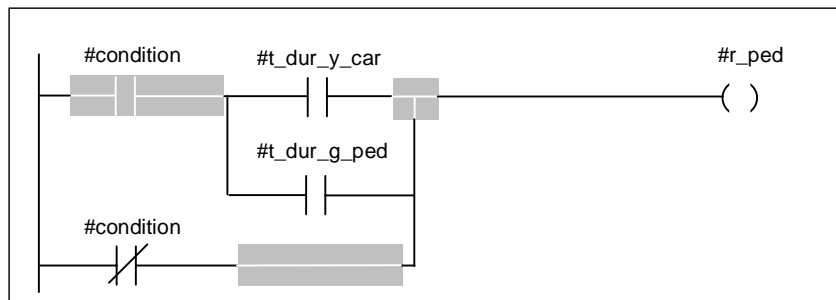
- Réactivez éventuellement le mode d'insertion en appuyant sur la touche INSERT. Le mode en vigueur est indiqué dans la barre d'état en bas à droite.

Sélection dans les réseaux CONT

Vous parvenez dans un réseau CONT en cliquant sur l'un des éléments qui s'y trouvent. En règle générale, vous pouvez sélectionner trois domaines au sein d'un réseau par clic simple de la souris :

- les éléments CONT, par exemple un contact ou une boîte,
- les points de jonction,
- les éléments vides (ligne ou branche ouverte).

Vous ne pouvez sélectionner qu'un domaine à la fois. La figure suivante montre des exemples de sélection, cependant plusieurs domaines ont été représentés simultanément.



Vous pouvez choisir vous-même la couleur de la sélection dans la boîte de dialogue "CONT/LOG/LIST". Vous parvenez à cette boîte de dialogue via la commande **Outils > Paramètres....**

Insertion de réseaux CONT supplémentaires

Pour créer un nouveau réseau :

- Choisissez la commande **Insertion > Réseau**.
- Cliquez sur l'icône correspondante de la barre d'outils.
- Faites un double-clic sur "Nouveau réseau" dans la bibliothèque d'opérations ou amenez cet élément par glisser-lâcher dans l'éditeur de bloc.
- Choisissez la commande Réseau dans le menu contextuel.
Positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler ce menu contextuel.

Le nouveau réseau est alors inséré sous le réseau sélectionné. Il ne comporte qu'une branche.

Si vous saisissez plus d'éléments que l'écran ne peut en afficher, le réseau se décalera vers la gauche de l'écran. Les commandes **Affichage > Réduire/Agrandir/Facteur d'agrandissement...** vous permettent d'adapter la représentation afin d'avoir une meilleure vue d'ensemble.

Vous parvenez dans un réseau CONT en cliquant sur l'un des éléments qui s'y trouvent. En règle générale, le clic permet de sélectionner trois domaines différents au sein d'un réseau.

Création de branches parallèles dans les réseaux CONT

Dans le langage de programmation CONT, vous réalisez les fonctions logiques OU en créant des branches parallèles.

Procédez de la manière suivante :

1. Sélectionnez l'élément avant lequel vous voulez ouvrir la branche parallèle.
2. Pour ouvrir la branche parallèle :
 - choisissez la commande **Ouvrir branche** dans le menu contextuel (positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler le menu contextuel)
 - ou appuyez sur la touche F8
 - ou cliquez sur l'icône correspondante de la barre d'outils.
3. Insérez les éléments CONT désirés dans la branche parallèle ouverte.
4. Sélectionnez, dans la branche principale, l'élément CONT après lequel vous voulez fermer la branche parallèle.
5. Pour fermer la branche parallèle :
 - choisissez la commande **Fermer branche** dans le menu contextuel (positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler le menu contextuel)
 - ou appuyez sur la touche F9
 - ou cliquez sur l'icône correspondante de la barre d'outils.

6.3.5.5 Création de nouvelles branches dans les réseaux CONT

Vous pouvez ouvrir plusieurs branches dans un réseau CONT.

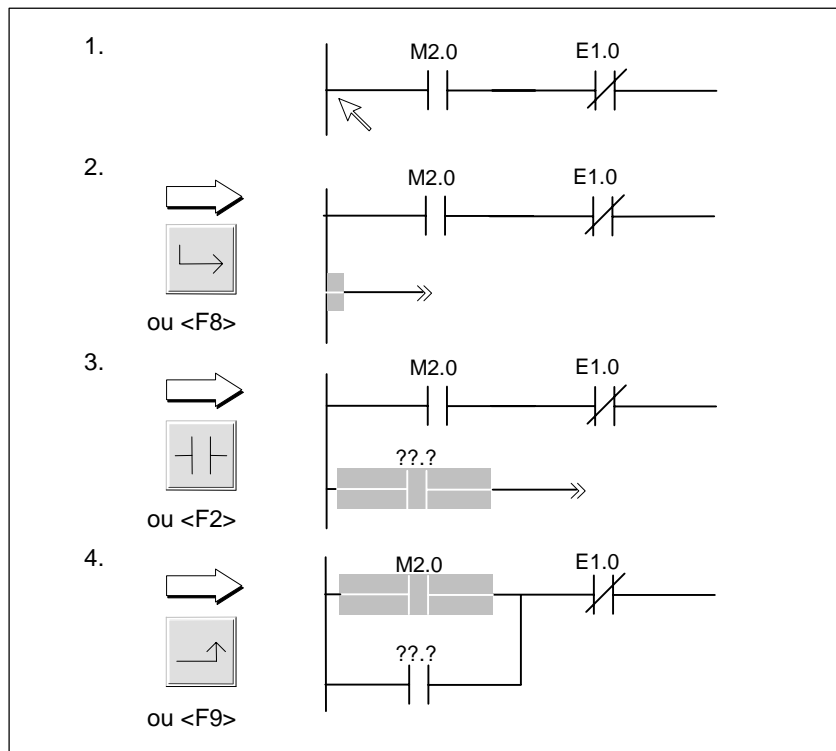
1. Sélectionnez avec le pointeur le point de départ de la branche sous laquelle vous voulez insérer une nouvelle branche.
2. Pour ouvrir la nouvelle branche :
 - choisissez la commande **Ouvrir branche** dans le menu contextuel (positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler le menu contextuel)
 - ou appuyez sur la touche F8
 - ou cliquez sur l'icône correspondante de la barre d'outils.

Création d'une branche fermée dans les réseaux CONT

Pour créer une branche fermée, procédez de la manière suivante :

1. Sélectionnez l'élément avant lequel vous voulez ouvrir la branche parallèle.
2. Ouvrez la branche parallèle avec F8.
3. Insérez l'opération CONT.
4. Fermez la branche parallèle avec F9.

La figure suivante montre comment créer une branche en utilisant exclusivement des touches de fonction ou des icônes de la barre d'outils.

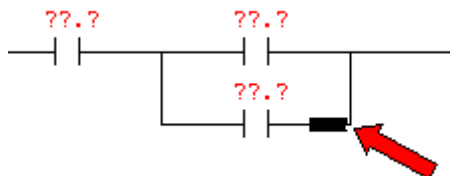


Lorsque vous fermez une branche parallèle, les éléments vides requis sont complétés. Quand cela s'avère nécessaire, les branches sont disposées de sorte que les croisements soient évités. Si vous fermez la branche directement depuis la branche parallèle, elle sera fermée après l'élément CONT le plus proche possible.

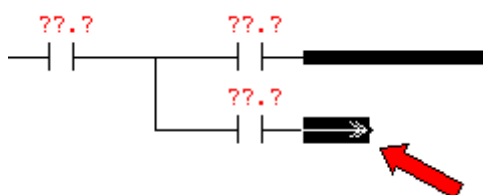
Ouverture de branches parallèles fermées dans les réseaux CONT

Vous pouvez ouvrir une branche parallèle fermée de la manière suivante :

1. Sélectionnez la branche parallèle à l'endroit indiqué avant le point de jonction entre la branche parallèle et la branche principale.



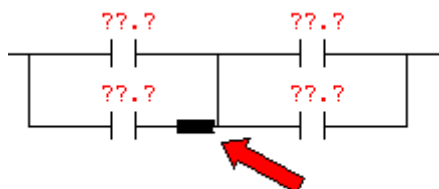
2. Appuyez sur la touche DEL (Eff). Vous pouvez insérer ensuite de nouveaux éléments CONT à l'endroit ouvert.



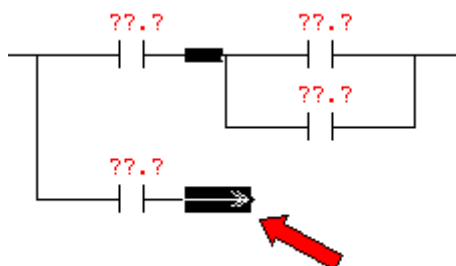
Décomposition d'une jonction dans les réseaux CONT

Lorsque, à un endroit d'un réseau CONT, une branche parallèle se ferme et une autre s'ouvre simultanément, on parle de jonction. Il est possible de décomposer une telle jonction de la manière suivante :

1. Sélectionnez la jonction à l'endroit indiqué dans le réseau.



2. Appuyez sur la touche DEL (Eff). Vous pouvez insérer ensuite de nouveaux éléments CONT à l'endroit ouvert.



3. Pour insérer un élément CONT :
 - cliquez sur l'icône correspondante dans la barre d'outils (pour contact à fermeture, contact à ouverture)
 - ou appuyez sur la touche de fonction correspondante F2 ou F3 (pour contact à fermeture, contact à ouverture)
 - ou bien faites un double-clic sur l'élément sélectionné dans la bibliothèque d'opérations ou amenez-le par glisser-lâcher dans l'éditeur de bloc.

6.3.5.6 Création de branches dans les réseaux CONT

La commande **Ouvrir branche** du menu contextuel ouvre une branche parallèle sans bobine, en commençant avant l'élément de programme sélectionné. Vous pouvez insérer d'autres éléments logiques dans la nouvelle branche.

1. Sélectionnez avec le pointeur l'élément de programme avant lequel vous voulez insérer une nouvelle branche.
2. Pour ouvrir la nouvelle branche :
 - choisissez la commande **Ouvrir branche** dans le menu contextuel (positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler le menu contextuel)
 - ou cliquez sur l'icône correspondante de la barre d'outils
 - ou appuyez sur la touche F8.
3. Sélectionnez maintenant un élément CONT à insérer dans la branche.

6.3.6 Edition d'instructions LOG dans la section des instructions

6.3.6.1 Paramètres pour le langage de programmation LOG

Définition de la mise en page pour LOG

Vous pouvez définir la mise en page pour la programmation en langage LOG. Le format choisi (format vertical A4, format horizontal A4, taille maximale) a une influence sur le nombre d'éléments LOG représentables dans une branche.

1. Choisissez la commande **Outils > Paramètres**.
2. Dans la boîte de dialogue qui s'affiche, choisissez le format désiré dans la zone "Mise en page". Indiquez la taille de format désirée.

Remarques pour l'impression

Si vous désirez imprimer la section des instructions LOG, nous vous conseillons de définir le format de page approprié avant même d'écrire la section des instructions.

Paramètres de base sous Outils > Paramètres

La commande **Outils > Paramètres** vous permet de définir des paramètres généraux tels que le format de la mise en page ou la largeur du champ de l'opérande.

6.3.6.2 Règles de saisie des instructions LOG

La description du langage "LOG" figure dans le manuel *LOG pour S7-300/400 – Programmation de blocs* ou dans l'aide en ligne sur LOG.

Un réseau LOG peut être composé de plusieurs éléments. Tous les éléments doivent être reliés entre eux (CEI 1131-3).

Vous devez observer quelques règles lors de la programmation en LOG. Des messages vous signaleront d'éventuelles erreurs.

Insertion et édition d'adresses et de paramètres

Lors de l'insertion d'un élément LOG, les chaînes de caractères "???" et "..." sont utilisées pour réserver l'emplacement des adresses ou des paramètres.

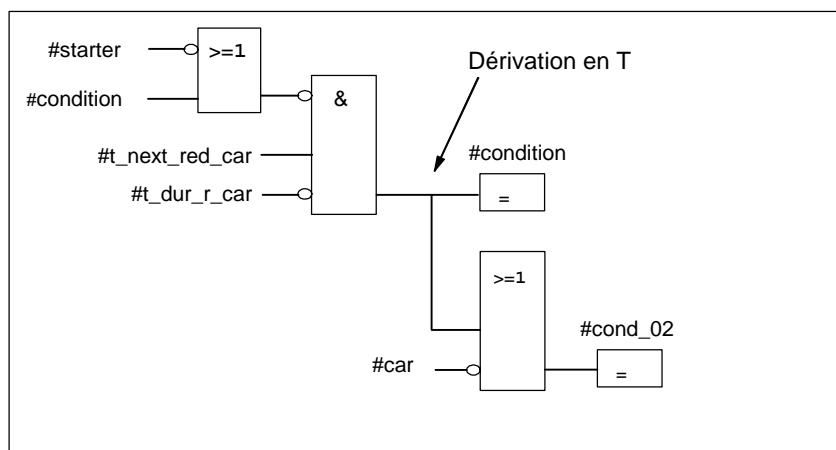
- La chaîne de caractères "???" en rouge signale les adresses et paramètres devant être définis.
- La chaîne de caractères "..." en noir signale les adresses et paramètres pouvant être définis.

Le type de données escompté s'affiche lorsque vous placez le pointeur sur les adresses ou paramètres non définis.

Placement de boîtes

Aux boîtes avec des fonctions logiques binaires (&, >=1, XOR), vous pouvez accoler des boîtes standard (bascules, compteurs, temporisations, opérations de calcul, etc.). Les boîtes de comparaison sont exclues de cette règle.

Dans un réseau, vous n'êtes pas autorisé à programmer des fonctions logiques séparées par des sorties distinctes. A l'aide de la branche, vous pouvez cependant attribuer plusieurs affectations à une séquence de boîtes logiques. La figure suivante représente un réseau contenant deux affectations.



Les boîtes suivantes ne doivent être placées qu'à l'extrémité droite de la séquence logique, où elles forment la terminaison de la séquence :

- Réinitialiser le compteur
- Comptage, décomptage
- Activer la temporisation "Impulsion" / "Impulsion prolongée"
- Activer la temporisation "Retard à la montée" / "retard à la retombée".

Il existe des boîtes pour lesquelles une fonction logique booléenne est requise et d'autres qui l'interdisent.

Boîtes nécessitant une fonction logique :

- sortie, mise à 1 sortie, remise à 0 sortie `_[R]`
- connecteur `_[#]_`, front positif `_[P]_`, front négatif `_[N]_`
- toutes les boîtes pour compteurs et temporisations
- sauts si 0 `_[JMPN]`
- activation relais de masquage `_[MCR<]`
- chargement du résultat logique dans le registre RB `_[SAVE]`
- retour saut `_[RET]`

Boîtes interdisant une fonction logique :

- début de relais de masquage `[MCRA]`
- fin de relais de masquage `[MCRD]`
- ouverture bloc de données `[OPN]`
- désactivation relais de masquage `[MCR>]`

Pour toutes les autres boîtes, les fonctions logiques sont autorisées sans être requises.

Entrée de validation/ sortie de validation

La connexion de l'entrée de validation "EN" ou de la sortie de validation "ENO" de boîtes est possible mais pas impérativement requise.

Suppression et modification

La suppression d'une boîte entraîne également la suppression de toutes les branches reliées avec les entrées booléennes de cette boîte, à l'exception de la branche principale.

Pour le simple échange d'éléments du même type, vous disposez du mode de substitution.

6.3.6.3 Comment saisir des éléments LOG

Procédez de la manière suivante :

1. Sélectionnez, dans un réseau, l'endroit derrière lequel un élément LOG doit être inséré.
2. Pour insérer l'élément souhaité :
 - cliquez sur l'icône correspondante dans la barre d'outils (pour boîte OU, boîte ET),
 - ou appuyez sur la touche de fonction correspondante F2 ou F3 (pour boîte OU, boîte ET),
 - ou bien faites un double-clic sur l'élément sélectionné dans la bibliothèque d'opérations ou amenez-le par glisser-lâcher dans l'éditeur de bloc.

L'élément LOG sélectionné est alors inséré, avec "???" ou "..." pour représenter les adresses et les paramètres.

Nota

Vous pouvez également éditer la section des instructions en sélectionnant des éléments LOG y figurant déjà et en choisissant les commandes **Edition > Couper**, **Edition > Copier** ou **Edition > Coller**.

Saisie d'adresses ou de paramètres

3. Positionnez le pointeur sur l'adresse ou le paramètre non défini avec un clic de la souris ou avec la touche TAB.
4. Saisissez l'adresse ou le paramètre sur les points ou les points d'interrogation (adressage direct ou indirect). Si l'affichage de la sélection des mnémoniques est activé (commande **Affichage > Avec sélection des mnémoniques**), la liste des mnémoniques définis s'affiche. Le mnémonique commençant par les caractères saisis est sélectionné et peut être repris via la touche ENTREE.
5. Appuyez sur la touche ENTREE.
 - Si la syntaxe est correcte, l'adresse s'affiche formatée en noir et l'éditeur de bloc ouvre automatiquement la zone de texte suivante qui ne contient pas encore d'adresse ou de paramètre.
 - Si la syntaxe est erronée, l'éditeur ne quitte pas le champ de saisie et le message d'erreur correspondant s'affiche dans la barre d'état. Si vous appuyez encore une fois sur la touche ENTREE, le champ de saisie se ferme et le texte erroné apparaît en rouge et en italique.

Nota

La chaîne de caractères ">>" à la sortie d'un bloc que vous cherchez à enregistrer ou à charger signifie que la sortie n'a pas été connectée.

Remplacement d'éléments LOG

Le mode de substitution vous permet de permuter facilement des éléments LOG de même type. L'avantage est que vous n'avez pas besoin de saisir à nouveau les adresses et paramètres. Il n'est possible de substituer à un élément LOG qu'un élément LOG de même type. Vous pouvez, par exemple, remplacer une boîte ET par une boîte OU, une bascule "mise à 0, mise à 1" par une bascule "mise à 1, mise à 0", une temporisation ou un compteur par une autre temporisation ou compteur.

Procédez de la manière suivante :

1. Appuyez sur la touche INSERT pour activer le mode de substitution. Le mode en vigueur est indiqué dans la barre d'état en bas à droite.
2. Sélectionnez l'élément LOG à remplacer.
3. Pour insérer l'élément souhaité :
 - cliquez sur l'icône correspondante de la barre d'outils
 - ou appuyez sur la touche de fonction correspondante F2 ou F3 (pour boîte OU, boîte ET)
 - ou bien faites un double-clic sur l'élément sélectionné dans la bibliothèque d'opérations ou amenez-le par glisser-lâcher dans l'éditeur de bloc.

Réactivez éventuellement le mode d'insertion en appuyant sur la touche INSERT. Le mode en vigueur est indiqué dans la barre d'état en bas à droite.

Sélection dans les réseaux LOG

Vous pouvez sélectionner les domaines suivants au sein d'un réseau par clic simple de la souris :

- les éléments LOG, par ex. une boîte ET ou les boîtes standard, par ex. un compteur,
- les lignes de connexion,
- les opérandes,
- les contacts d'entrée ou de sortie.

Vous pouvez choisir vous-même la couleur de la sélection dans la boîte de dialogue "CONT/LOG/LIST". Cette boîte de dialogue s'obtient par la commande **Outils > Paramètres...**

Pour sélectionner un réseau dans lequel vous pouvez saisir des éléments LOG, procédez de la manière suivante :

1. Cliquez sur la désignation du réseau (par ex. "Réseau 1").
2. Vous pouvez couper, coller ou copier le réseau ainsi sélectionné.

Insertion de réseaux LOG supplémentaires

Pour créer un nouveau réseau :

- Choisissez la commande **Insertion > Réseau**.
- Cliquez sur l'icône correspondante dans la barre d'outils.
- Faites un double-clic sur "Nouveau réseau" dans la bibliothèque d'opérations ou bien amenez cet élément par glisser-lâcher dans l'éditeur de bloc.
- Choisissez la commande Réseau dans le menu contextuel (positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler le menu contextuel).

Le nouveau réseau est alors inséré sous le réseau sélectionné.

Si vous saisissez plus d'éléments que l'écran ne peut en afficher, le réseau se décalera vers la gauche de l'écran. Les commandes **Affichage > Réduire/Agrandir/Facteur d'agrandissement...** vous permettent d'adapter la représentation afin d'avoir une meilleure vue d'ensemble.

Pour sélectionner un réseau, cliquez sur sa désignation (ex. : "Réseau 1"). Vous pouvez alors le couper, l'insérer de nouveau ou le copier.

Création de branches dans les réseaux LOG

Vous pouvez programmer plusieurs branches à l'intérieur d'un réseau LOG. Une branche ouvre une branche parallèle, en commençant avant l'entrée binaire sélectionnée. Vous pouvez insérer d'autres fonctions logiques dans la nouvelle branche.

1. Sélectionnez avec le pointeur l'entrée binaire avant laquelle vous voulez insérer une nouvelle branche.
2. Pour ouvrir la nouvelle branche :
 - choisissez la commande **Branchement** dans le menu contextuel (positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler le menu contextuel)
 - ou appuyez sur la touche F11
 - ou cliquez sur l'icône correspondante de la barre d'outils.

Insérer une connexion dans un réseau LOG

Vous pouvez relier deux séquences d'éléments LOG à l'intérieur d'un réseau LOG, mais à condition que le réseau ne comporte qu'une seule affectation.

Procédez de la manière suivante :

1. Sélectionnez avec le pointeur l'entrée binaire et la sortie binaire que vous désirez relier.
2. Pour connecter les objets binaires :
 - choisissez la commande **Connexion** dans le menu contextuel (positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler le menu contextuel)
 - ou appuyez sur la touche F12
 - ou cliquez sur l'icône correspondante de la barre d'outils.

Effacer et insérer une connexion dans un réseau LOG

1. Sélectionnez une entrée binaire.
2. Séparez la connexion en appuyant sur la touche SUPPR.
3. Insérez de nouveaux éléments LOG si vous le désirez.
4. Sélectionnez la sortie binaire.
5. Tout en maintenant le bouton de la souris enfoncé, insérez une connexion à l'entrée binaire désirée.

Le cas échéant, la disposition graphique des éléments est modifiée.

6.3.7 Edition d'instructions LIST dans la section des instructions

6.3.7.1 Paramètres pour le langage de programmation LIST

Choix d'un jeu d'abréviations

Vous pouvez choisir parmi deux jeux d'abréviations :

- allemandes ou
- anglaises

Avant d'ouvrir un bloc, vous sélectionnez un jeu d'abréviations dans la boîte de dialogue ouverte par **Outils > Paramètres....** Il n'est pas possible de changer de jeu d'abréviations durant l'édition du bloc.

C'est dans l'onglet "Propriétés" que vous pouvez modifier les propriétés du bloc.

Vous pouvez ouvrir plusieurs blocs dans l'éditeur de bloc et les éditer à tour de rôle.

6.3.7.2 Règles de saisie des instructions LIST

La description du langage "LIST" figure dans le manuel *LIST pour S7-300/400 – Programmation de blocs* ou dans l'aide en ligne sur LIST (Description des langages).

Vous devez observer les règles suivantes lors de la saisie d'instructions LIST en mode de saisie incrémentale.

- Respectez l'ordre de programmation des blocs : il faut programmer les blocs appelés avant les blocs appelants.
- Une instruction se compose d'un repère de saut (facultatif), d'une opération, d'un opérande et d'un commentaire (facultatif).
Exemple M001: U E1.0 //commentaire
- Une ligne contient une seule instruction.
- Un bloc peut contenir 999 réseaux au maximum.
- Un réseau peut contenir environ 2000 lignes. Le nombre de lignes possibles varie selon que l'affichage est agrandi ou réduit.
- Vous pouvez saisir les opérations et les adresses absolues indifféremment en majuscules ou en minuscules.

6.3.7.3 Saisie d'instructions LIST

Comment saisir des instructions LIST

Lorsque vous créez un nouveau bloc de code, vous pouvez immédiatement en éditer le premier réseau. Vous parvenez dans un réseau par clic de la souris sur l'une des lignes qu'il contient. Dans chaque réseau, vous saisissez simplement les instructions ligne par ligne via le clavier. Pour l'édition, vous disposez de toutes les fonctions d'édition habituelles.

1. Ouvrez la zone de texte pour le réseau en cliquant sur la surface libre en dessous de la zone de commentaire verte (ou bien en dessous du titre de réseau si l'affichage des commentaires est désactivé).
2. Ecrivez l'opération puis, en laissant un espace libre, l'adresse (adressage direct ou indirect).
3. Appuyez sur la barre d'espace et saisissez le commentaire d'instruction précédé de deux barres obliques (ce commentaire est facultatif).
4. Clôturez la saisie de l'instruction - avec ou sans commentaire - en appuyant sur ENTREE.

Une fois la ligne saisie, le logiciel vérifie la syntaxe et affiche l'instruction formatée. Les minuscules éventuelles dans l'opération ou l'adresse absolue sont converties en majuscules.

Les erreurs de syntaxe s'affichent en rouge. Vous devez les corriger avant d'appliquer ou d'enregistrer le bloc de code.

Sélection de zones de texte dans les instructions LIST

Vous pouvez sélectionner du texte caractère par caractère dans un réseau LIST.

1. Positionnez le curseur sur le premier caractère.
2. Pour sélectionner le texte, maintenez le bouton gauche de la souris enfoncé et faites glisser le curseur sur le texte à sélectionner.

Vous pouvez sélectionner plusieurs lignes d'instructions simultanément en faisant glisser la souris verticalement, tout en maintenant son bouton gauche enfoncé. Une autre manière de sélectionner les zones de texte consisterait à utiliser les touches de direction DROITE, GAUCHE, HAUT ou BAS tout en maintenant la touche MAJ enfoncée.

Nota

Vous pouvez choisir vous-même la couleur des sélections. Pour cela, ouvrez la boîte de dialogue avec **Outils > Paramètres...** et choisissez la couleur de "l'élément sélectionné".

Insertion de réseaux LIST supplémentaires

Pour créer un nouveau réseau :

- Choisissez la commande **Insertion > Réseau**.
- Cliquez sur l'icône correspondante de la barre d'outils.
- Faites un double-clic sur "Nouveau réseau" dans la bibliothèque d'opérations ou amenez cet élément par glisser-lâcher dans l'éditeur de bloc.
- Choisissez la commande Réseau dans le menu contextuel.
Positionnez le pointeur et cliquez avec le bouton droit de la souris pour appeler ce menu contextuel.

Le nouveau réseau est inséré sous le réseau sélectionné.

Si vous saisissez plus d'éléments que l'écran ne peut en afficher, le réseau se décalera vers la gauche de l'écran. Les commandes **Affichage > Réduire/Agrandir/Facteur d'agrandissement...** vous permettent d'adapter la représentation afin d'avoir une meilleure vue d'ensemble.

Pour sélectionner un réseau, cliquez sur sa désignation (ex. : "Réseau 1"). Vous pouvez alors le couper, l'insérer de nouveau ou le copier.

Saisie de commentaires dans les instructions LIST

Vous pouvez préciser un commentaire pour chaque instruction dans le langage de programmation LIST.

1. Appuyez sur la barre d'espacement après avoir saisi l'adresse ou le mnémonique.
2. Faites précéder votre commentaire d'instruction de deux barres obliques (//).
3. Clôturez la saisie du commentaire en appuyant sur ENTREE.

6.3.8 Actualisation d'appels de bloc

En choisissant la commande **Edition > Appel > Actualiser** dans l'éditeur de bloc, vous pouvez mettre à jour automatiquement des appels de bloc ou des utilisations d'UTD devenus invalides après les modifications d'interfaces suivantes :

- insertion de nouveaux paramètres formels,
- suppression de paramètres formels,
- changement du nom de paramètres formels,
- modification du type de paramètres formels,
- changement de l'ordre (recopie) de paramètres formels.

L'affectation entre paramètres formels et paramètres effectifs est réalisée d'après les règles suivantes, dans l'ordre spécifié.

1. Noms de paramètre identiques

Les paramètres effectifs sont affectés automatiquement lorsque le nom du paramètre formel est resté identique.

Cas particulier : dans CONT et LOG, la fonction logique précédente de paramètres d'entrée binaires ne peut être affectée automatiquement que si le type de données est identique (BOOL). Si dans un tel cas, le type de données a été modifié, la fonction logique précédente reste conservée sous forme de branche ouverte.

2. Types de données de paramètre identiques

Une fois que les paramètres de nom identique ont été affectés, les paramètres effectifs non encore affectés le seront à des paramètres formels de même type de données que "l'ancien" paramètre formel.

3. Position de paramètre identique

Les paramètres effectifs non encore affectés d'après les règles 1 et 2, seront à présent affectés aux nouveaux paramètres formels, d'après leur position dans "l'ancienne" interface.

4. Si des paramètres effectifs ne peuvent pas être affectés d'après les trois règles précitées, ils seront supprimés ou, dans le cas de fonctions logiques précédentes dans CONT ou LOG, resteront conservés sous forme de branches ouvertes.

Après avoir exécuté cette fonction, vérifiez les modifications réalisées dans la table de déclaration des variables ainsi que dans la section des instructions du programme.

6.4 Création des blocs de données

6.4.1 Principes de la création des blocs de données

Dans un bloc de données, vous stockez des données auxquelles votre machine ou installation accède. Contrairement au bloc de code, que vous avez programmé dans l'un des langages de programmation CONT/LOG ou LIST, le bloc de données ne contient que la table de déclaration des variables. Il ne possède pas de section d'instructions et ne demande donc pas la programmation de réseaux.

Générer un bloc de données

1. Choisissez la commande **Fichier > Nouveau > Bloc**.
2. Dans la boîte de dialogue qui s'ouvre, sélectionnez le type de bloc "Bloc de données" et entrez le numéro.

Après avoir ouvert un bloc de données, vous pouvez l'afficher dans la vue des déclarations ou dans la vue des données. Vous passez d'une vue à l'autre en choisissant les commandes **Affichage > Vue des déclarations** et **Affichage > Vue des données**.

Vue des déclarations

Choisissez la vue des déclarations pour

- lire ou déterminer la structure de données des DB globaux,
- lire la structure de données des DB associés à un type de données utilisateur (UDT),
- lire la structure de données des DB associés à un bloc fonctionnel.

Vous ne pouvez pas modifier la structure des blocs de données associés à un FB ou à un type de données utilisateur. Il vous faudrait d'abord modifier le FB ou l'UDT correspondant, puis créer un nouveau DB.

Vue des données

Choisissez la vue des données pour modifier des données. Seule la vue des données vous permet d'afficher, de saisir ou de modifier la valeur en cours de chaque élément. Dans la vue des données des blocs de données, les éléments des variables avec types de données complexes sont énumérés chacun avec leur nom complet.

Différence entre un bloc de données d'instance et un bloc de données global

Le bloc de données global n'est pas affecté à un bloc de code. Il sert à stocker des données pour les installations ou machines commandées par la CPU et peut être appelé et édité à un endroit quelconque du programme.

Le bloc de données d'instance, quant à lui, est directement affecté à un bloc de code, par exemple à un bloc fonctionnel. Il contient les données ayant été stockées dans la table de déclaration des variables d'un bloc fonctionnel.

6.4.2 Vue des déclarations d'un bloc de données

Il n'est pas possible de modifier la vue des déclarations pour les blocs de données non globaux.

Colonne	Explication
Adresse	Affiche l'adresse que STEP 7 Lite assigne automatiquement à la variable quand vous clôturez la saisie d'une déclaration.
Déclaration	Cette colonne s'affiche seulement pour les blocs de données d'instance. Elle indique comment les variables ont été déclarées dans la déclaration de variables du FB : <ul style="list-style-type: none"> paramètre d'entrée ("in"), paramètre de sortie ("out"), paramètre d'entrée-sortie ("in_out"), données statiques ("stat").
Nom	Tapez ici le nom que vous devez attribuer à chaque variable.
Type	Tapez ici le type de données de la variable (BOOL, INT, WORD, ARRAY, etc.). Il peut s'agir d'un type de données simple, complexe ou utilisateur.
Valeur initiale	Tapez ici une valeur initiale si vous ne voulez pas que le logiciel utilise la valeur par défaut du type de données saisi. Elle doit être compatible avec le type de données. Si vous ne précisez pas explicitement une valeur actuelle pour la variable, c'est cette valeur initiale qui sera utilisée comme valeur actuelle lors du premier enregistrement du bloc de données.
Commentaire	Ce champ vous permet de saisir, pour expliciter la variable, un commentaire qui ne doit pas dépasser 80 caractères.

6.4.3 Vue des données d'un bloc de données

La vue des données affiche les valeurs actuelles de toutes les variables du bloc de données. C'est le seul endroit où vous pouvez modifier ces valeurs. Le tableau présenté est le même pour tous les blocs de données globaux. Il comporte en plus une colonne "Déclaration" pour les blocs de données d'instance.

Pour les variables de type de données complexe ou de type de données utilisateur, la vue des données indique tous les éléments un par un, chacun dans une ligne avec son nom complet. Lorsque les éléments se trouvent dans la zone in_out d'un bloc de données d'instance, un pointeur est positionné sur le type de données complexe ou utilisateur dans la colonne "Valeur en cours".

La vue des données comporte les colonnes suivantes :

Colonne	Explication
Adresse	Affiche l'adresse que STEP 7 Lite assigne automatiquement à la variable.
Déclaration	Cette colonne s'affiche seulement pour les blocs de données d'instance. Elle indique comment les variables ont été déclarées dans la déclaration de variables du FB : <ul style="list-style-type: none"> paramètre d'entrée ("in"), paramètre de sortie ("out"), paramètre d'entrée-sortie ("in_out"), données statiques ("stat").
Nom	C'est le nom attribué à la variable. Vous ne pouvez pas éditer ce champ dans la vue des données.
Type	C'est le type de données attribué à la variable. Pour un bloc de données global, cette colonne n'affiche plus que des types de données simples, puisque la vue des données énumère un par un les éléments des variables de type de données complexe ou utilisateur. Pour un bloc de données d'instance, cette colonne affiche en plus les types de données paramètre ; pour les paramètres d'entrée-sortie ("in_out") de type complexe ou utilisateur, un pointeur est positionné sur le type de données dans la colonne "Valeur en cours".
Valeur initiale	Il s'agit de la valeur initiale que vous avez attribuée à la variable si vous ne voulez pas que le logiciel utilise la valeur par défaut du type de données saisi. Si vous ne précisez pas explicitement une valeur actuelle pour la variable, c'est cette valeur initiale qui sera utilisée comme valeur actuelle lors du premier enregistrement du bloc de données.
Valeur en cours	Hors ligne : c'est la valeur qu'avait la variable à l'ouverture du bloc de données ou après la dernière modification que vous avez enregistrée (cette indication n'est pas mise à jour, même si vous avez ouvert le DB en ligne !). En ligne : c'est la valeur à l'ouverture du bloc de données qui s'affiche, mais elle n'est pas mise à jour automatiquement. Appuyez sur la touche F5 pour la mettre à jour. Vous pouvez éditer ce champ s'il n'appartient pas à un paramètre d'entrée-sortie ("in_out") de type de données complexe ou utilisateur. Toutes les valeurs saisies doivent être compatibles avec les types de données.
Commentaire	C'est le commentaire affecté à la variable pour l'expliquer. Vous ne pouvez pas éditer ce champ dans la vue des données.

6.4.4 Saisie et enregistrement des blocs de données

6.4.4.1 Saisie de la structure de données de blocs de données globaux

Si vous avez ouvert un bloc de données qui n'est associé ni à un UDT ni à un FB, vous pouvez définir sa structure dans la vue des déclarations.

1. Ouvrez un bloc de données global, c'est-à-dire un bloc qui n'est associé ni à un UDT, ni à un FB. Un bloc de données global se reconnaît au langage de création "DB" (voir la vue "Propriétés").
2. En choisissant la commande de menu **Affichage > Vue des déclarations**, activez la vue des déclarations du bloc de données, si ce n'est déjà le cas.
3. Définissez sa structure en complétant la table affichée avec les données suivantes.

Vous ne pouvez pas modifier la vue des déclarations de blocs de données non globaux.

Colonne	Signification
Adresse	Ici s'affiche l'adresse que STEP 7 Lite affecte automatiquement à la variable lorsque vous achevez la saisie d'une déclaration.
Nom	Indiquez ici le nom de la variable.
Type	Indiquez ici le type de données pour la variable (BOOL, INT, WORD, ARRAY, etc.) ou choisissez-le dans le menu contextuel (clic sur le bouton droit de la souris. Les variables peuvent être du type de données simple, complexe ou utilisateur.
Valeur initiale	Vous pouvez préciser ici une valeur initiale si le logiciel ne doit pas prendre la valeur par défaut pour le type de données indiqué. Toutes les valeurs saisies doivent être compatibles avec les types de données. Cette valeur initiale est prise comme valeur en cours de la variable lors de la première application ou du premier enregistrement du bloc de données si vous ne lui affectez pas explicitement une valeur en cours.
Commentaire	Vous pouvez saisir dans cette zone un commentaire pour la documentation de la variable. La longueur du commentaire ne doit pas dépasser 80 caractères.

6.4.4.2 Saisie / affichage de la structure de données de blocs de données associés à un FB (DB d'instance)

Saisie

Lorsque vous associez un bloc de données à un bloc fonctionnel (DB d'instance), la déclaration des variables du FB détermine la structure du bloc de données. Les modifications ne sont possibles que dans le bloc fonctionnel associé.

1. Ouvrez le bloc fonctionnel associé.
2. Editez la table de déclaration des variables du bloc fonctionnel.
3. Recréez ensuite le bloc de données d'instance.

Affichage

Vous pouvez afficher dans la vue des déclarations du DB d'instance la manière dont les variables ont été déclarées dans le FB.

1. Ouvrez le bloc de données.
2. Affichez la vue des déclarations du bloc de données, si ce pas déjà le cas.
3. Les explications relatives à la table affichée figurent ci-après.

Vous ne pouvez pas modifier la vue des déclarations de blocs de données non globaux.

Colonne	Signification
Adresse	Cette colonne affiche l'adresse que STEP 7 Lite affecte automatiquement à la variable.
Déclaration	<p>Cette colonne vous indique comment les variables ont été déclarées dans le bloc fonctionnel :</p> <ul style="list-style-type: none"> • paramètres d'entrée ("in") • paramètres de sortie ("out") • paramètres d'entrée/sortie ("in_out") • données statiques ("stat") <p>Les données temporaires déclarées du FB ne sont pas dans le bloc de données d'instance.</p>
Nom	Il s'agit ici du nom affecté à la variable dans la déclaration des variables du FB.
Type	<p>Il s'agit ici du type de données affecté à la variable dans la déclaration des variables du FB. Les variables peuvent avoir un type de données simple, un type de données complexe ou un type de données utilisateur.</p> <p>Si d'autres blocs fonctionnels pour l'appel desquels des variables statiques ont été déclarées sont appelés à l'intérieur du FB, il est possible d'indiquer ici un FB ou un bloc fonctionnel système (SFB) comme type de données.</p>
Valeur initiale	<p>Il s'agit ici de la valeur initiale que vous avez définie pour la variable dans la déclaration des variables du FB si le logiciel ne doit pas utiliser la valeur par défaut du type de données.</p> <p>La valeur initiale est prise comme valeur en cours de la variable lors du premier enregistrement du bloc de données si vous ne lui affectez pas explicitement une valeur en cours.</p>
Commentaire	Il s'agit ici du commentaire défini dans la déclaration des variables du FB pour la documentation de l'élément de données. Vous ne pouvez pas éditer cette zone.

Nota

Dans les blocs de données associés à un FB, vous ne pouvez éditer que les valeurs en cours des variables. La saisie de ces valeurs en cours se fait dans la vue des données des blocs de données.

6.4.4.3 Saisie de la structure de types de données utilisateur (UDT)

1. Ouvrez le type de données utilisateur (UDT).
2. Affichez la vue des déclarations, si ce n'est déjà le cas.
3. Définissez la structure du type de données utilisateur en déterminant l'ordre des variables, leur type de données et, éventuellement, leur valeur initiale en fonction des données ci-après.
4. Vous achevez la saisie d'une variable en quittant la ligne avec la touche TAB ou ENTREE.

Colonne	Signification
Adresse	Ici s'affiche l'adresse que STEP 7 Lite affecte automatiquement à la variable lorsque vous achevez la saisie d'une déclaration.
Nom	Indiquez ici le nom que vous devez affecter à chaque variable.
Type	Indiquez ici le type de données pour la variable (BOOL, INT, WORD, ARRAY, etc.). Les variables peuvent avoir un type de données simple, un type de données complexe ou un type de données utilisateur.
Valeur initiale	Vous pouvez préciser ici une valeur initiale si le logiciel ne doit pas prendre la valeur par défaut pour le type de données indiqué. Toutes les valeurs saisies doivent être compatibles avec les types de données. Lorsque vous appliquez ou enregistrez pour la première fois une instance de l'UDT (variable ou bloc de données), la valeur initiale est prise comme valeur en cours pour la variable si vous n'indiquez pas explicitement une autre valeur en cours.
Commentaire	Vous pouvez saisir dans cette zone un commentaire pour la documentation de la variable. La longueur du commentaire ne doit pas dépasser 80 caractères.

6.4.4.4 Saisie / affichage de la structure de blocs de données associés à un UDT

Saisie

Lorsque vous associez un bloc de données à un type de données utilisateur (UDT), la structure des données de l'UDT détermine celle du bloc de données. Les modifications ne sont possibles que dans le type de données utilisateur associé.

1. Ouvrez le type de données utilisateur.
2. Editez la structure du type de données utilisateur.
3. Recréez le bloc de données.

Affichage

Dans la vue des déclarations du DB, vous pouvez afficher uniquement la manière dont les variables ont été déclarées dans l'UDT.

1. Ouvrez le bloc de données.
2. Affichez la vue des déclarations du bloc de données, si ce n'est déjà le cas.
3. Les explications relatives à la table affichée figurent ci-après.

Vous ne pouvez pas modifier la vue des déclarations. Les modifications ne sont possibles que dans le type de données utilisateur associé.

Colonne	Signification
Adresse	Cette colonne affiche l'adresse que STEP 7 Lite affecte automatiquement à la variable.
Nom	Il s'agit ici du nom affecté à la variable dans l'UDT.
Type	Il s'agit ici du type de données affecté à la variable dans l'UDT. Les variables peuvent avoir un type de données simple, un type de données complexe ou un type de données utilisateur.
Valeur initiale	Il s'agit ici de la valeur initiale que vous avez définie pour la variable dans l'UDT si le logiciel ne doit pas utiliser la valeur par défaut du type de données. Cette valeur initiale est prise comme valeur en cours de la variable lors du premier enregistrement du bloc de données si vous n'affectez pas explicitement une valeur en cours à cette variable.
Commentaire	Il s'agit ici du commentaire défini dans l'UDT pour la documentation de l'élément de données.

Nota

Dans les blocs de données associés à un UDT, vous ne pouvez éditer que les valeurs en cours des variables. La saisie de ces valeurs en cours se fait dans la vue des données des blocs de données.

6.4.4.5 Modification de valeurs dans la vue des données

L'édition de valeurs en cours n'est possible que dans la vue des données de blocs de données.

1. Activez, si nécessaire, la vue des données de la table à l'aide de la commande **Affichage > Vue des données**.
2. Saisissez les valeurs en cours désirées pour les éléments de données dans la colonne "Valeur en cours". Ces valeurs doivent être compatibles avec le type de données des éléments ;

Les saisies erronées - par exemple, une valeur saisie incompatible avec le type de données - sont reconnues immédiatement lors de l'édition et affichées en rouge. Vous devez les corriger avant d'enregistrer.

Attention

Les modifications des valeurs ne sont mémorisées que lors de l'enregistrement des blocs de données.

6.4.4.6 Réinitialisation de valeurs en leur substituant leur valeur initiale

La réinitialisation est uniquement possible dans la vue des données de blocs de données.

1. Activez, si nécessaire, la vue des données de la table à l'aide de la commande **Affichage > Vue des données**.
2. Choisissez pour ce faire la commande Edition > Réinitialiser bloc de données.

Toutes les variables sont réinitialisées, c'est-à-dire aux valeurs en cours de toutes les variables est substituée la valeur initiale correspondante.

Attention

Les modifications des valeurs ne sont mémorisées que lors de l'enregistrement des blocs de données.

6.5 Affichage des références

6.5.1 Références possibles

Servez-vous des références croisées "Liste des références croisées", "Opérandes utilisés" et "Structure du programme pour avoir une vue d'ensemble de l'utilisation des opérandes, des zones de mémoire, des blocs etc.. Pour afficher les références croisées, effectuez un double clic sur l'icône "Références croisées" dans la fenêtre du projet.

- Lorsque vous créez ou modifiez un programme, servez-vous des références pour garder une vue d'ensemble des opérandes utilisés et des appels de blocs.
- Lorsque vous testez un programme ou recherchez des erreurs, servez-vous des références croisées pour découvrir quel opérande est édité dans quel bloc et avec quelle instruction ou quel bloc est appelé par un autre bloc.
- Servez-vous des références croisées comme complément à votre documentation du projet afin de fournir aux utilisateurs finaux, une vue d'ensemble de tous les opérandes utilisés, des zones de mémoire et des blocs.

Le tableau suivant précise les renseignements donnés par les différents onglets :

Vue	Utilité
Liste des références croisées	Vue d'ensemble de l'emploi d'opérandes des zones de mémoire E, A, M, P, T, Z et d'appels de DB, FB, FC, SFB et SFC au sein du programme utilisateur. Utilisez la fonction de filtrage (filtres prédéfinis ou définis par l'utilisateur) afin de limiter la sélection des opérandes et zones de mémoire affichés.
Opérandes utilisés	Cette vue d'ensemble montre quels bits, octets, mots ou double mots des opérandes et zones de mémoire E, A et M sont déjà utilisés au sein du programme utilisateur ; elle constitue une base importante pour la programmation et l'extension du programme utilisateur. L'onglet "Opérandes utilisés" fournit en outre des informations sur les temporisations et compteurs utilisés.
Structure du programme	Affiche la hiérarchie d'appel des blocs au sein du programme utilisateur et offre une vue d'ensemble des blocs utilisés et de leurs relations d'interdépendance.

6.5.2 Vue synoptique des adresses

Pour afficher les adresses d'entrées et de sorties de tous les modules configurés, choisissez la commande de menu **Affichage > Vue synoptique des adresses**.

STEP 7 Lite affiche alors la vue synoptique des adresses sous forme de tableau. Les colonnes du tableau indiquent non seulement l'adresse et le type d'adresse (E,A), mais également des informations sur le lieu (châssis, emplacement...) et sur l'identification du module (nom, numéro de référence) auquel l'adresse est affectée.

La vue synoptique des adresses reste à l'avant-plan, même lorsque vous passez à une autre application de STEP 7 Lite.

Afficher/masquer des colonnes dans la vue synoptique des adresses

Un menu contextuel vous permet d'afficher/masquer les colonnes.

Exemple : cliquez dans la vue synoptique des adresses avec le bouton droit de la souris et choisissez la commande de menu **Afficher colonne > Référence**.

Filtrer la vue synoptique des adresses

Vous pouvez filtrer la vue synoptique des adresses en activant/désactivant les cases à cocher "Entrées" et "Sorties". Lorsque vous désactivez p. ex. la case à cocher "Sorties", seules les adresses d'entrée s'affichent.

6.5.3 Liste des références croisées

La liste des références croisées montre l'utilisation d'opérandes dans le programme utilisateur.

Elle énumère les opérandes utilisés faisant partie des zones de mémoire entrées (E), sorties (A), mémentos (M), temporisations (T), compteurs (Z), blocs fonctionnels (FB), fonctions (FC), blocs fonctionnels système (SFB), fonctions système (SFC), périphérie (P) et blocs de données (DB), avec leur adresse (adresse absolue, nom) et leur utilisation. Elle s'affiche dans une fenêtre de travail.

Chaque ligne de la vue correspond à une entrée de la liste. La fonction de recherche aide à trouver précisément certains opérandes et mnémoniques.

Pour ouvrir la liste des références croisées, il suffit d'un double-clic sur l'icône "Références croisées" dans la fenêtre du projet. Vous pouvez alors choisir entre les onglets "Liste des références croisées", "Opérandes utilisés" et "Structure du programme".

Composition

Une entrée de la liste des références croisées comporte les colonnes suivantes :

Colonne	Contenu/Signification
Opérande	Adresse absolue de l'opérande
Mnémonique	Nom de l'opérande
Bloc	Bloc dans lequel l'opérande est utilisé
Mnémonique bloc	Identificateur symbolique du bloc
Réseau	Numéro du réseau dans lequel l'opérande est utilisé
Ligne	Position où l'opérande est utilisé dans le réseau
Accès	Indique s'il l'accès à l'opérande est en lecture (R) et/ou en écriture (W)
Langage	Langage dans lequel l'accès est programmé
Opération	Opération dans laquelle l'opérande est utilisé

Vous pouvez adapter la largeur des colonnes à vos besoins à l'aide de la souris.

Tri

Par défaut, la liste des références croisées est triée selon les zones de mémoire. Si vous cliquez sur le titre d'une colonne, elle sera triée selon les entrées de cette colonne.

Filtre

Vous pouvez filtrer la liste des références croisées. Vous pouvez utiliser des filtres prédéfinis ou créer vos propres filtres.

Sélectionnez votre filtre dans la liste déroulante "Filtre".

Pour créer de nouveaux filtres ou modifier des filtres existants, cliquez sur le bouton "Filtre". Dans le dialogue suivant, vous pouvez définir les propriétés du filtre et l'appliquer. Un filtre non appliqué est caractérisé dans la liste déroulante par un astérisque (*) et n'est pas enregistré lors de la sauvegarde du projet.

Exemple de liste des références croisées

Opérande	Mnémonique	Bloc	Mnémo. bloc	Réseau	Ligne	Accès	Langage	Opération
E 1.0	Moteur_marche	OB 2	Cycle	1		R	LIST	CALL
M1.2	BitMemento	FC 2	Moteur	2	3	RW	CONT	-()-
Z2	Compteur2	FB2	Multiinstance	5	1		LOG	

6.5.4 Opérandes utilisés

Trois listes énumèrent les opérandes déjà utilisés dans le programme utilisateur. Elles sont fondamentales pour rechercher les erreurs ou modifier le programme.

Liste "Bits et octets utilisés"

Cette liste indique les bits utilisés dans les octets des zones de mémoire Entrées (E), Sorties (A) et Mémentos (M).

Chaque ligne contient un octet de la zone de mémoire. Les bits utilisés sont marqués d'une croix. Les lignes apparaissant sur fond bleu signifie que l'accès est effectué au niveau octet, mot ou double mot.

Symboles employés dans la liste "Bits et octets utilisés" :

X	utilisation directe de l'opérande
Barre bleue	traitement indirect de l'opérande (au niveau octet, mot ou double mot), le fond des cellules est bleu

Colonnes de la liste "Bits et octets utilisés"

Colonne	Contenu/signification
7 6 5 4 3 2 1 0	Numéro de bit dans l'octet concerné
B	Cet octet est occupé par un accès au niveau octet.
W	Cet octet est occupé par un accès au niveau mot.
D	Cet octet est occupé par un accès au niveau double mot.

Exemple

Voici un exemple typique de liste d'utilisation des entrées, sorties et mémentos (E/A/M).

	7	6	5	4	3	2	1	0	B	W	D
EB0		X	X	X	X	X	X				
EB1		X	X	X		X	X	X			
AB4						X	X	X			
AB5		X	X	X		X	X	X			
MB2											
MB3											
MB4											
MB5											

La première ligne montre l'affectation de l'octet d'entrée EB 0. Les entrées de l'opérande EB 0 sont utilisées directement (accès par bit). Les colonnes "1", "2", "3", "4", "5" et "6" contiennent chacune un "X" pour les accès par bit. Pour les octets de memento 2, 3, 4 et 5, l'accès a lieu au niveau du mot, c'est pourquoi la colonne "W" contient une "barre" et les cellules ont en plus un fond bleu ; la pointe noire de la barre indique le début de l'accès par mot.

Listes "Temporisations utilisées" et "Compteurs utilisés"

Avec les listes "Temporisations utilisées" et "Compteurs utilisés", vous voyez d'un coup d'œil quels sont les temporisations (T) et les compteurs (Z) utilisés.

Symboles employés dans la liste "Temporisations/compteurs utilisés" :

Champ bleu la temporisation/le compteur est utilisé.

Champ blanc la temporisation/le compteur n'est pas utilisé.

Exemple de "Temporisations utilisées"

Tempo	0	1	2	3	4	5	6	7	8	9
T-		T1	T2							
T1-									T18	
T2-	T20									
T17-										T179
T22-					T224					

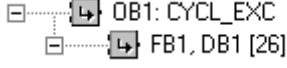
Dans cet exemple, les temporisations T1, T2, T18, T20, T179 et T224 sont utilisées.

6.5.5 Structure du programme

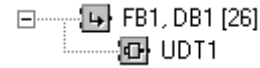
La structure du programme décrit les relations ou interdépendances des blocs utilisés dans un programme utilisateur.

Une relation ou interdépendance correspond à

- un appel (p. ex. le bloc A appelle le bloc B via l'instruction CALL)

	<p>Exemple de représentation d'un appel :</p> <p>l'OB 1 appelle le FB 1 avec indication du DB 1 comme DB d'instance</p>
---	---

- l'utilisation d'une déclaration d'interface (p. ex. le bloc A utilise l'UDT B ou le FB C dans sa déclaration d'interface)

	<p>Exemple de représentation de l'utilisation d'une déclaration d'interface :</p> <p>le FB 1 utilise l'UDT 1 (comme type de données) dans sa déclaration d'interface</p>
---	--

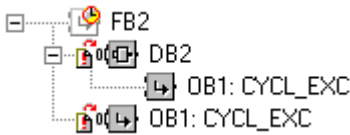
Besoin en données locales

La structure du programme vous donne également un aperçu du besoin en données locales. Lorsqu'un OB 1 existe, le besoin maximal en données locales de la CPU pour l'exécution de cet OB est précisé à côté entre crochets. Le besoin en données locales d'un chemin d'appel est déterminé par addition du besoin en données locales de chaque bloc individuel, à partir de l'OB 1. Le besoin en données locales maximal des OB d'erreur de synchronisation (OB 121 et OB 122) est également déterminé et indiqué entre crochets derrière le texte de l'OB 1.

Incohérences

Vous pouvez aussi visualiser des incohérences, p. ex. la manière dont elles apparaissent dans un bloc, du fait de modifications d'interfaces. Vous pouvez ouvrir les blocs dans la structure du programme (fonction "Aller à...") et corriger successivement les incohérences ou alors les corriger automatiquement en choisissant la commande de menu

Outils > Vérification de la cohérence du programme.

<p>Exemple de représentation d'incohérences :</p>	
---	---

Afficher la structure du programme

Pour ouvrir la structure du programme, effectuez un double clic sur le bouton "Références croisées" dans la fenêtre du projet. Parmi les onglets "Liste des références croisées", "Opérandes utilisés" et "Structure du programme" au bas de la fenêtre, choisissez l'onglet "Structure du programme".

Possibilités d'affichage et de paramétrage de la structure du programme


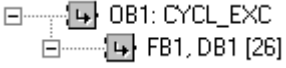



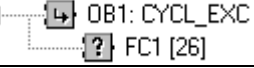

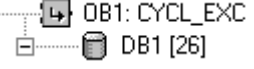

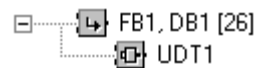
Le paramétrage des champs "A partir du bloc" et "Afficher" a une influence sur la représentation de la structure du programme. Les paramètres possibles sont décrits ci-après. Afin de simplifier la description, un "Bloc" désignera également un type de données (UDT).

Champ	Paramétrage possible	Signification
A partir du bloc	Blocs "système" et blocs existant dans le projet	Le bloc est sélectionné dans la structure du programme, à l'occurrence à laquelle il apparaît en premier dans la structure du programme affichée. "Système" correspond au niveau hiérarchique le plus élevé représenté par le système d'exploitation de la CPU. Le système appelle tous les OB.
Afficher	<ul style="list-style-type: none"> Structure d'appel Structure d'appel avec appels multiples Structure d'application Structure d'application (seulement conflits) 	<p>La Structure d'appel affiche les blocs appelés de même que la relation entre ces blocs, à partir de l'OB1. Tout à gauche, se trouvent les OB qui peuvent uniquement être appelés par le système d'exploitation de la CPU. En dessous et de manière décalée se trouvent les blocs appelés ou utilisés par l'OB respectif. Cette hiérarchie se poursuit ainsi en fonction de la profondeur d'imbrication des appels. Seul le premier appel ou la première utilisation sont affichés.</p> <p>La Structure d'appel avec appels multiples affiche tous les appels ou toutes les utilisations de blocs.</p> <p>La Structure d'application affiche pour chaque bloc du projet, ses interdépendances avec les autres blocs. Le bloc est représenté tout à gauche. En dessous et de manière décalée se trouvent les blocs qui appellent ou utilisent ce bloc.</p> <p>La Structure d'application (seulement conflits) affiche uniquement les relations présentant des conflits liés à l'horodatage dans l'interface ou à la table des mnémoniques :</p> <ul style="list-style-type: none"> L'horodatage dans l'interface du bloc appelé (ou utilisé) s'est modifié depuis le dernier enregistrement du bloc appelant (ceci peut également se produire lors du chargement de blocs dans la PG). "Classement symbolique" est sélectionné dans le projet et des mnémoniques ont été modifiés après l'enregistrement des blocs. <p>Lorsqu'il n'y a aucun conflit, seul l'icône "Système" est affichée.</p>

Nota



Vous pouvez également modifier les affichages (structure d'appel, etc.) en choisissant les commandes du menu Affichage (**menu Affichage > Structure d'appel**).

Représentation des interdépendances entre les blocs dans la structure du programme

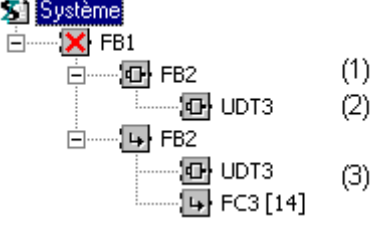
Icône	Signification	Exemple
	Bloc appelé de manière classique avec CALL	
	Bloc appelé de manière inconditionnelle avec UC	
	Bloc appelé de manière conditionnelle avec CC	
	Bloc de données ouvert et accès à son contenu (p. ex. L DB 1.DBW 10)	
	Bloc utilisant une déclaration d'interface p. ex. FB, SFB, UDT	

Représentation de blocs non utilisés

Les blocs qui existent dans le programme sans y être utilisés sont représentés avec une croix:

Icône	Signification
	Bloc de données non utilisé
	FB, FC, SFB, SFC, UDT non utilisés

L'exemple suivant décrit un appel de bloc ainsi que l'utilisation de la déclaration d'interface d'un bloc dans une structure d'appel.

	<p>Le FB 1 n'est pas appelé</p> <p>Le FB 1 utilise le FB 2 en tant que multiinstance dans sa propre déclaration d'interface</p> <p>Le FB 2 utilise l'UDT 3 dans sa propre déclaration d'interface. Il n'est pas indiqué ici que le FB 2 appelle le FB 3, cependant cela est indiqué par l'icône d'appel, cf. (3)</p> <p>Le FB 1 appelle la multiinstance FB 2 (Call) et le FB 2 utilise l'UDT 3 dans sa déclaration d'interface et le FB 2 appelle la FC 3 (Call)</p>
---	---





Représentation d'appels de blocs non existants

Lorsqu'un bloc utilisé par un autre bloc est effacé, le texte de tels blocs est écrit en rouge et se termine par "???".

Représentation de récursions dans des interdépendances de blocs

Les récursions résultent des interdépendances suivantes de blocs :

- Le bloc 1 appelle le bloc 2 **et** le bloc 2 appelle le bloc 1.
- Le bloc 1 appelle le bloc 2 **et** le bloc 2 utilise la déclaration d'interface du bloc 1, p. ex. : le FB1 appelle le DB d'instance du FB 1
- Le bloc 1 utilise la déclaration d'interface du bloc 2 **et** le bloc 2 utilise la déclaration d'interface du bloc 1. Cette constellation n'est pas autorisée et peut uniquement résulter de la copie de blocs dans un projet existant. Les blocs concernés ne peuvent plus être compilés.
- Les récursions dans les interdépendances de blocs sont représentées par ajout d'une flèche :

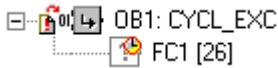
Icône	Signification
	Récursion et appel de bloc avec CALL
	Récursion et appel de bloc avec UC
	Récursion et appel de bloc avec CC
	Récursion résultant d'une déclaration d'interface dans le bloc utilisé

Représentation de conflits d'horodatage dans les interdépendances de blocs

Des horodatages d'interfaces différents peuvent entraîner des conflits, lorsque l'interface du bloc appelé a été modifiée (p. ex. possède moins de variables), mais que le bloc appelant utilise encore l'"ancienne" déclaration d'interface pour l'appel.

L'indicateur d'éventuelles interfaces incohérentes est l'horodatage d'interface. Lorsque ce dernier est plus récent pour le bloc appelé que pour le bloc appelant, cela est signalé dans la structure du programme par ajout d'une icône d'horloge.





En cas de conflit, le bloc appelant est précédé d'une icône d'état, indiquant qu'il doit être compilé une nouvelle fois.

Exemple de représentation de conflits d'horodatage	
--	---

Représentation de conflits de mnémoniques dans les interdépendances de blocs

Lorsque 'Classement symbolique' est sélectionné dans les paramètres généraux du projet, des incohérences (conflits entre mnémoniques) risquent de se produire lorsque vous enregistrez un bloc, puis que vous modifiez l'un des mnémoniques utilisés dans ce bloc dans la table des mnémoniques.

Les icônes des interdépendances de blocs sont alors les suivantes :

Icône	Signification
	Conflit de mnémoniques et appel de bloc avec CALL
	Conflit de mnémoniques et appel de bloc avec UC
	Conflit de mnémoniques et appel de bloc avec CC
	Conflit de mnémonique résultant d'une déclaration d'interface dans le bloc utilisé

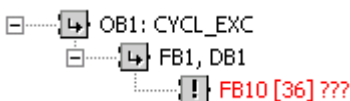
Représentation de numéros de blocs dans le cas d'entrées indirectes ou de l'utilisation du type de paramètre BLOCK

Pour les blocs entrés de manière indirecte dans l'appel (p. ex. UC FC[MW 10]), le numéro est représenté par une série de points d'interrogation :



Lorsqu'un paramètre d'entrée de type BLOCK a été utilisé dans la déclaration d'interface d'un bloc (p. ex. BLOCK_FB), le numéro de bloc est défini par le bloc appelant. Lorsque STEP 7 Lite est en mesure de déterminer le numéro de manière récursive, il est également affiché. Dans ce cas, le texte placé à côté de l'icône du bloc est représenté en rouge dans la structure du programme. Lorsque le numéro de blocs ne peut pas être déterminé, p. ex. parce que le bloc avec le type de paramètre BLOCK n'est pas appelé, il est représenté par des points d'interrogation.

Exemple :

L'OB 1 affecte la valeur "FB 10" au paramètre BLOCK_FB dans l'appel du FB 1	
---	---

6.5.6 Utilisation des données de référence

6.5.6.1 Positionnement rapide sur des occurrences dans le programme

Pendant l'élaboration du programme, vous pouvez vous servir des données de référence pour sauter à l'occurrence d'un opérande.

Marche à suivre

1. Sélectionnez l'opérande en question dans un bloc ouvert.
2. Choisissez la commande **Edition > Aller à > Occurrence**.
Ceci ouvre une boîte de dialogue énumérant les occurrences de cet opérande dans le programme.
3. Activez l'option "Pour tous les opérandes de la plage d'adresses spécifiée" si vous voulez afficher en plus les occurrences des opérandes dont l'adresse ou la plage d'adresses recouvre partiellement celle de l'opérande appelé. Une colonne "Opérande" s'ajoute au tableau.
4. Sélectionnez une occurrence dans la liste et cliquez sur le bouton "Aller à".

Liste des occurrences

Dans la boîte de dialogue, la liste des occurrences donne les indications suivantes :

- bloc dans lequel l'opérande est utilisé,
- mnémonique du bloc, s'il existe,
- détails, c'est-à-dire informations sur l'occurrence dépendant du langage de création du bloc/de la source (SCL) et, le cas échéant, de l'opération,
- type d'accès à l'opérande : en lecture (R), en écriture (W), en lecture et en écriture (RW), indéterminable (?),
- langage du bloc.

Vous pouvez filtrer l'affichage des occurrences pour n'afficher, par exemple, que les accès en écriture à un opérande. L'aide en ligne sur la boîte de dialogue renseigne en détail sur les possibilités de saisie et d'affichage.

Avertissement

Les données de référence existent seulement hors ligne. Cette fonction utilise donc toujours les références croisées des blocs hors ligne, même si vous l'appellez dans un bloc en ligne.

6.5.6.2 Exemple d'utilisation d'occurrences

Vous souhaitez savoir où la sortie A1.0 est mise à 1 (directement/indirectement). Nous emploierons comme exemple le code LIST suivant dans le bloc OB1 :

```

Réseau 1 : .....
U A 1.0    // Insignifiant
= A 1.1    // dans cet exemple

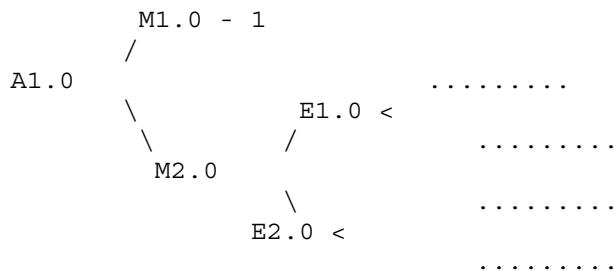
Réseau 2 :
U M1.0
U M2.0
= A 1.0    // Affectation

Réseau 3 :
//Ligne de commentaire seulement
SET
= M1.0     // Affectation

Réseau 4 :
U E 1.0
U E 2.0
= M2.0     // Affectation

```

Il en résulte le schéma d'affectation suivant pour A1.0 :



Vous procédez alors de la manière suivante :

1. Dans l'éditeur de bloc, positionnez-vous dans l'OB1 sur (Re 1, Inst 1).
2. Choisissez **Edition > Aller à > Occurrence** dans la barre des menus ou "Occurrence" dans le menu contextuel (bouton droit de la souris).

La boîte de dialogue affiche entre autres toutes les affectations à A1.0 :

OB1	Cycle Execution	Re 2	Inst 3	/=	W	LIST
OB1	Cycle Execution	Re 1	Inst 1	/U	R	LIST

3. Au moyen de "Aller à", sautez de la boîte de dialogue à "Re 2 Inst 3" dans l'éditeur :

```

Réseau 2 :
U M1.0
U M2.0
= A 1.0

```

4. Il faut maintenant contrôler les affectations à M1.0 comme à M2.0. Positionnez-vous donc pour commencer sur M1.0 dans l'éditeur de bloc.

5. Choisissez **Edition > Aller à > Occurrence** dans la barre des menus ou "Occurrence" dans le menu contextuel. La boîte de dialogue affiche entre autres toutes les affectations à M1.0 :

```
OB1      Cycle Execution      Re 3   Inst 2   /=      W      LIST
OB1      Cycle Execution      Re 2   Inst 1   /U      R      LIST
```

6. Au moyen de "Aller à", sautez à "Re 3 Inst 2" dans l'éditeur de bloc.
7. Dans le réseau 3 de l'éditeur de bloc, nous constatons que l'affectation de M1.0 ne nous intéresse pas (puisque'il est toujours TRUE) et qu'il faudrait donc examiner à la place celle de M2.0.
8. Amenez au premier plan la boîte de dialogue "Aller à l'occurrence" encore ouverte ou appelez "Aller à l'occurrence" depuis la position actuelle dans l'éditeur de bloc.
9. Au moyen de "Aller à" (voir n° 3), sautez de la boîte de dialogue à "Re 2 Inst 3" dans l'éditeur :

```
Réseau 2 :
U M1.0
U M2.0
= A 1.0
```

10. Au n° 4 et aux suivants, nous avons contrôlé l'affectation à M1.0. Il faut contrôler à présent toutes les affectations (directes/indirectes) à M2.0. Positionnez-vous donc sur M2.0 dans l'éditeur et appelez "Aller à l'occurrence" : toutes les affectations à M2.0 s'affichent, entre autres :

```
OB1      Cycle Execution      Re 4   Inst 3   /=      W      LIST
OB1      Cycle Execution      Re 2   Inst 2   /U      R      LIST
```

11. Au moyen de "Aller à", sautez à "Re 4 Inst 3" dans l'éditeur :

```
Réseau 4 :
U E 1.0
U E 2.0
= M2.0
```

12. Il faut contrôler à présent les affectations à E1.0 et E2.0. Nous y renoncerons dans cet exemple, car la marche à suivre est identique à celle décrite jusqu'ici (n° 4 et suivants).

En sautant de l'éditeur de bloc à la boîte de dialogue des occurrences et vice-versa, vous pouvez déterminer les positions essentielles de votre programme et les vérifier.

6.5.6.3 Marche à suivre pour utiliser des données de référence

Saut de la liste des références croisées à la partie de programme

Pour sauter de la liste des références croisées à la partie de programme correspondante :

- Sélectionnez la ligne de l'opérande souhaité.
- Cliquez sur le bouton "Aller à l'occurrence"
- Vous pouvez également exécuter cette fonction via la commande de menu **Edition > Aller à > Occurrence**

Vous pourriez également procéder de la manière suivante :

1. Sélectionnez un opérande dans la liste des références croisées.
2. Cliquez avec le bouton droit de la souris pour afficher le menu contextuel.
3. Choisissez la commande **Aller à l'occurrence**.

Saut de la structure du programme à la partie de programme

Pour sauter de la structure du programme à la partie de programme correspondante :

1. Sélectionnez un bloc dans la structure du programme.
2. Cliquez avec le bouton droit de la souris pour afficher le menu contextuel.
3. Choisissez la commande **Aller au bloc** pour ouvrir le bloc même ou bien la commande **Aller à l'appel** pour ouvrir le bloc de niveau supérieur et vous y positionnez sur l'appel du bloc sélectionné.

La commande **Aller à l'appel** n'est disponible que s'il existe un bloc de niveau supérieur au bloc sélectionné.

Ces commandes du menu contextuel figurent aussi dans la barre des menus :

Edition > Aller à > Bloc et

Edition > Aller à > Appel

Affichage des accès se recouvrant partiellement

Pour afficher les références croisées des opérandes dont les plages d'adresses se chevauchent, procédez de la manière suivante :

1. Sélectionnez un opérande dans la liste des références croisées.
2. Appuyez sur le bouton droit de la souris et choisissez dans le menu contextuel **Références croisées de l'opérande**.

Une autre fenêtre affiche alors en plus les références croisées des opérandes dont les plages d'adresses recouvrent en partie celle de l'opérande sélectionné.

Dans l'éditeur de bloc, procédez comme suit :

1. Sélectionnez l'opérande dans la section des instructions.
2. Choisissez la commande **Edition > Aller à > Occurrence**.
3. Dans la boîte de dialogue "Aller à l'occurrence", sélectionnez l'option "Pour tous les opérandes de la plage spécifiée".

6.6 Vérification de la cohérence du programme et horodatage comme propriété de bloc

6.6.1 Vérification de la cohérence du programme

Introduction

Si, durant ou après la création du programme, vous devez adapter ou compléter les interfaces ou le code de blocs individuels, il peut en résulter des incohérence de programmation entre des blocs ou blocs de référence appelants et appelés.

La fonction de vérification de la cohérence du programme se charge d'une grande partie de ce travail de correction. Elle résout automatiquement la plupart des conflits d'horodatage et d'incohérences de programmation. Dans le cas de blocs pour lesquels les incohérences de programmation ne peuvent pas être résolues automatiquement, cette fonction vous conduit aux positions à modifier dans l'éditeur de blocs. Vous y effectuez les modifications nécessaires. Vous corrigez ainsi toutes les incohérences de programmation les unes après les autres et compilez les blocs.

Vérification de la cohérence du programme

Lorsque vous lancez la vérification de la cohérence du programme, l'horodatage des interfaces des blocs est vérifié et les blocs susceptibles de provoquer des incohérences de programmation sont sélectionnés.

1. Choisissez la commande de menu **Outils > Vérifier la cohérence**.
STEP 7 Lite résout automatiquement (dans la mesure du possible) les conflits d'horodatage et d'incohérence de programmation, puis compile les blocs. Lorsqu'un conflit d'horodatage ou une incohérence ne peuvent pas être corrigés automatiquement dans un bloc, un message d'erreur est émis.
Les blocs concernés sont affichés en rouge (= erronés) et en gras dans la fenêtre du projet.
2. Dans la fenêtre du projet, effectuez un double clic sur un bloc affiché en rouge.
Le bloc s'ouvre. Les erreurs s'affichent dans la barre des résultats de l'éditeur de bloc.
3. Effectuez un double clic sur une entrée de la barre des résultats.
La position erronée est indiquée.
4. Corrigez toutes les erreurs, puis appliquez les modifications. Fermez le bloc. Répétez cette procédure pour tous les blocs erronés. Une fois toutes les erreurs corrigées, le bloc correspondant est affiché en noir dans le fenêtre du projet pour y être vérifié.
5. Enregistrez le projet

Création automatique de DB d'instance lors de la vérification de la cohérence du programme

Si vous chargez un FB dans le DB d'instance correspondant ainsi que le bloc appelant le FB depuis la CPU dans la PG, un DB d'instance sera automatiquement créé pour le FB, lors de la vérification de la cohérence du programme.

6.6.2 Horodatage et conflits d'horodatage

Les blocs contiennent un horodatage de code et un horodatage d'interface. Ces horodatages s'affichent dans l'onglet "Propriétés" de l'éditeur de bloc. Ils permettent de surveiller la cohérence des programmes STEP 7 Lite.

STEP 7 Lite signale un conflit d'horodatage lorsque la comparaison des horodatages révèle une des infractions suivantes aux règles :

- Un bloc appelé est plus récent que le bloc appelant (CALL).
- Un bloc référencé est plus récent que le bloc qui l'utilise. Exemples :
 - Un UDT est plus récent que le bloc qui l'utilise, par ex. un DB ou un autre UDT, ou un FC, FB, OB qui utilise cet UDT dans la table de déclaration des variables.
 - Un FB est plus récent que le DB d'instance correspondant.
 - Un FB2 est défini comme multi-instance dans un FB1, or FB2 est plus récent que FB1.

Nota

Même quand la relation entre les horodatages d'interface est correcte, des incohérences peuvent se produire :

- la définition de l'interface du bloc référencé ne concorde pas avec l'interface utilisée à son occurrence.
 - De telles incohérences sont appelées conflits d'interface. Elles peuvent résulter, par exemple, de la copie de blocs provenant de programmes différents.
-

6.6.3 Horodatage des blocs de code

Horodatage de code

L'instant de création du bloc y est inscrit. L'horodatage est actualisé en cas de

- modification du code du programme,
- modification de la description d'interface,
- modification du commentaire,
- modification des propriétés de bloc.

Horodatage d'interface

L'horodatage est actualisé en cas de

- modification de la description d'interface (modification de types de données ou de valeurs initiales, nouveaux paramètres).

L'horodatage n'est pas actualisé en cas de

- modification de mnémoniques,
- modification de commentaires dans la table déclaration de variables,
- modifications dans la zone TEMP.

Règles pour les appels de bloc

- L'horodatage d'interface du bloc appelé doit être antérieur à l'horodatage de code du bloc appelant.
- Ne modifiez l'interface d'un bloc que si aucun bloc l'appelant n'est ouvert. En effet, si vous enregistrez les blocs appelants après le bloc modifié, cette incohérence ne vous sera pas signalée par l'horodatage.

En cas de conflit d'horodatage

Un conflit d'horodatage est signalé à l'ouverture du bloc appelant. Après modification de l'interface d'une FC ou d'un FB, tous les appels de ce bloc sont représentés en rouge dans les blocs appelants.

Quand vous modifiez l'interface d'un bloc, il faut adapter tous les blocs qui l'appellent.

Après modification de l'interface d'un FB, vous devez actualiser les définitions de multi-instances et les blocs de données existants.

6.6.4 Horodatage des blocs de données globaux

Horodatage de code

L'horodatage est actualisé

- à la première création,
- en cas de modifications dans la vue des déclarations ou dans la vue des données du bloc.

Horodatage d'interface

L'horodatage est actualisé

- en cas de modifications de la description d'interface dans la vue des déclarations (modification de types de données ou de valeurs initiales, nouveaux paramètres).

6.6.5 Horodatage des blocs de données d'instance

Un bloc de données d'instance mémorise les paramètres formels et les données statiques de blocs fonctionnels.

Horodatage de code

L'instant de création du bloc de données d'instance y est inscrit. L'horodatage est actualisé lorsque vous saisissez des valeurs effectives dans la vue des données du bloc de données d'instance. L'utilisateur ne peut pas modifier la structure d'un bloc de données d'instance, car elle est dérivée du bloc fonctionnel (FB) ou du bloc fonctionnel système (SFB) correspondant.

Horodatage d'interface

Quand un bloc de données d'instance est créé, l'horodatage d'interface du FB ou du SFB correspondant y est inscrit.

Règles pour une ouverture sans conflit

L'horodatage d'interface du FB/SFB et celui du bloc de données d'instance associé doivent concorder.

En cas de conflit d'horodatage

Lorsque vous modifiez l'interface d'un FB, son horodatage d'interface est mis à jour. L'ouverture d'un bloc de données d'instance associé signale alors un conflit d'horodatage, puisque celui du DB et celui du FB ne concordent plus. Dans la section de déclaration du DB, l'interface est représentée avec des mnémoniques générés par le compilateur (pseudo-mnémoniques). Le bloc de données d'instance ne peut plus servir qu'à la consultation.

Pour résoudre un tel conflit d'horodatage, vous devez générer de nouveau le DB d'instance associé au FB modifié.

6.6.6 Horodatage des UDT et des DB dérivés d'UDT

Vous pouvez utiliser des types de données utilisateur (UDT), par exemple pour créer plusieurs blocs de données de structure identique.

Horodatage de code

L'horodatage de code est actualisé à chaque modification.

Horodatage d'interface

L'horodatage d'interface est actualisé lors de la modification de la description d'interface (modification de types de données ou de valeurs initiales, nouveaux paramètres).

Règles pour une ouverture sans conflit

- L'horodatage d'interface du type de données utilisateur doit être antérieur à celui des blocs de code dans lequel ce type de données est utilisé.
- L'horodatage d'interface du type de données utilisateur doit être identique à celui d'un DB dérivé de l'UDT.
- L'horodatage d'interface du type de données utilisateur doit être postérieur à celui d'un UDT qui y est contenu.

En cas de conflit d'horodatage

Lorsque vous modifiez la définition d'un UDT utilisée dans un DB, une FC, un FB ou une autre définition d'UDT, STEP 7 Lite signale un conflit d'horodatage à l'ouverture d'un tel bloc.

Le composant UDT est représenté non assemblé, sous forme de structure. Tous les noms de variable sont remplacés par des valeurs par défaut du système.

6.6.7 Comment éviter des erreurs lors de l'appel de bloc

STEP 7 Lite écrase des données dans le registre DB

STEP 7 Lite modifie les registres des CPU S7-300 lors de certaines opérations. Les contenus des registres DB et DI sont, par exemple, permutés lors de l'appel d'un FB : cela permet d'ouvrir le DB d'instance du FB appelé sans perdre l'adresse du DB d'instance précédent.

En adressage absolu, des erreurs peuvent se produire lors de l'accès à des données figurant dans les registres : dans certains cas, les adresses dans le registre d'adresse 1 (AR1) et dans le registre de DB sont écrasées. Il se peut donc que vous lisiez des adresses erronées ou que vous écriviez à des adresses erronées.



Danger

Il existe un risque de dégâts matériels et de dommages physiques lorsque vous utilisez :

1. CALL FC, CALL FB, CALL multi-instance,
2. des accès à un DB indiqués intégralement (par ex. DB20.DBW10),
3. des accès à des variables de type de données complexe.

Il est possible que les contenus des registres de bloc de données (DB et DI), des registres d'adresse (AR1, AR2) et des accumulateurs (ACCU1 et ACCU2) soient modifiés.

En outre, il n'est pas possible d'utiliser le résultat logique RLG comme paramètre supplémentaire (implicite) lors de l'appel d'une fonction ou d'un bloc fonctionnel.

Si vous utilisez les méthodes de programmation ci-dessus, vous devez vous-même faire en sorte que ces contenus soient corrects afin d'éviter tout dysfonctionnement.

Enregistrement de données correctes

Le contenu du registre de DB s'avère tout particulièrement important lorsque vous accédez à des données en format abrégé de l'adresse absolue. Si, par exemple, vous partez du principe que le DB20 est ouvert (son numéro est donc enregistré dans le registre DB), vous pouvez indiquer DBX0.2 pour accéder aux données figurant dans le bit 2 de l'octet 0 du DB dont l'adresse figure dans le registre DB, donc le DB20. Toutefois, si le registre DB contient une autre adresse, vous accédez à des données erronées.

Pour éviter toute erreur lors d'accès aux données du registre DB, nous vous conseillons :

- d'utiliser l'adressage symbolique,
- de donner l'adresse absolue complète (par ex. DB20.DBX0.).

Avec ces deux méthodes d'adressage, STEP 7 Lite ouvre automatiquement le bon DB. Si vous utilisez le registre AR1 pour l'adressage indirect, vous devez toujours charger l'adresse correcte dans AR1.

Situation dans lesquelles les registres sont modifiés

La manipulation des registres d'adresse pour l'adressage indirect ne concerne que le langage LIST. Les autres langages n'autorisent pas l'accès indirect aux registres d'adresse.

En revanche, il faut tenir compte de la modification du registre du DB par le compilateur dans tous les langages de programmation afin de garantir une transmission correcte des paramètres lors d'appels de bloc.

Le contenu du registre d'adresse AR1 et du registre de DB du bloc appelant est écrasé dans les situations suivantes :

Situation	Signification
Paramètres effectifs provenant d'un DB	<ul style="list-style-type: none"> Une fois que vous avez affecté à un bloc un paramètre effectif qui est enregistré dans un bloc de données (par ex. DB20.DBX0.2), STEP 7 Lite ouvre ce bloc de données (DB20) et modifie le contenu du registre de DB en conséquence. Après l'appel de bloc, le programme utilise alors le DB modifié.
Appel de bloc en relation avec des types de données complexes	<ul style="list-style-type: none"> Le contenu du registre AR1 et du registre de DB du bloc appelant est modifié après un appel de bloc dans une FC qui transmet un composant d'un paramètre formel de type de données complexe (chaîne, tableau, structure ou UDT) au bloc appelé. Il en est de même lors d'un appel dans un FB si le paramètre se situe dans la zone var_in_out du bloc appelant.
Accès à des composants de type de données complexe	<ul style="list-style-type: none"> STEP 7 Lite utilise le registre d'adresse AR1 et le registre de DB lors de l'accès d'un FB à un composant d'un paramètre formel de type de données complexe dans la zone var_in_out (chaîne, tableau, structure ou UDT). Cela entraîne donc la modification du contenu de ces deux registres. STEP 7 Lite utilise le registre d'adresse AR1 et le registre de DB lors de l'accès d'une FC à un composant d'un paramètre formel de type de données complexe (chaîne, tableau, structure ou UDT). Cela entraîne donc la modification du contenu de ces deux registres.

Avertissement

- Lors de l'appel d'un FB dans un bloc de version 1, le paramètre effectif pour le premier paramètre booléen IN ou IN_OUT n'est pas transmis correctement si l'opération avant l'appel ne délimite pas les RLG. Dans ce cas, ce paramètre est combiné au RLG existant.
- Il y a écriture dans le registre d'adresse AR2 lors de l'appel d'un FB (simple ou multi-instance).
- Le traitement correct d'un FB n'est plus garanti si le registre d'adresse AR2 est modifié à l'intérieur de ce FB.
- Si l'adresse absolue du DB n'est pas transmise en entier à un paramètre ANY, le pointeur ANY ne contient pas le numéro du DB ouvert, mais toujours le numéro 0.

6.6.8 Remarques sur la modification du contenu des registres

Si vous utilisez les possibilités de programmation décrites ci-après ainsi que les registres/ACCU cités, vous devez veiller à restaurer vous-même le contenu des registres ou des ACCU afin d'éviter des erreurs.

L'utilisation des éléments de langage évolué suivants peut entraîner la modification du contenu des **registres DB** et du **registre d'adresses AR1** :

- accès à un DB indiqué intégralement (par ex. DB20.DBW10) en tant que paramètre effectif de FC,
- appel de FB et de multi-instance,
- élément de structure d'un paramètre formel en tant qu'opérande dans une FC ou un FB,
- élément de structure d'un paramètre formel en tant que paramètre effectif de FC ou FB.

RL ou **ACCU1** et **ACCU2** ne doivent pas être utilisés comme paramètres (implicites) supplémentaires dans les appels de FB, FC et multi-instance.

Le **registre DI** et le **registre d'adresses AR2** sont utilisés par le système pour l'appel de FB et de multi-instance et ne doivent donc pas être modifiés dans les FB.

Le **registre d'adresses AR1** est utilisé par une partie des blocs standard chargeables.

Dans un FB, l'instruction "L P#Nom_paramètre" charge le décalage d'adresse du paramètre spécifié par rapport au **registre d'adresses AR2**. Pour fournir le décalage absolu dans le bloc de données d'instance pour les FB multi-instance, le pointeur (uniquement l'adresse) interne à la zone du registre AR2 doit encore être additionné à cette valeur.

De plus amples informations sur les registres de CPU figurent dans l'aide sur le langage de programmation (CONT/LOG/LIST).

7 Etablissement d'une liaison en ligne et choix de la CPU

7.1 Etablissement de liaisons en ligne

Une liaison en ligne entre la PG/le PC et la CPU est nécessaire pour charger des programmes utilisateurs et des blocs, pour charger des blocs de la CPU dans la PG/le PC et pour exécuter les tâches suivantes :

- test de programmes utilisateur ,
- affichage de l'état de fonctionnement de la CPU et changement d'état,
- affichage et réglage de l'heure et de la date de la CPU,
- affichage de l'état du module,
- comparaison de blocs en ligne/hors ligne,
- diagnostic du matériel.

Pour pouvoir établir une liaison en ligne, il faut que la PG/le PC et la CPU soient reliés entre eux via l'interface MPI.

Si deux programmes (Par ex. deux PG/PC ou un PG/PC avec STEP 7 Lite et STEP 7) accèdent à la même CPU, les informations Online doivent être actualisées régulièrement via la touche "F5".

STEP 7 Lite immédiatement "en ligne"

Aussitôt après son démarrage, STEP 7 Lite tente d'établir une liaison en ligne à la CPU.

S'il ne peut joindre **aucune** CPU, il reste hors ligne. Même si vous le démarrez à nouveau, aucune liaison en ligne n'est établie. Dans ce cas, vous devez éliminer ce qui empêche d'établir la liaison en ligne, puis cliquer sur le bouton "En ligne/Hors ligne".

Sans liaison en ligne, il n'y a pas d'icônes de synchronisation dans la fenêtre du projet et il n'est pas possible d'y sélectionner le panneau de commande CPU ni la vue CPU.

Alternance entre En ligne et Hors ligne

Vous trouvez dans la barre d'outils (et dans le menu) un bouton "En ligne/Hors ligne" servant à établir ou à suspendre la liaison à la CPU. Ce bouton est enfoncé quand la liaison existe et saillant quand elle n'existe pas. Une fois la liaison établie, les fenêtres désactivées redeviennent sélectionnables et les icônes de synchronisation réapparaissent. L'adresse MPI de la CPU est affichée entre crochets dans la barre de titre du panneau de commande de la CPU.

7.1.1 Protection de l'accès aux systèmes cible par mot de passe

La protection par mot de passe vous permet

- de protéger le programme utilisateur et ses données dans la CPU contre les modifications involontaires (protection en écriture),
- de préserver le savoir faire (know how) contenu dans votre programme utilisateur (protection en lecture),
- d'empêcher les fonctions en ligne qui gêneraient le processus.

Pour que vous puissiez protéger un module par un mot de passe, il faut qu'il possède cette fonctionnalité.

Paramétrage de la protection par mot de passe

Si vous souhaitez protéger un module par un mot de passe, vous devez définir le niveau de protection et le mot de passe dans le cadre du paramétrage de la CPU, puis charger le paramétrage modifié dans la CPU.

Pour paramétrer la CPU en conséquence, effectuez un double clic sur "Hardware" dans la fenêtre du projet, puis un double clic sur la CPU à l'emplacement 2. Vous paramétrez le niveau de protection et le mot de passe dans la section "Protection".

Interrogation du mot de passe en fonctionnement

Quand l'exécution d'une fonction en ligne nécessite la saisie d'un mot de passe, la boîte de dialogue "Saisie du mot de passe" s'affiche. La saisie du mot de passe correct vous donne l'autorisation d'accéder à la CPU pour laquelle vous avez défini un niveau de protection particulier dans le cadre du paramétrage. Vous pouvez alors établir des connexions en ligne au module protégé et exécuter les fonctions en ligne correspondant au niveau de protection.

Vous pouvez aussi saisir le mot de passe dans le panneau de commande élargi de la CPU (en cliquant ensuite sur le bouton "Ouvrir la session"). Le panneau de commande élargi de la CPU vous permet également de fermer la liaison en ligne ouverte au moyen du mot de passe. Un nouvel accès est alors uniquement possible avec la nouvelle saisie du mot de passe.

7.2 Affichage de l'état de fonctionnement et changement

Cette fonction vous permet, par exemple, de remettre la CPU à l'état de marche (RUN) après avoir corrigé une erreur.

Affichage de l'état de fonctionnement dans le panneau de commande CPU

La condition requise est l'établissement d'une liaison en ligne à la CPU.

Si le panneau de commande de la CPU n'est pas ouvert, cliquez sur la flèche double "Accéder à la CPU en ligne" au-dessus de la fenêtre du projet.

La flèche double vers le bas permet d'ouvrir la fenêtre réduite à son icône.

Une fois ouvert, le panneau de commande CPU indique l'état de fonctionnement en cours ainsi que la position actuelle du commutateur de mode de fonctionnement. Selon le cas, le commutateur de mode de fonctionnement de la CPU est représenté comme un commutateur à clé, un commutateur à bascule ou un commutateur rotatif. Vous avez sous les yeux une image de la platine avant de la CPU.

Changement d'état de fonctionnement dans le panneau de commande CPU

Les boutons RUN et STOP servent à faire passer la CPU à un autre état de fonctionnement. Selon l'état de fonctionnement en cours, ils sont activables ou pas.

7.3 Affichage et réglage de l'heure et de la date

Procédez de la manière suivante :

1. Ouvrez le panneau de commande CPU (Accéder à la CPU en ligne).
2. Cliquez sur la flèche double vers la droite. Vous voyez s'ouvrir la vue élargie du panneau de commande. La zone "Régler l'heure" indique l'horodatage de la PG/PC et celui de la CPU.
 - Pour que la CPU adopte l'horodatage du PG/PC, cochez la case "Adopter l'heure PG/PC", puis cliquez sur le bouton "Régler".
 - Pour régler l'heure indépendamment du PG/PC, retirez la coche de la case "Adopter l'heure PG/PC", tapez l'heure de la CPU, puis cliquez sur le bouton "Régler".

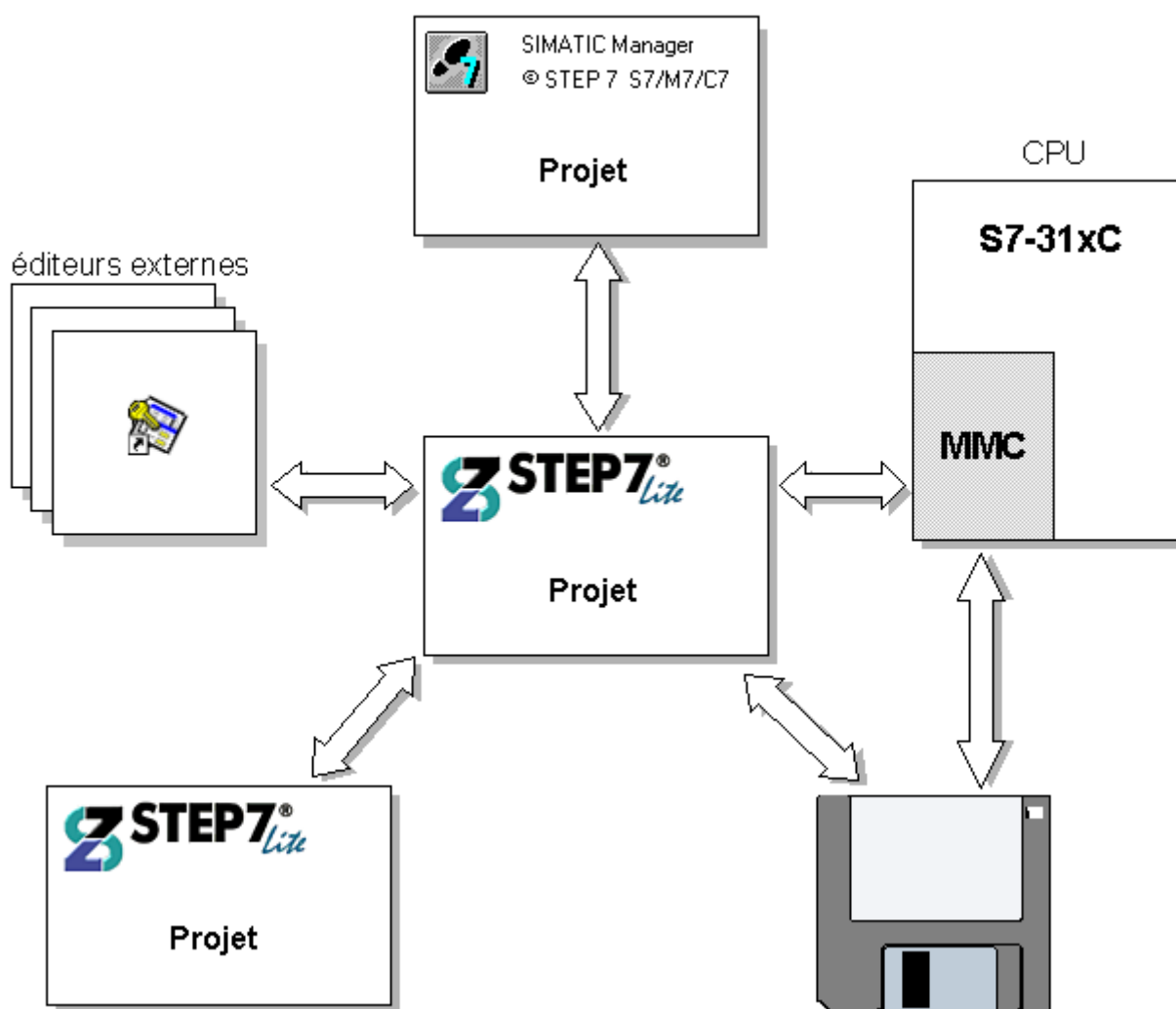
Nota

Lorsqu'un module ne dispose pas d'horloge temps réel intégrée, la date affichée est 00.00.00 et l'heure affichée est 00:00:00.

8 Importer, exporter, enregistrer sous

8.1 Importer, exporter, enregistrer sous

L'illustration suivante représente les sources et cibles possibles pour les fonctions d'importation/exportation et les fonctions d'enregistrement de STEP 7 Lite.



8.2 Enregistrer des projets sur un support de données

Vous avez la possibilité d'enregistrer le projet STEP 7 Lite complet p. ex. sur un disque dur ou sur une disquette.

Avec la commande **Fichier > Enregistrer**, toutes les modifications effectuées dans le projet sont enregistrées dans le projet initial ouvert.

Avec la commande **Fichier > Enregistrer sous**, vous pouvez enregistrer le projet ouvert avec toutes les modifications sous un nouveau nom dans un autre chemin ou sur un autre support de données.

Nota

Il est à noter que la commande **Edition > Appliquer** n'enregistre pas les contenus du projet. Après des modifications, elle permet de rétablir la cohérence de différentes vues ouvertes dans STEP 7 Lite.

8.3 Enregistrer des données de projet sur une carte mémoire Micro (MMC)

Avec STEP 7 Lite, vous avez la possibilité d'enregistrer les données de votre projet STEP 7 Lite sur la carte mémoire SIMATIC Micro (MMC) d'une CPU 31xC. Vous pouvez ainsi également accéder aux données du projet avec des consoles de programmation sur lesquelles le projet n'est pas enregistré.

Quelles données de projet est-il possible d'enregistrer sur la carte mémoire MMC ?

Dans STEP 7 Lite, vous pouvez enregistrer les données de projet suivantes sur une carte MMC, dans les formats indiqués :

- Le projet complet sous forme de fichier *.k7p.
- Des blocs sélectionnés et la table des mnémoniques sous forme de "fichier d'exportation S7Lite" (*.k7e).
- Tous les blocs du programme utilisateur sous forme de fichier *.awl.
- La liste des mnémoniques sous forme de fichier *.sdf.

Conditions requises

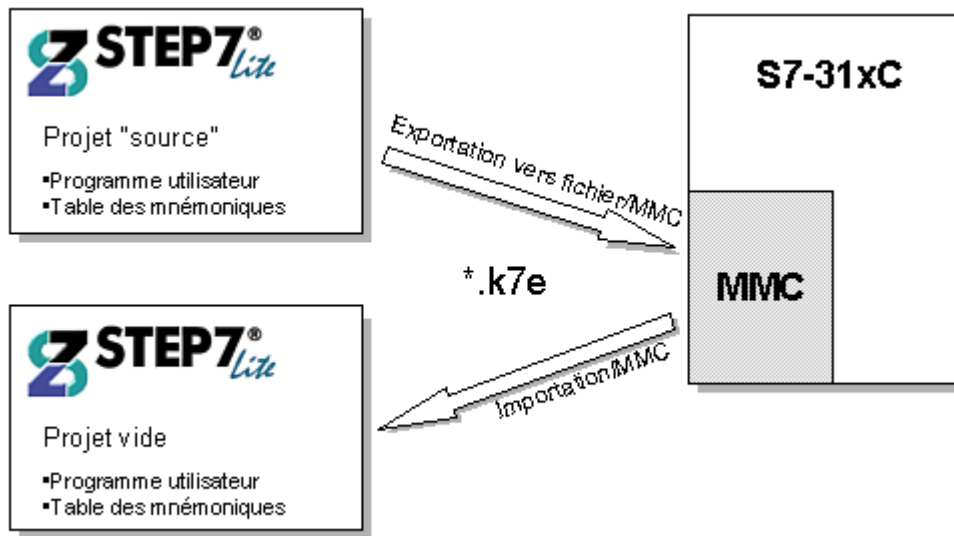
L'enregistrement de données du projet sur une carte MMC est uniquement possible lorsque celle-ci est enfichée dans le logement de la CPU 31xC et qu'une liaison en ligne est établie avec cette CPU.

La taille de la carte MMC doit être suffisante pour y enregistrer toutes les données.

Marche à suivre pour enregistrer le projet complet

1. Choisissez la commande de menu **Fichier > Enregistrer sous**.
2. Dans la boîte de dialogue qui s'ouvre, choisissez l'onglet "Carte mémoire".
3. Dans le champ "Nom de fichier", entrez un nom de fichier spécifique sans extension.
4. Dans la liste déroulante "Type de fichier", sélectionnez le type de fichier "Projets (*.k7p)".
5. Cliquez sur le bouton "Enregistrer".

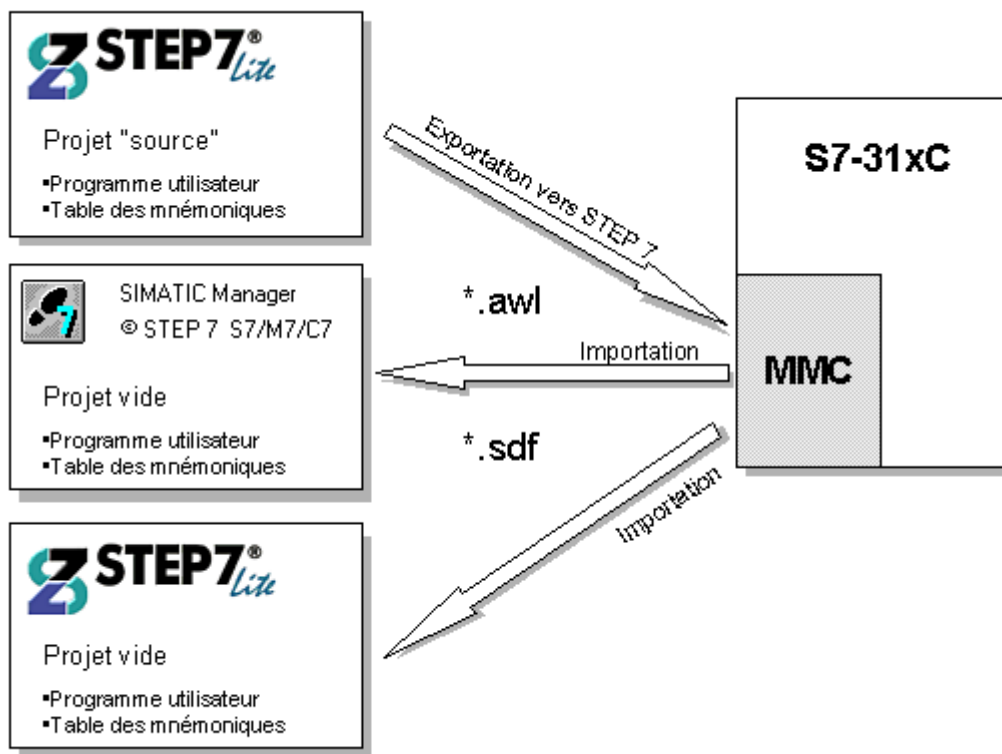
Marche à suivre pour enregistrer des blocs sélectionnés et la table des mnémoniques sous forme de "fichier d'exportation S7Lite"



1. Choisissez la commande de menu **Fichier > Exporter > Dans fichier**.
2. Dans la boîte de dialogue qui s'ouvre, choisissez l'onglet "Carte mémoire".
3. Cliquez sur le bouton "Exporter"

Marche à suivre pour enregistrer les données de projet sous forme de fichier *.awl et *.sdf

L'avantage de l'enregistrement des données du projet sous forme de fichier *.awl et de fichier *.sdf réside dans le fait que ces données projet pourront également être importées dans les versions futures de STEP 7.



1. Sélectionnez l'icône "Programme" ou l'icône "Table des mnémoniques" dans la fenêtre du projet.
(si vous souhaitez enregistrer les deux éléments sur la carte MMC, sélectionnez-les en maintenant la touche CTRL enfoncée).
2. Choisissez la commande de menu **Fichier > Exporter > Pour STEP 7**.
3. Choisissez l'onglet "Carte mémoire".
4. Choisissez le nom du fichier d'exportation sans extension. Le nom du fichier à générer s'affiche avec son chemin dans la champ "Fichier généré".
5. Cliquez sur le bouton "Exporter".

8.4 Utiliser une carte mémoire Micro comme support de données

Avec STEP 7 Lite, la carte mémoire SIMATIC Micro (MMC) d'une CPU 31xC peut être mise en œuvre comme tout support de données externe normal. À condition que la taille de la carte mémoire MMC est suffisante, il est possible d'y enregistrer tous les fichiers visibles dans l'Explorateur de fichiers du système d'exploitation. Vous pouvez ainsi mettre à disposition de vos collaborateurs des schémas, instructions de maintenance ou descriptions de fonctions supplémentaires sur votre installation.

Marche à suivre pour transférer des fichiers sur la carte mémoire MMC

1. Ouvrez la vue "Carte mémoire" dans l'onglet "CPU en ligne" de la fenêtre du projet.
2. Choisissez la commande de menu **Fichier > Transférer les fichiers > Sur la carte mémoire**.
3. Dans la fenêtre de sélection, naviguez jusqu'aux fichiers que vous souhaitez enregistrer sur la carte mémoire MMC (liste déroulante "Rechercher dans").
4. Dans la liste proposée, sélectionnez les fichiers que vous souhaitez enregistrer sur la carte mémoire MMC.
5. Cliquez sur le bouton "Ouvrir".

Marche à suivre pour transférer des fichiers de la carte mémoire MMC dans le système de fichiers

1. Ouvrez la vue "Carte mémoire" dans l'onglet "CPU en ligne" de la fenêtre du projet.
2. Choisissez la commande de menu **Fichier > Transférer les fichiers > Sur la PG**.
3. Dans l'onglet "Carte mémoire" affiché, sélectionnez les fichiers que vous souhaitez transférer sur la PG.
4. Cliquez sur le bouton "Transférer".
5. Dans la boîte de dialogue qui s'ouvre, sélectionnez le dossier dans lequel vous souhaitez transférer les fichiers.
6. Cliquez sur le bouton "OK".

8.5 Echange de données de projet entre STEP 7 Lite et STEP 7

Que peut-on exporter dans STEP 7, que peut-on importer depuis STEP 7 ?

Entre STEP 7 Lite et STEP 7, vous pouvez échanger les données de projet suivantes :

- Le programme complet sous forme de fichier *.awl.
- La table des mnémoniques complète sous forme de fichier *.sdf.

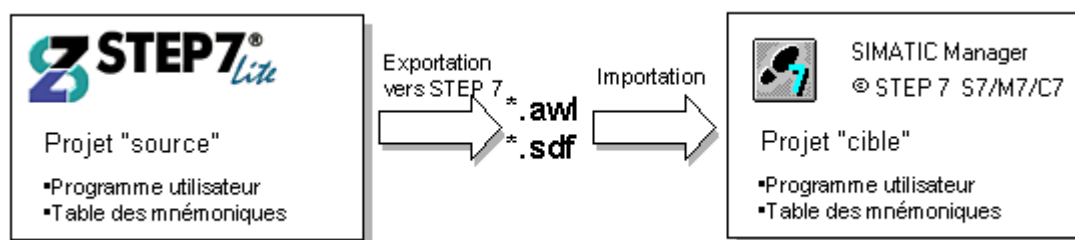
Lors de l'exportation, STEP 7 Lite exporte tous les blocs du programme ou l'ensemble de la table des mnémoniques. Lors de l'importation depuis STEP 7, tous les blocs du fichier *.awl et tous les mnémoniques du fichier *.sdf sont importés.

Conditions requises

Lors de l'exportation et de l'importation d'un programme ou d'une table des mnémoniques, tous les objets dans STEP 7 Lite doivent être fermés.

Marche à suivre lors de l'exportation vers STEP 7

Dans STEP 7 Lite, exportez le programme utilisateur complet sous forme de fichier *.awl ou la table des mnémoniques complète sous forme de fichier *.sdf. Vous devez ensuite importer le fichier *.awl avec la fonction **Insertion > Sources externes** et le fichier *.sdf avec la fonction **Table > Importer** dans le projet STEP 7.



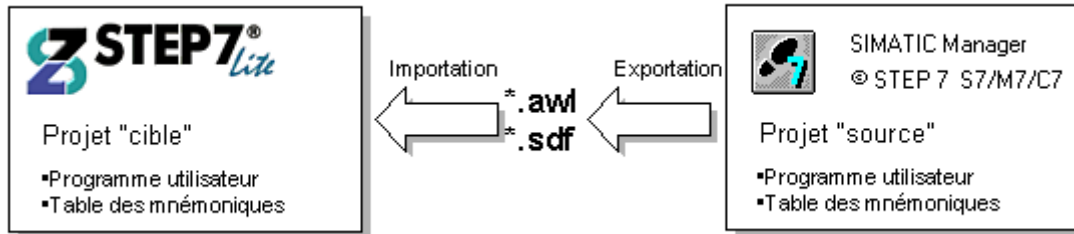
Pour l'exportation vers STEP 7, procédez de la manière suivante :

1. Sélectionnez l'icône "Programme" ou "Table des mnémoniques" dans la fenêtre du projet. (si vous souhaitez exporter les deux éléments, sélectionnez-les simultanément avec la touche CTRL).
2. Choisissez la commande **Fichier > Exporter > Pour STEP 7**.
3. Dans la boîte de dialogue qui s'affiche, sélectionnez le répertoire cible. L'option "Identificateur absolu" ou "Identificateur symbolique" détermine la forme sous laquelle les éléments sélectionnés seront exportés.
4. Sélectionnez le nom des fichiers d'exportation sans extension. Les noms de fichier à générer sont affichés avec indication du chemin dans le champ "Fichiers générés".
5. Cliquez sur le bouton "Exporter".
6. L'importation dans le projet STEP 7 est décrite dans l'aide de STEP 7 (rubriques "Insertion de sources externes" et "Importation d'une table de mnémoniques").

Marche à suivre lors de l'importation depuis STEP 7

Dans STEP 7, exportez le programme utilisateur complet ou des blocs individuels sous forme de fichier *.awl avec la fonction "Exporter la source". La table des mnémonique doit être exportée sous forme de fichier *.sdf avec la fonction "Table > Exporter".

Importez ensuite le fichier *.awl et le fichier *.sdf dans STEP 7 Lite.



Pour l'importation depuis STEP 7, procédez de la manière suivante :

1. Dans STEP 7, exportez les sources et la table des mnémoniques. L'exportation de sources STEP 7 et de la table des mnémoniques STEP 7 sont décrites dans l'aide de STEP 7 (rubriques "Exportation de sources" et "Exportation d'une table des mnémoniques").
2. Choisissez la commande de menu **Fichier > Importer** dans STEP 7 Lite.
3. Dans la boîte de dialogue qui s'ouvre, sélectionnez le répertoire source et les fichiers *.awl et *.sdf correspondants.
4. Avec la case à cocher "Ecraser les objets", indiquez si vous souhaitez que les blocs existants soient écrasés ou pas.
5. Cliquez sur le bouton "Importer" (la boîte de dialogue qui s'ouvre donne la liste d'éventuelles erreurs d'importation).

Nota

Vous pouvez également importer des données de projet depuis STEP 5. Utilisez à cet effet le convertisseur livré "Convertisseur de fichiers S5" ainsi que l'aide en ligne correspondante.

Après la conversion, poursuivez avec l'étape 2 décrite ci-avant. Comme fichiers à importer, sélectionnez les fichiers <Nom>AC.AWL et <Nom>S7.SEQ créés par le convertisseur.

8.6 Exportation de données de projet pour éditeurs externes

8.6.1 Format de données pour l'importation/exportation d'une table des mnémoniques

Vous pouvez importer le format de fichier System Data Format (SDF) dans la table des mnémoniques ou l'en exporter :

- Vous pouvez ouvrir, éditer, puis enregistrer les fichiers SDF avec l'application Microsoft Access.
- Utilisez le format SDF pour importer des données dans l'application Microsoft ACCESS ou pour les en exporter.
- Sélectionnez, dans ACCESS, le format de fichier "Texte (avec séparateurs)".
- Utilisez le guillemet (") comme séparateur de texte.
- Utilisez la virgule (,) comme séparateur de champ.

System Data Format (SDF)

Type de fichier	*.SDF
Structure	Chaînes de caractères entre guillemets, sections séparées par des virgules.
Exemple	"phase_verte_piet","T 2","TIMER","Durée de la phase verte pour piétons" "rouge_piet","A 0.0","BOOL","Rouge pour piétons"

Pour ouvrir un fichier SDF dans Microsoft Access, choisissez le format de fichier "texte (avec séparateur)". Indiquez comme séparateur de texte les guillemets (") et comme séparateur de champ la virgule (,).

8.6.2 Gestion multilingue des textes

STEP 7 Lite permet d'exporter des textes stockés dans un projet en une seule langue, pour les faire traduire, puis les réimporter et les afficher dans la langue de traduction.

Les types de textes suivants autorisent cette gestion multilingue.

- Titres et commentaires :
 - titres et commentaires de station et de module,
 - titres de catégorie,
 - titres de bloc et commentaires de bloc,
 - titres de réseau et commentaires de réseau,
 - commentaires de ligne dans les programmes LIST et les tables de variables,
 - commentaires tirés des tables de mnémoniques, des tables de déclaration de variables, des types de données utilisateur et des blocs de données.
- Textes affichés (pas avec STEP 7 Lite) :
 - textes de message,
 - bibliothèques de textes système.

Exportation

L'exportation est effectuée pour tous les types de texte appartenant à l'objet sélectionné. Un fichier d'exportation est généré pour chaque type de texte. Il contient une colonne pour la langue source et une pour la langue cible. Il est interdit de modifier les textes dans la langue source.

Importation

L'importation consiste à adopter dans le projet sélectionné le contenu des colonnes de la langue cible (colonne droite). Seuls sont adoptés les textes pour lesquels la colonne de la langue source contient un texte conforme à un texte existant dans le projet.

Changement de langue

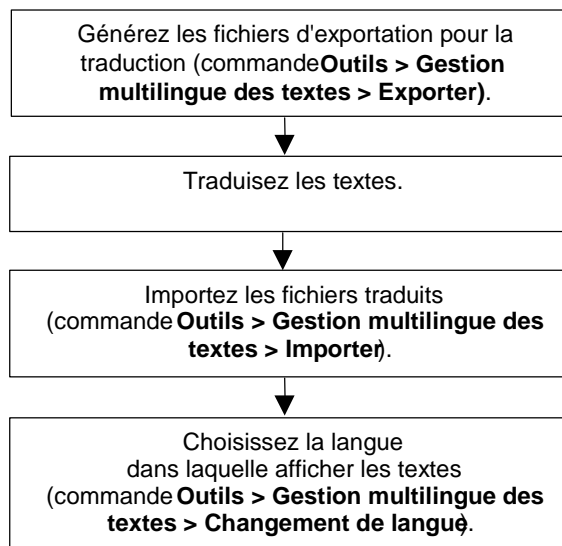
Vous pouvez choisir ici toutes les langues que vous avez indiquées lors de l'importation dans le projet sélectionné.

Effacer langue

Lorsque vous effacez une langue, tous les textes traduits dans cette langue sont effacés de la base de données interne.

Il est recommandé de toujours sélectionner une langue de référence dans votre projet. Il peut s'agir, par exemple, de votre langue nationale. N'effacez jamais cette langue. Lors de l'exportation et de l'importation, indiquez toujours cette langue de référence comme langue source. Choisissez la langue cible que vous souhaitez.

Marche à suivre



8.6.2.1 Types de textes à gestion multilingue

A l'exportation, un fichier est créé pour chaque type de texte. Le nom de ce fichier est celui du type de texte et son extension est le format d'exportation (TypeTexte.Format : par ex. SymbolComment.CSV). Les fichiers ne respectant pas les conventions de nom ne peuvent être utilisés comme source ou comme cible.

Les textes traduisibles au sein d'un projet sont classés en types de texte suivants :

Type de texte	Signification
HardwareTitle	Titre de station et de module
HardwareComment	Commentaire de station et de module
CategoryTitle	Titre de catégorie
VariableComment	Commentaire de ligne dans une table de déclaration de variables
BlockTitle	Titre de bloc
BlockComment	Commentaire de bloc
NetworkTitle	Titre de réseau
NetworkComment	Commentaire de réseau
LineComment	Commentaire de ligne dans LIST
InterfaceComment	Commentaire de Var_Section (table de déclaration dans les blocs de code) Commentaire d'UDT (type de données utilisateur) Commentaire de bloc de données
SymbolComment	Commentaire de mnémonique

8.6.2.2 Structure du fichier d'exportation

Le fichier d'exportation est toujours construit de la manière suivante.

Exemple :

\$ Langues		
7(1) Français (France)	9(1) Anglais (USA)	
\$ Typ(SymbolComment)		
Première chaîne de caractères à traduire	Traduction	
Seconde chaîne de caractères à traduire	Traduction	

Langue source

Langue cible

Voici les règles fondamentales à connaître :

1. Il est interdit de modifier, d'écraser ou d'effacer :
 - les champs précédés d'un "\$_" (ce sont des mots-clés),
 - les numéros indiquant la langue (dans l'exemple ci-dessus, 7(1) pour la langue source Français (France) et 9(1) pour la langue cible Anglais).
2. Un fichier contient toujours des textes du même type. Dans notre exemple, il s'agit du type de texte Commentaire de mnémonique (\$_Typ(SymbolComment)). Les règles pour le traducteur qui édite ce fichier sont rangées dans le texte introductif du fichier d'exportation.
3. Les informations complémentaires sur les textes ou commentaires doivent toujours figurer avant la définition du type (\$_Typ...) ou après la dernière colonne.

Nota

Si la colonne de la langue cible contient seulement "\$_Undefined", c'est qu'il n'en a pas été indiqué lors de l'exportation. Pour plus de clarté, vous pouvez remplacer cette chaîne par la langue cible, "Anglais" par exemple. Dans ce cas, il faudra contrôler la langue cible proposée à l'importation et la choisir de nouveau au besoin.

Format du fichier d'exportation

Les fichiers d'exportation sont stockés en format CSV. Pour les éditer avec Excel, il faut tenir compte du fait qu'EXCEL n'ouvre un fichier CSV correctement que par la commande du menu Fichier. **L'ouverture d'un fichier CSV par double-clic dans l'Explorateur le rend fréquemment inutilisable.** Votre travail avec EXCEL sera plus facile si vous procédez comme suit :

1. Ouvrez les fichiers d'exportation avec EXCEL.
2. Enregistrez les fichiers en tant que fichiers XLS.
3. Traduisez les textes dans les fichiers XLS.
4. Enregistrez les fichiers XLS en format CSV avec EXCEL.

Attention

Il est interdit de renommer les fichiers d'exportation.

8.6.2.3 Détails sur la gestion multilingue des textes

Exportation de textes à gestion multilingue

Condition requise

Aucun fichier d'exportation ne doit être ouvert.

Marche à suivre

1. Choisissez la commande **Outils > Gestion multilingue des textes > Exporter**.
2. Dans la boîte de dialogue "Exporter", déterminez langue source et langue cible ainsi que les types de texte.

Conseil 1 : choisissez habituellement tous les types de texte ; pour les traductions de retouche (par ex. de commentaires modifiés) seulement les types de texte requis.

Conseil 2 : si vous voulez faire traduire le projet en **plusieurs langues**, laissez en blanc la zone de la langue cible. Cette dernière est alors inscrite comme "\$_Undefined" dans le fichier d'exportation. Copiez ce fichier pour plusieurs traducteurs et écrivez chaque fois la langue à traduire à la place du texte "\$_Undefined". Il faudra choisir explicitement la langue cible lorsque vous importerez les fichiers traduits !

3. S'il y a déjà des fichiers d'exportation (cible d'exportation), la boîte de dialogue suivante vous permet d'indiquer s'il faut les compléter ou les remplacer.
Au cas où vous choisissez de compléter les fichiers existants, les textes déjà traduits sont conservés et les nouveaux textes (encore à traduire) y sont ajoutés.
4. Acquitez la boîte de dialogue avec "OK" et envoyez les fichiers de textes créés au traducteur.

Traduction de textes à gestion multilingue

Quand les fichiers d'exportation sont édités avec MS-Excel, il faut les ouvrir au moyen de la commande **Fichier > Ouvrir** d'Excel et non pas par double-clic sur le fichier (format CSV), sinon ils ne seront pas ouverts correctement.

- Traduisez les textes dans la deuxième colonne sous le mot-clé "\$_Typ(...)".

Attention

Sachez qu'EXCEL interprète certains caractères comme des formules et modifie par conséquent la langue source s'il l'enregistre. Dans ce cas, les textes traduits ne sont pas importés.

Importation de textes à gestion multilingue

1. Choisissez la commande **Outils > Gestion multilingue des textes > Importer**.
2. Dans la boîte de dialogue "Importer", déterminez la source de l'importation et le format.
Si vous n'avez pas saisi de langue cible à l'exportation, un dialogue vous invite à la choisir maintenant.
3. Acquitez la boîte de dialogue avec "OK"
4. Si des erreurs surviennent, évaluez le journal qui s'affiche.

Choix de la langue

1. Choisissez la commande Outils > Gestion multilingue des textes > Changement de langue.
2. Dans la boîte de dialogue "Changement de langue", cochez la langue de votre choix pour les types de texte.
3. Acquitez la boîte de dialogue avec "OK".

Suppression d'une langue

1. Choisissez la commande Outils > Gestion multilingue des textes > Effacer une langue.
2. Dans la boîte de dialogue qui s'ouvre, sélectionnez la langue à effacer et précisez si vous souhaitez effacer également les titres et les commentaires.
3. Acquitez la boîte de dialogue avec "OK".

8.6.2.4 Conseils pour la traduction

Optimisation du modèle à traduire

Vous pouvez simplifier le "matériau source" de la traduction en unifiant les termes ou expressions ayant le même sens.

Exemple

Avant la simplification (fichier d'exportation)

\$ _Langues		
7(1) Français (France)	7(1) Français (France)	
\$ _Typ(SymbolComment)		
ValidAuto		
Validation automatique		
Validation auto		

Langue source

Langue cible

Réduction à une même expression :

\$ _Langues		
7(1) Français (France)	7(1) Français (France)	
\$ _Typ(SymbolComment)		
ValidAuto	Validation auto	
Validation automatique	Validation auto	
Validation auto	Validation auto	

Langue source

Langue cible

Après la simplification (c'est-à-dire après importation suivie d'une exportation) :

\$ _Langues		
7(1) Français (France)	7(1) Français (France)	
\$ _Typ(SymbolComment)		
Validation auto	Validation auto	

Langue source

Langue cible

Optimisation de la traduction

Pour les projets dont la structure et les textes sont analogues à ceux d'un projet précédent, vous pouvez optimiser l'opération de traduction.

Le procédé décrit ci-après est recommandé en particulier pour les projets que vous avez créés par copie et modification ultérieure.

Condition requise

Il y a déjà une cible d'exportation (fichiers CSV).

Marche à suivre

1. Copiez les fichiers d'exportation dans le répertoire du nouveau projet à traduire.
2. Ouvrez le nouveau projet et exportez les textes (commande **Outils > Gestion multilingue des textes > Exporter**).
La cible d'exportation existant déjà, un dialogue vous demande s'il faut la compléter ou la remplacer.
3. Cliquez sur le bouton "Compléter".
4. Faites traduire les fichiers d'exportation (seuls les nouveaux textes sont à traduire).
5. Importez pour finir les fichiers traduits.

9 Chargement dans la CPU et la PG

9.1.1 Conditions requises pour le chargement

Conditions requises pour le chargement dans la CPU

- Une liaison est établie entre votre PG et la CPU du système cible via l'interface MPI.
- Il est possible d'accéder à la CPU.
- La compilation du programme à charger s'est faite sans erreur.
- La CPU doit se trouver dans un état de fonctionnement autorisant le chargement (arrêt ou marche RUN-P).
A l'état de marche (RUN-P), veillez à transférer le programme bloc par bloc. En effet, des conflits risquent de survenir si vous écrasez un ancien programme CPU, par exemple quand des paramètres de bloc ont été modifiés. La CPU passerait alors à l'état d'arrêt durant l'exécution du cycle. C'est pourquoi il est recommandé de mettre la CPU à l'état d'arrêt avant de réaliser le chargement.
- Nous vous recommandons d'effectuer un effacement général de la CPU avant de charger votre programme utilisateur, afin d'être sûr qu'il n'y a plus d'anciens blocs dans la CPU.

Etat de fonctionnement "Arrêt" (STOP)

Faites passer la CPU de l'état "Marche" (RUN) à l'état "Arrêt" (STOP) avant :

- de charger le programme utilisateur complet ou certaines de ses parties dans la CPU,
- d'effectuer un effacement général de la CPU,
- de comprimer la mémoire utilisateur.

Démarrage à chaud (passage à l'état de fonctionnement "Marche")

Lorsque vous commandez un démarrage à chaud à partir de l'état "Arrêt" (STOP), le programme recommence au début et c'est d'abord le programme de mise en route (OB100) qui est exécuté à l'état de fonctionnement "Mise en route". Si la mise en route s'achève sans erreur, la CPU passe à l'état "Marche" (RUN). Un démarrage à chaud est nécessaire après :

- l'effacement général de la CPU,
- le chargement du programme utilisateur à l'état "Arrêt" (STOP).

9.1.2 Charger quoi et quand dans la CPU ?

Le menu "Fichier" propose la commande "Charger dans CPU". Le tableau ci-après dresse la liste des objets que vous pouvez charger et dans quelles conditions.

Charger quoi ?	Condition	Observations
Configuration matérielle	La configuration matérielle est momentanément ouverte Ou bien l'objet 'Matériel' est sélectionné.	Vous pouvez sélectionner l'objet 'Matériel' <ul style="list-style-type: none"> • dans la vue d'ensemble du projet, • dans la vue 'Détails' (après double-clic sur l'icône 'Projet' dans la vue d'ensemble du projet).
Blocs	Le bloc est momentanément ouvert dans l'éditeur de bloc ou bien un ou plusieurs blocs sont sélectionnés.	Vous pouvez sélectionner des blocs <ul style="list-style-type: none"> • dans la vue d'ensemble du projet, • dans la vue 'Détails' (après double-clic sur l'icône 'Projet' ou sur l'icône 'Programme' dans la vue d'ensemble du projet).
Un ou plusieurs objets quelconques (indépendamment de la vue affichée)	L'objet est sélectionné dans la vue d'ensemble du projet.	Si vous avez sélectionné le projet, tous les blocs et la configuration matérielle seront chargés. Si vous avez sélectionné le programme, tous les blocs du programme seront chargés.

S'il y a déjà des blocs aux numéros identiques dans la CPU, la boîte de dialogue qui s'ouvre vous permet de préciser s'il faut les écraser ou pas.

Quand l'objet à charger (un bloc, par exemple, ou la configuration matérielle) est ouvert dans STEP 7 Lite, c'est cet état actuellement visible qui est chargé (et non pas les blocs enregistrés ou la configuration matérielle enregistrée) ! Pour des raisons de cohérence, n'oubliez pas d'enregistrer au préalable l'état que vous souhaitez charger !

9.1.3 Différence entre l'enregistrement et le chargement de blocs

Remarque sur les modifications apportées à un bloc : enregistrer avant de charger

Pour enregistrer un bloc nouvellement créé ou les modifications apportées à la section des instructions d'un bloc de code, aux tables de déclaration ou aux valeurs dans un bloc de données, vous devez enregistrer le bloc concerné (commande de menu **Fichier > Enregistrer**). Vous enregistrez ainsi l'ensemble du projet.

Il est indispensable d'enregistrer également sur le disque dur de la PG, avant de quitter l'éditeur, les modifications que vous avez faites et transférées directement dans la CPU avec la commande **Fichier > Charger** - par exemple, parce qu'elles étaient minimes et que vous vouliez les tester immédiatement. Vous aurez sinon des versions différentes du programme utilisateur dans la CPU et dans la console de programmation.

En général, il est conseillé d'enregistrer d'abord les modifications et de ne les charger qu'ensuite.

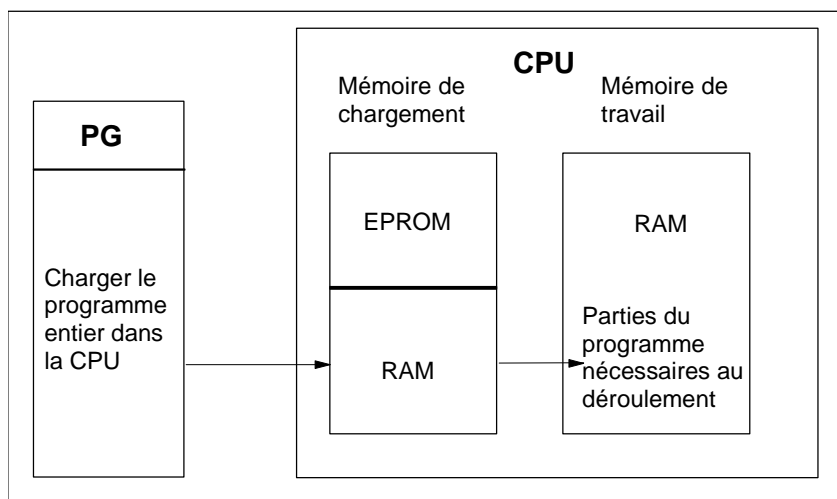
9.1.4 Mémoire de chargement et mémoire de travail dans la CPU

Une fois terminés la configuration, le paramétrage et l'écriture du programme, vous pouvez transférer des programmes utilisateur entiers ou des blocs isolés dans la CPU connectée. Pour tester des blocs isolés, il faut charger au moins un OB ainsi que les FB et les FC qui y sont appelés et les DB utilisés.

Vous devez transférer également la configuration matérielle dans le système cible.

Interaction de la mémoire de chargement et de la mémoire de travail de la CPU

Le programme utilisateur entier est chargé dans la mémoire de chargement ; les parties du programme nécessaires à l'exécution sont chargées aussi dans la mémoire de travail.



Mémoire de chargement de la CPU

- La mémoire de chargement sert à recevoir le programme utilisateur sans table des mnémoniques ni commentaires (qui restent dans la zone de mémoire de la PG).
- Les blocs repérés comme non nécessaires à l'exécution sont admis exclusivement dans la mémoire de chargement.
- Suivant la CPU, la mémoire de chargement peut être une mémoire RAM, ROM ou EPROM.
- Dans les appareils S7-300, la mémoire de chargement peut comporter aussi une partie EEPROM intégrée en plus de la partie RAM intégrée (par ex. la CPU312 IFM et la CPU314 IFM).

Mémoire de travail de la CPU

La mémoire de travail (RAM intégrée) sert à recevoir les parties du programme utilisateur nécessaires à son exécution.

Sortes de chargement possibles

La fonction de chargement vous permet de charger le programme utilisateur ou les objets chargeables (par ex. des blocs) dans le système cible. Lorsqu'un bloc se trouve déjà dans la RAM de la CPU, un message vous demande s'il faut l'écraser ou pas.

- Vous pouvez sélectionner les objets chargeables dans la fenêtre du projet, puis exécuter la commande **Fichier > Charger dans la CPU**.

Inversement, la fonction de chargement peut vous servir à charger dans votre PG les contenus actuels de blocs se trouvant dans la mémoire RAM de chargement de la CPU.

9.1.5 Possibilités de chargement selon la mémoire de chargement

La structure de la mémoire de chargement formée d'une partie RAM et d'une partie EPROM n'est pas sans conséquence sur les possibilités de chargement de votre programme utilisateur ou de blocs individuels. Le chargement des données dans la CPU peut s'effectuer de plusieurs manières :

Mémoire de chargement	Possibilités de chargement
Mémoire vive (RAM)	Chargement et effacement de blocs individuels
	Chargement et effacement d'un programme utilisateur entier
	Chargement a posteriori de blocs individuels
EPROM intégrée ou enfichable	Chargement de programmes utilisateur entiers
EPROM enfichable	Chargement de programmes utilisateur entiers

Chargement dans la RAM via une liaison en ligne

En cas de panne secteur, les données ne sont pas protégées dans la CPU si la mémoire vive n'est pas sauvegardée. Dans ce cas, les données se trouvant dans la RAM sont perdues.

Enregistrement sur une carte mémoire EPROM

Les blocs ou le programme utilisateur sont enregistrés sur une carte mémoire EPROM qui doit se trouver dans le logement prévu à cet effet sur la CPU.

Les données enregistrées sont conservées en cas de coupure de courant ou d'effacement général de la CPU. Le contenu de l'EPROM est à nouveau copié dans la zone RAM de la mémoire de la CPU après effacement général de la CPU et après retour du courant suite à une panne secteur avec mémoire vive non sauvegardée.

Enregistrement dans l'EPROM intégrée

Pour la CPU 312, il y a encore une possibilité : c'est d'enregistrer le contenu de la mémoire vive dans l'EPROM intégrée où les données sont rémanentes en cas de panne secteur. Le contenu de l'EPROM intégré est à nouveau copié dans la zone RAM de la mémoire de la CPU après effacement général de la CPU et après retour du courant suite à une panne secteur avec mémoire vive non sauvegardée.

9.1.6 Chargement de blocs et de la configuration dans la CPU et enregistrement sur Memory Card

9.1.6.1 Chargement a posteriori de blocs dans la CPU

Vous pouvez écraser les blocs présents dans la mémoire de chargement (mémoire vive) ou dans la mémoire de travail de la CPU avec une nouvelle version (chargement a posteriori). L'ancienne version des blocs est alors écrasée.

Le chargement de blocs a posteriori s'effectue comme le chargement, à ceci près que le système vous demandera si vous voulez écraser les blocs existants.

Un bloc enregistré dans une EPROM ne peut être écrasé, mais sera déclaré non valable après le chargement a posteriori. Le bloc qui le remplace sera chargé dans la mémoire vive. Il en résulte des intervalles dans la mémoire de chargement ou dans la mémoire de travail, susceptibles d'empêcher le chargement de nouveaux blocs. Dans ce cas, il est recommandé de compresser les mémoires.

Attention

En cas de retour de courant après une panne secteur avec une mémoire vive non sauvegardée, ou en cas d'effacement général de la CPU, les "anciens blocs" de l'EPROM redeviennent valables et sont chargés dans la CPU !

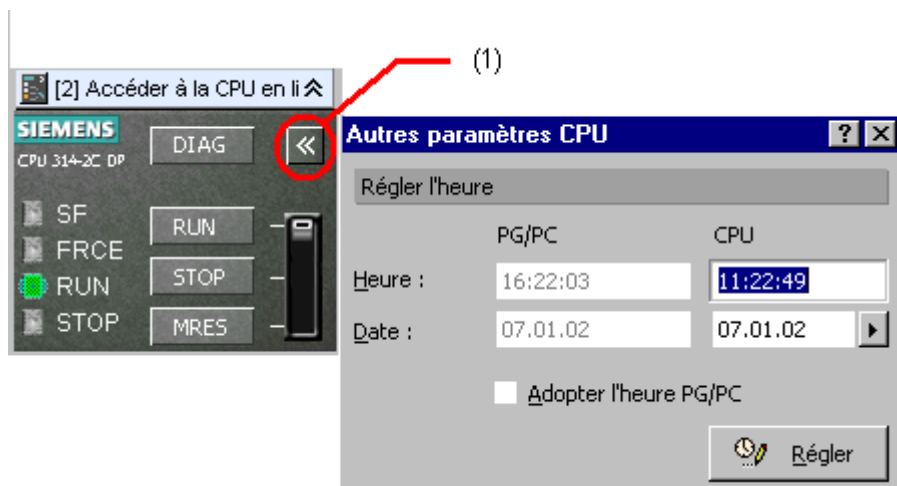
9.1.6.2 Enregistrer des blocs chargés dans l'EPROM intégrée ou dans la carte mémoire de la CPU

Avec les modules CPU disposant d'une EPROM intégrée (par ex. la CPU 314-2C DP), vous pouvez copier des blocs de la mémoire de chargement (RAM) dans l'EPROM intégrée afin de ne pas perdre les données en cas de coupure de courant ou d'effacement général.

De même, quand une carte mémoire S7 (5V-FEPROM) est enfichée dans la CPU, vous pouvez transférer les blocs de la mémoire de chargement (RAM) sur la carte mémoire S7.

Procédez de la manière suivante :

1. Chargez les blocs dans la CPU.
2. Elargissez le panneau de commande CPU dans l'interface utilisateur de STEP 7 Lite.



(1) : cliquez pour élargir/réduire le panneau de commande de la CPU

3. Dans le panneau élargi, cliquez sur le bouton "RAM > ROM" (sous "Autres fonctions").

9.1.6.3 Chargement d'une configuration dans un système cible

Astuce

Avant de charger votre configuration de station, il est bon d'en vérifier la correction à l'aide de la commande **Outils > Vérifier la cohérence**, la configuration matérielle étant ouverte. STEP 7 Lite vérifie dans ce cas que la configuration actuelle permet bien de générer des données système chargeables. La liste des erreurs trouvées s'affiche.

Conditions préalables au chargement

- La console de programmation est connectée à l'interface MPI de la CPU au moyen d'un câble MPI.
- Si l'installation est en réseau (console de programmation connectée à un sous-réseau) : tous les modules d'un sous-réseau ont impérativement des adresses de réseau différentes.
- La configuration réalisée correspond à la configuration réelle de la station. Une configuration ne peut être chargée dans la station que si elle est cohérente et exempte d'erreurs. C'est la condition pour que soient générés les blocs de données système (SDBs) qui sont chargés dans les modules.

Marche à suivre

1. Dans la fenêtre du projet, sélectionnez l'entrée "Matériel".
2. Choisissez la commande **Fichier > Charger dans CPU** ou appuyez sur le bouton droit de la souris et choisissez dans le menu contextuel la commande **Charger dans CPU**

La configuration du système d'automatisation entier est alors chargée dans la CPU. Les paramètres de CPU entrent en vigueur immédiatement, ceux des autres modules leur sont transférés à la mise en route.

Avertissement

Il n'est pas possible de charger dans une station des configurations partielles, par ex. celles de certains profilés support. Pour des raisons de cohérence, STEP 7 Lite charge toujours la configuration complète dans la CPU.

Modifier l'état de fonctionnement de la CPU lors du chargement

Lorsque vous lancez la fonction **Fichier > Charger dans CPU**, vous pouvez exécuter les actions suivantes de manière interactive depuis la PG :

- Mettre la CPU en STOP
(si le commutateur de mode est en position RUN-P ou si la liaison à la CPU a été autorisée par mot de passe).
- Comprimer la mémoire
(s'il n'y a pas assez de mémoire libre continue).
- Remettre la CPU en RUN.

9.2 Chargement depuis la CPU dans la PG

Cette fonction vous permet de réaliser les tâches suivantes :

- mémoriser des informations provenant de la CPU (par ex. à des fins de maintenance),
- configurer et éditer rapidement une station lorsque les composants matériels sont disponibles au début du travail de configuration.

Mémorisation d'informations provenant du système cible

Cette mesure peut s'avérer nécessaire, par exemple quand vous ne disposez pas (ou seulement en partie) des données de projet hors ligne correspondant à l'état en cours dans la CPU. Vous pouvez alors charger dans votre PG au moins la partie des données de projet qui est disponible en ligne.

Configuration rapide

Pour saisir aisément la composition de la station, chargez les données de configuration depuis le système cible dans votre PG après avoir monté le matériel et effectué un démarrage à chaud de la station. Vous obtenez ainsi la composition de la station avec les indications de type des différents modules. Vous n'aurez plus ensuite qu'à remplacer les modules par d'autres tirés du catalogue et à les paramétrer.

Les informations chargées dans la PG concernent la configuration du profilé support de base ("Rack 0") et des profilés support d'extension éventuels.

Avertissement

Pour le chargement dans la PG (en l'absence d'une configuration hors ligne), STEP 7 Lite n'est pas en mesure de fournir tous les numéros de référence des composants dans leur intégralité.

Vous pourrez remplacer les modules aux numéros de référence "incomplets" par les modules appropriés tirés du catalogue lorsque vous configurerez le matériel.

9.2.1 Recharger quoi et quand dans la PG ?

Le menu "Fichier" contient la commande "Charger dans PG" qui permet de recharger des données de la CPU dans le projet hors ligne se trouvant dans la PG/le PC.

Cette commande recharge dans la PG/le PC les objets que vous avez sélectionnés dans la fenêtre de projet ou dans la vue "CPU en ligne". Si le projet hors ligne contient déjà des blocs portant des numéros identiques, une boîte de dialogue vous permettra d'indiquer s'il faut les écraser ou pas.

Ce qu'il n'est pas possible de recharger

Lorsqu'un programme est chargé de la PG dans la CPU, tout ce qui est enregistré dans la base de données du projet n'est pas chargé dans la CPU. Par conséquent, ces données manquent quand vous rechargez des blocs, par exemple.

Pour les données chargées de la CPU dans la PG, il faut savoir que :

- Les blocs ne contiennent pas de mnémoniques pour les paramètres formels, les variables temporaires et les repères de saut. STEP 7 Lite génère des noms comme IN0, STAT1, M001 pour remplacer ces mnémoniques. Les commentaires de la table de déclaration des variables manquent également.
- Des structures sont affichées et employées à la place des types de données utilisateur (UDT). La modification d'un UDT déjà employé n'est pas prise en considération.
- Les blocs ne contiennent pas de commentaire. Toutes les traductions du projet manquent.
- Les blocs LOG ou CONT ne contiennent ni commentaire ni titre de réseau, de même ni commentaire ni titre de bloc.
- Les blocs LIST ne contiennent pas de commentaires de ligne.
- Les informations nécessaires à la fonction "Mise à jour de l'interface dans les appels de bloc" manquent. Par conséquent, quand le bloc appelé n'existe pas hors ligne ou qu'il existe avec une interface modifiée, le code du réseau présentant un conflit avec le bloc appelé sera représenté en LIST étendu (Disassembly=MC7), quel que soit le langage de création du bloc.
- Les informations nécessaires à la fonction "Classement symbolique" manquent. Les blocs rechargés dans la PG obéissent toujours au "classement absolu", quelle que soit l'option en vigueur dans le projet.
- Les références croisées manquent. Solution : enregistrer de nouveau le bloc dans l'éditeur de programme.
- Si la CPU a été chargée avec le logiciel de base STEP 7, il n'est pas possible de poursuivre le traitement des données concernant la "communication par données globales (GD)", la "configuration de messages sur mnémonique", la "configuration de réseaux" et la "périphérie décentralisée".
- Tous les commentaires et les formats manquent dans les tâches de forçage permanent rechargées dans la PG.
- Les commentaires figurant dans les dialogues des modules ne sont pas rechargés.

Si vous rechargez un projet "vide", il manquera en plus :

- la table des mnémoniques avec les noms symboliques des opérandes et les commentaires,
- les types de données utilisateur (UDT).

Sélection des objets à recharger

Seuls des éléments se trouvant réellement dans la CPU peuvent être rechargés dans la PG. En d'autres termes, quand vous sélectionnez des éléments qui sont caractérisés comme "hors ligne seulement", la commande "Charger dans PG" n'est pas disponible.

Sélectionné dans la vue d'ensemble du projet (CPU en ligne)	Chargé dans la PG	Observations
Le projet	Tout ce qui peut l'être	Configuration matérielle, tous les blocs
La configuration matérielle	La configuration matérielle	Valable dans toutes les vues de la configuration HW (comparaison HW, diagnostic HW)
Le programme	Tous les blocs du programme utilisateur	-
Un ou plusieurs blocs	Les blocs sélectionnés	-

9.2.2 Chargement d'objets de la CPU dans la PG/le PC

1. Ouvrez le projet dans lequel vous souhaitez charger les objets en question (par ex. des blocs).
2. Sélectionnez dans la fenêtre du projet, onglet "CPU en ligne", l'objet ou le choix d'objets à recharger dans la PG ou le PC.
Il peut s'agir du matériel, du programme (avec tous les blocs) ou de certains blocs seulement.
3. Choisissez la commande **Fichier > Charger dans PG**.

Les objets sélectionnés sont alors transférés dans la base de données du projet dans la PG ou le PC.

9.2.3 Edition de blocs chargés dans la PG/le PC

Vous pouvez utiliser la fonction de chargement (commande **Fichier > Charger dans PG**) pour charger dans votre PG les contenus actuels de blocs se trouvant dans la RAM de chargement de la CPU.

Attention !

Conflit d'horodatage en cas de traitement en ligne et hors ligne !

Les procédés décrits provoquent des conflits d'horodatage et il est donc préférable de les éviter.

Des conflits d'horodatage se produisent lorsque vous ouvrez un bloc dans la "vue en ligne de la CPU" alors que des modifications effectuées hors ligne n'ont pas été chargées dans la CPU.

Des conflits d'horodatage se produisent en outre lorsque vous ouvrez un bloc dans la vue du projet (c'est-à-dire hors ligne) alors que le bloc en ligne avec conflit d'horodatage a été copié hors ligne dans le programme utilisateur, puis le bloc ouvert hors ligne.

9.2.4 Edition d'une configuration matérielle chargée dans votre PG/PC

Charger une configuration matérielle depuis la CPU dans un nouveau projet

Lorsque vous chargez une configuration matérielle depuis la CPU dans la PG ou le PC alors que ces derniers ne contiennent pas de données du projet, STEP 7 Lite n'est pas en mesure de déterminer complètement le numéro de référence exact du module et par conséquent de ses propriétés.

Ces modules sont représentés par des points d'interrogation dans la table de configuration.

Vous pouvez spécifier les modules de la manière suivante :

1. Sélectionnez le module dans la table de configuration
2. Choisissez l'onglet "Compatible" dans le catalogue du matériel.
Vous y trouverez la liste de tous les modules (compatibles) pouvant remplacer le module sélectionné.
3. Remplacez, par glisser-lâcher, le module dans la table de configuration par le module compatible effectivement utilisé.

9.3 Effacement dans la CPU

9.3.1 Effacement de la mémoire de chargement/travail et effacement général de la CPU

Avant de charger votre programme utilisateur dans la CPU, nous vous recommandons de la soumettre à un effacement général afin d'être sûr qu'elle ne contient plus "d'anciens" blocs.

Condition requise pour l'effacement général

Pour l'effacement général, il faut que la CPU soit à l'arrêt (commutateur de mode de fonctionnement positionné sur STOP ou bien sur RUN-P avec passage à STOP à l'aide du panneau de commande CPU).

Effacement général de CPU

Voici ce qui se déroule lors de l'effacement général d'une CPU :

- La CPU est remise à zéro.
- Toutes les données utilisateur sont effacées (les blocs et les blocs de données système (SDB) à l'exception des paramètres MPI).
- La CPU suspend toutes les liaisons en cours.
- S'il existe des données dans une EPROM (carte mémoire ou EPROM intégrée), la CPU en copie le contenu, après l'effacement général, dans la zone RAM de la mémoire.

Le contenu de la mémoire tampon de diagnostic, du compteur d'heures de fonctionnement et les paramètres de l'interface MPI sont conservés.

Effacement général avec STEP 7 Lite

1. Mettez la CPU à l'état de fonctionnement "Arrêt" de la manière suivante :
 - Positionnez le commutateur de mode de fonctionnement sur STOP.
 - Si ce commutateur est positionné sur RUN-P (RUN pour la CPU 31xC), vous pouvez aussi passer à l'état de fonctionnement "Arrêt" dans le panneau de commande CPU.
 - Cliquez sur le bouton MRES dans le panneau de commande.
Vous pouvez également choisir la commande de menu **Outils > Effacement général**.
2. Confirmez l'effacement général dans la boîte de dialogue qui s'affiche.

9.3.2 Suppression de certains blocs dans la CPU

Durant la phase de test du programme de la CPU, il peut s'avérer nécessaire d'effacer certains blocs dans la CPU. Les blocs sont enregistrés dans la mémoire utilisateur de la CPU soit dans l'EPROM, soit dans la RAM (en fonction de la CPU et de la procédure de chargement).

- Vous pouvez effacer directement les blocs chargés dans la mémoire vive. L'espace mémoire qui était occupé dans les mémoires de chargement et de travail est alors libéré.
- Les blocs enregistrés dans l'EPROM intégrée sont toujours copiés dans la zone de mémoire vive après effacement de la CPU. Vous pouvez effacer directement ces copies dans la mémoire vive. Les blocs effacés seront alors déclarés non valables dans l'EPROM jusqu'au prochain effacement général ou jusqu'à la prochaine panne secteur, lorsque la mémoire vive n'est pas sauvegardée. En cas d'effacement général ou de panne secteur lorsque la mémoire vive n'est pas sauvegardée, les blocs "effacés" sont à nouveau copiés de l'EPROM dans la mémoire vive, où ils sont alors à nouveau actifs. Les blocs enregistrés dans l'EPROM intégrée (par exemple de la CPU 312) sont effacés par écrasement par le nouveau contenu de la mémoire vive.

Suppression dans la RAM de la CPU

Vous pouvez supprimer des blocs isolés ou plusieurs blocs à l'état de fonctionnement Arrêt (STOP) ou RUN-P. Si vous supprimez à l'état RUN-P un bloc qui est encore appelé, la CPU passera en STOP ou un OB d'erreur sera appelé.

Procédez comme suit :

1. Sélectionnez les blocs à supprimer dans la fenêtre "CPU en ligne".
2. Choisissez la commande **Edition > Effacer** ou appuyez sur la touche SUPPR.

Pour effacer de la CPU l'ensemble du programme utilisateur, vous pouvez aussi effectuer un effacement général.

Suppression dans l'EPROM intégrée

Pour effacer l'EPROM intégrée de la CPU 312, il faut l'écraser à nouveau avec le contenu actuel de la RAM dans laquelle tous les blocs utilisateur ont été supprimés.

Procédez comme suit :

1. Effacez le programme utilisateur de la RAM de la CPU comme il est dit ci-dessus.
2. Ouvrez le panneau de commande étendu de la CPU.
3. Dans ce panneau, cliquez sur le bouton "RAM > ROM".

9.3.3 Effacement de la carte mémoire dans la CPU

Selon la CPU utilisée, c'est l'ensemble du programme utilisateur ou également des blocs individuels qui peuvent être effacés de la carte mémoire.

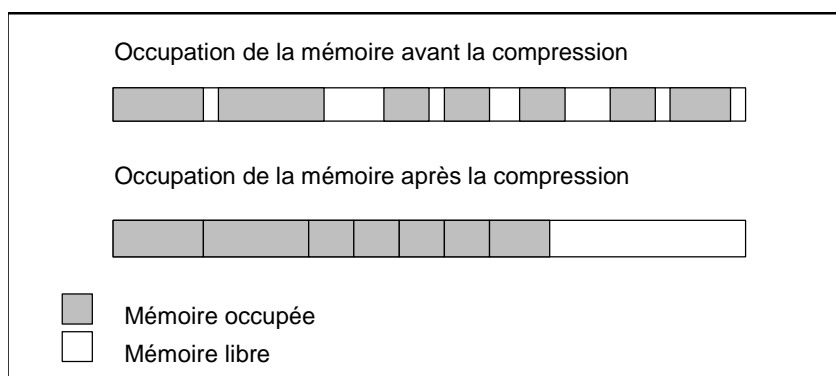
1. Ouvrez la vue "CPU en ligne" de la fenêtre du projet.
2. Cliquez deux fois sur l'icône de carte mémoire.
3. Sélectionnez le programme utilisateur (ou certaines parties seulement) et choisissez la commande **Edition > Effacer** ou appuyez sur la touche SUPPR.

9.4 Compression de la mémoire utilisateur (RAM)

9.4.1 Intervalles dans la mémoire utilisateur (RAM)

Les intervalles résultant des effacements et chargements successifs de blocs réduisent l'espace mémoire utilisable. Il importe de comprimer la mémoire utilisateur en réorganisant les blocs existants afin de créer une zone de mémoire libre d'un seul tenant.

La figure ci-après montre schématiquement comment les blocs de mémoire occupés sont déplacés par la fonction "Compression de la mémoire".



La compression à l'état de fonctionnement "Arrêt" est recommandée.

Seule la compression à l'état de fonctionnement "Arrêt" (STOP) permet d'éliminer tous les intervalles en mémoire. Si vous effectuez la compression à l'état de fonctionnement RUN-P (position du commutateur de mode de fonctionnement), les blocs en cours d'édition ne seront pas déplacés, puisqu'ils sont ouverts. La fonction de compression ne peut pas être exécutée à l'état de fonctionnement RUN (position du commutateur de mode de fonctionnement) (protection en écriture !).

9.4.2 Compression du contenu de la mémoire d'une CPU

Possibilités de compression

Vous avez deux possibilités pour comprimer la mémoire utilisateur.

- Si un manque de mémoire apparaît dans le système cible lors du chargement, une boîte de dialogue vous signalant l'incident s'affiche. Vous pouvez comprimer la mémoire en cliquant sur le bouton correspondant dans cette boîte de dialogue.
- En guise de mesure préventive, vous pouvez afficher l'occupation de la mémoire :
 - Ouvrez le matériel dans la vue "CPU en ligne".
 - Cliquez sur l'onglet "Diagnostic du matériel" et sélectionnez-y une CPU.
 - Choisissez la commande **Outils > Etat du module**.
 - Dans la boîte de dialogue qui s'affiche, cliquez sur l'onglet "Mémoire". Vous y trouvez un bouton pour comprimer la mémoire quand la CPU assure cette fonction.

10 Test

10.1 Présentation des différents modes de test

STEP 7 Lite vous permet de tester l'exécution de votre programme sur le système cible. Il faut pour cela que le programme exécutable soit chargé dans le système cible. Vous pouvez alors visualiser des états de signaux et des valeurs de variables ou bien encore forcer des variables afin de simuler des situations données pour l'exécution du programme.

Vous disposez à cet effet des modes de test suivants :

- **Test avec la table des variables**
Application : simulation de diverses situations.
- **Test avec la visualisation d'état du programme**
Application : suivi de l'exécution du programme, étape par étape
- **Test avec le programme de simulation**
Application : test en l'absence d'un système cible

Pour le test avec le programme de simulation, le logiciel optionnel S7-PLCSIM doit être installé.

10.2 Test avec des tables de variables et des tables de forçage

10.2.1 Introduction au test avec des tables de variables et des tables de forçage

Les tables de variables et de forçage permettent d'enregistrer des environnements de test différents et donc de reproduire sans peine les tests et les observations au cours d'une mise en service ou à des fins de maintenance.

Pour effectuer le test avec des tables de variables, vous disposez des fonctions suivantes :

- **Visualisation de variables**
Cette fonction vous permet d'afficher sur la PG ou le PC les valeurs en cours de certaines variables d'un programme utilisateur ou d'une CPU.
- **Forçage de variables**
Cette fonction vous permet d'attribuer des valeurs fixes à certaines variables d'un programme utilisateur ou d'une CPU. Le forçage est également possible lors du test avec l'état du programme.
- **Débloquer PA et Forçage immédiat**
Ces deux fonctions vous permettent d'attribuer des valeurs fixes à certaines sorties de périphérie d'une CPU à l'état d'arrêt.

- **Forçage permanent de variables**

Cette fonction vous permet d'attribuer à certaines variables d'un programme utilisateur ou d'une CPU une valeur fixe que le programme utilisateur ne peut pas écraser.

Vous pouvez visualiser ou forcer les variables suivantes :

- entrées, sorties, mémentos, temporisations et compteurs,
- contenus de blocs de données,
- périphérie.

10.2.2 Marche à suivre pour visualiser et forcer avec la table des variables

Pour utiliser les fonctions de **visualisation** et de **forçage**, procédez de la manière suivante :

4. Cliquez deux fois sur l'icône "Visualisation/Forçage" dans la fenêtre du projet.
La zone de travail affiche alors la vue d'une table de variables dans laquelle vous pouvez visualiser et forcer des opérandes.
5. Saisissez les opérandes à visualiser ou à forcer dans la table de variables ou sélectionnez une table créée auparavant dans la zone "Tables de variables".
6. Démarrez vos tests avec "Visualiser" ou "Forcer".
7. Enregistrez les modifications apportées à la table de variables en choisissant la commande **Fichier > Enregistrer**.

10.2.3 Marche à suivre de principe pour la visualisation et le forçage avec des tables de forçage

Pour utiliser les fonctions de visualisation/forçage, procédez de la manière suivante :

1. Effectuez un double clic sur l'icône "Visualisation/forçage" dans la fenêtre du projet.
La vue d'une table de forçage s'affiche dans la fenêtre de travail ; vous pouvez y visualiser et y forcer des variables.
2. Entrez les opérandes à visualiser ou à forcer dans la table de forçage ou sélectionnez une table de forçage existante sous "Table de forçage".
3. Démarrez vos tests en cliquant sur "Visualiser" ou "Forcer les valeurs".
4. Enregistrez les modifications dans la table de forçage en choisissant la commande de menu **Fichier > Enregistrer**.

10.2.4 Edition et enregistrement de tables de variables et des tables de forçage

10.2.4.1 Création et ouverture d'une table de variables

Les variables à visualiser ou à forcer sont mémorisées dans des tables de variables. Une fois créée, vous pouvez enregistrer, dupliquer et imprimer la table et vous en resservir quand bon vous semble pour la visualisation et le forçage.

Marche à suivre pour créer et ouvrir une nouvelle table de variables

1. Dans la vue "Visualisation/Forçage", cliquez sur le bouton "Gérer les tables".
2. Dans la boîte de dialogue qui s'ouvre, cliquez sur le bouton "Nouvelle" et tapez un nom pour la nouvelle table.
3. Dans la boîte de dialogue de gestion des tables de variables, cliquez sur le bouton "Afficher" pour afficher la nouvelle table de variables créée.

Marche à suivre pour ouvrir une table de variables

Pour ouvrir une table des variables déjà créée, sélectionnez-la dans la zone "Tables de variables" de la vue "Visualisation/Forçage".

10.2.4.2 Créer et ouvrir une table de forçage

Les variables à visualiser ou à forcer sont enregistrées dans des tables de forçage. Après avoir créé une table de forçage, vous pouvez l'enregistrer, la dupliquer, l'imprimer et toujours l'utiliser à nouveau pour visualiser et forcer des variables.

Marche à suivre pour créer et ouvrir une nouvelle table de forçage

Dans la vue "Visualisation/forçage", cliquez sur le bouton "Gérer les tables".

Dans la boîte de dialogue de gestion des tables de forçage, cliquez sur le bouton "Nouvelle" et entrez un nom pour la nouvelle table de forçage.

Dans la boîte de dialogue de gestion des tables de forçage, cliquez sur le bouton "Afficher" pour afficher la nouvelle table de forçage créée.

Marche à suivre pour ouvrir une table de forçage

Vous pouvez ouvrir une table de forçage déjà créée en la sélectionnant dans le champ "Table de forçage" de la vue "Visualisation/forçage".

10.2.4.3 Copier / Dupliquer une table de variables

Si vous souhaitez utiliser une table de variables comme modèle pour une autre table à créer, vous en obtiendrez un duplicata en procédant comme suit :

1. Dans la vue "Visualisation/Forçage", cliquez sur le bouton "Gérer les tables".
2. Dans la zone présentant les tables de variables disponibles, sélectionnez celle dont vous voulez établir une copie.
3. Cliquez sur le bouton "Dupliquer".
4. Donnez un autre nom au duplicata.
5. Cliquez sur "Appliquer" et quittez la boîte de dialogue avec "Fermer".
La nouvelle table de variables présente la même structure que l'originale et peut être éditée.

Reprendre des tables de variables provenant d'un projet existant

Si vous souhaitez utiliser une table existante pour un nouveau projet, procédez de la manière suivante :

1. Dans le projet cible, sélectionnez la vue "Visualisation/Forçage".
2. Toujours dans le projet cible, sélectionnez la table de variables dans laquelle vous voulez insérer le contenu de la source.
3. Démarrez STEP 7 Lite une seconde fois et ouvrez le projet source.
4. Dans le projet source, sélectionnez la vue "Visualisation/Forçage".
5. Toujours dans le projet source, sélectionnez la table de variables à insérer en tant que source.
6. Sélectionnez dans cette table la plage que vous souhaitez reprendre (copier).
7. Choisissez la commande **Edition > Copier**.
8. Passez dans le projet cible et positionnez le pointeur dans la table de variables du nouveau projet.
9. Choisissez la commande **Edition > Coller**.

10.2.4.4 Copier / Dupliquer une table de forçage

Si vous souhaitez utiliser comme modèle une table de forçage déjà créée, alors dupliquez-la de la manière suivante :

1. Dans la vue "Visualisation/forçage", cliquez sur le bouton "Gérer les tables".
2. Dans le champ des tables de forçage disponibles, sélectionnez celle que vous souhaitez dupliquer comme modèle.
3. Cliquez sur le bouton "Dupliquer".
4. Entrez un nouveau nom pour la table dupliquée.
5. Dans la boîte de dialogue de gestion des tables de forçage, cliquez sur le bouton "Afficher" pour afficher la nouvelle table de forçage dupliquée.

Reprendre des tables de forçage de projets existants

Si pour un nouveau projet, vous souhaitez utiliser une table de forçage existante, procédez de la manière suivante :

1. Dans le projet cible, sélectionnez la vue "Visualisation/forçage".
2. Dans le projet cible, sélectionnez la table de forçage dans laquelle vous souhaitez le contenu de la source.
3. Démarrez STEP 7 Lite une seconde fois et ouvrez le projet source.
4. Dans le projet source, sélectionnez la vue "Visualisation/forçage".
5. Dans le projet source, sélectionnez la table de forçage que vous souhaitez insérer comme source.
6. Dans cette table de forçage, sélectionnez la zone que vous souhaitez reprendre (copier).
7. Choisissez la commande de menu **Edition > Copier**.
8. Passez au projet cible et positionnez le curseur dans la table de forçage du nouveau projet.
9. Choisissez la commande de menu **Edition > Insertion**.

10.2.4.5 Enregistrement d'une table de variables

Vous pouvez réutiliser les tables de variables enregistrées pour visualiser ou forcer les variables lors d'un nouveau test de votre programme.

1. Enregistrez la table de variables avec la commande **Fichier > Enregistrer**.

Lorsque vous enregistrez une table de variables, la largeur des colonnes, le mode de visualisation et le mode de forçage sont également enregistrés.

10.2.4.6 Enregistrer une table de forçage

Lorsque vous effectuez un nouveau test de votre programme, vous pouvez utiliser des tables de forçage enregistrées pour la visualisation et le forçage.

1. Enregistrez la table de forçage en choisissant la commande de menu **Fichier > Enregistrer**.

Lorsque vous enregistrez la table de forçage, vous enregistrez également les largeurs de colonnes et le mode de visualisation.

10.2.5 Saisie de variables dans des tables de variables et des tables de forçage

10.2.5.1 Saisie d'opérandes ou de mnémoniques dans une table de variables

Déterminez les variables dont vous souhaitez visualiser l'état ou que vous désirez forcer, et entrez-les dans la table.

Par exemple, si vous désirez visualiser l'état du bit d'entrée 1.0, du mot de mémoire 5 et de l'octet de sortie 0, entrez les valeurs suivantes dans la colonne de l'opérande :

Exemple

E 1.0
MW 5
AB 0

Exemple de table de variables complétée

La figure montre une table de variables avec les colonnes suivantes : Opérande, Mnémonique, Format d'affichage, Valeur d'état et Valeur de forçage.

	Opérande	Mnémonique	Format affichage	Valeur état	Val forçage
1	//OB1 Réseau 1				
2	E 0.1	"Bouton-poussoir 1"	BOOL	true	
3	E 0.2	"Bouton-poussoir 2"	BOOL	true	
4	A 4.0	"Feu vert"	BOOL	false	
5	//OB1 Réseau 3				
6	E 0.5	"Automatique activé"	BOOL	true	
7	E 0.6	"Manuel activé"	BOOL	true	
8	A 4.2	"Fonction. automatique"	BOOL	true	true
9	//OB1 Appel FB1 pour démarrage moteur essence				
10	E 1.0	"ME_démarrage"	BOOL	false	
11	E 1.1	"ME_arrêt"	BOOL	false	
12	E 1.2	"ME_Défaillance"	BOOL	false	
13	A 5.1	"ME_Consigne_atteinte"	BOOL	false	
14	A 5.0	"ME_marche"	BOOL		true
15	//OB1 Appel FB1 pour démarrage moteur diesel				
16	E 1.4	"MD_démarrage"	BOOL	false	
17	E 1.5	"MD_arrêt"	BOOL		

MPI = 3 (direct) Run

Remarques sur la saisie de mnémoniques

- Vous indiquez la variable à forcer par son opérande (adresse absolue) ou par son mnémonique. Vous pouvez saisir des opérandes et des mnémoniques aussi bien dans la colonne "Opérande" que dans la colonne "Mnémonique". L'entrée est automatiquement reportée dans la colonne qui convient.
Si le mnémonique correspondant est défini dans la table des mnémoniques, vous pourrez saisir l'opérande ou le mnémonique souhaité en le sélectionnant dans une liste qui se déroule.
- Vous ne pouvez inscrire que des mnémoniques déjà définis dans la table des mnémoniques.
- Vous devez saisir les mnémoniques exactement comme ils ont été définis dans la table des mnémoniques.
- Ecrivez entre guillemets les mnémoniques contenant des caractères spéciaux (par exemple, "Moteur.Arrêt", " Moteur+Arrêt", " Moteur- Arrêt").
- Pour définir de nouveaux mnémoniques dans la table des mnémoniques, choisissez la commande **Affichage > Mnémoniques**.

Vérification de la syntaxe

Lorsque vous inscrivez des variables dans la table, la syntaxe de chaque ligne est vérifiée avant que vous la quittiez. Les entrées erronées sont repérées en rouge.

Taille maximale

Une table de variables peut comprendre 1024 lignes au plus.

Exclure certains opérandes du forçage ou de la visualisation

Vous pouvez exclure certains opérandes du forçage ou de la visualisation en désactivant la ligne correspondante. Positionnez le curseur dans la ligne concernée, appelez le menu contextuel (clic avec le bouton droit de la souris) et choisissez la commande **Désactiver la ligne**.

10.2.5.2 Saisie d'opérandes ou de mnémoniques dans une table de forçage

Définissez les variables dont vous souhaitez visualiser ou forcer les valeurs et saisissez-les dans la table de forçage.

Si vous souhaitez p. ex. visualiser le bit d'entrée 1.0 et l'octet de sortie 0, entrez les opérandes suivants dans la colonne des opérandes :

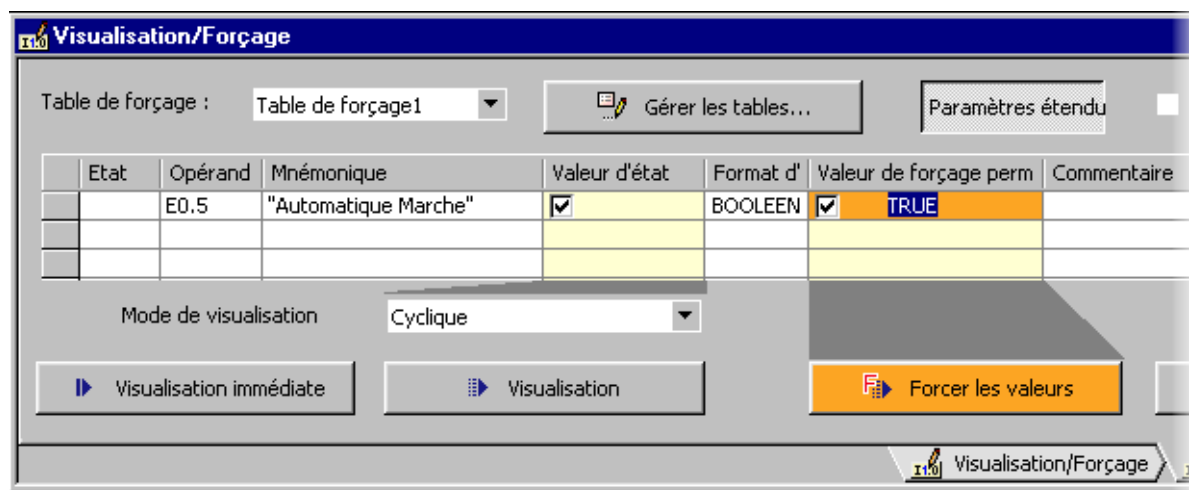
Exemple :

E 1.0

AB 0

Exemple d'une table de forçage complétée

La figure suivante représente une table de forçage avec les colonnes : opérande, mnémonique, format d'affichage, valeur d'état et valeur de forçage.



Remarques sur la saisie de variables sous forme de mnémoniques

- Vous pouvez saisir la variable à forcer sous forme d'opérande ou de mnémonique. Vous pouvez entrer les opérandes et les mnémoniques aussi bien dans la colonne "Opérande" que dans la colonne "Mnémonique". L'entrée s'inscrit automatiquement dans la colonne correspondante.
Si le mnémonique correspondant est défini dans la table des mnémoniques, vous pouvez sélectionner l'opérande ou le mnémoniques souhaités dans une liste durant la saisie.
- Vous pouvez uniquement entrer des mnémoniques qui sont déjà définis dans la table des mnémoniques.
- Un mnémonique doit être saisi exactement de la manière de laquelle il est défini dans la table des mnémoniques.
- Les noms de mnémoniques comportant des caractères spéciaux doivent être saisis entre guillemets (p. ex. "Arrêt.Moteur", "Arrêt+Moteur", "Arrêt-Moteur").
- Pour définir de nouveaux mnémoniques dans la table des mnémoniques, choisissez la commande de menu **Affichage > Mnémonique**.

Vérification de syntaxe

Lorsque vous entrez des variables dans la table de forçage, une vérification de syntaxe est effectuée avant que vous ne quittiez la ligne. Les entrées erronées sont signalées en rouge.

Taille maximale

Une table de forçage peut comporter au maximum 1024 lignes.

Exclure des opérandes individuels de la visualisation et du forçage

Pour exclure des opérandes individuels de la visualisation et du forçage, désactivez la ligne correspondante. Positionnez le curseur dans la ligne souhaitée et choisissez la commande **Désactiver la ligne** dans le menu contextuel (clic avec le bouton droit de la souris).

10.2.5.3 Insertion d'une plage d'opérandes continue dans une table de variables

1. Ouvrez une table de variables.
2. Positionnez le curseur dans la ligne à la suite de laquelle vous souhaitez insérer la plage d'opérandes.
3. Choisissez la commande **Insertion > Plage d'opérandes** : la boîte de dialogue "Insérer une plage d'opérandes" s'affiche.
4. Dans le champ "Opérande initial", tapez p. ex. EW1.
5. Dans le champ "Nombre", tapez le nombre d'opérandes à insérer.
6. Choisissez le format d'affichage souhaité dans la liste affichée.
7. Cliquez sur OK.

La plage d'opérandes est alors insérée dans la table des variables.

10.2.5.4 Insertion d'un plage d'opérandes continue dans une table de forçage

1. Ouvrez une table de forçage.
2. Positionnez le curseur dans la ligne après laquelle vous souhaitez insérer la plage d'opérandes continue.
3. Choisissez la commande de menu **Insertion > Plage d'opérandes**. La boîte de dialogue "Insérer plage d'opérandes" s'ouvre.
4. Dans le champ "A partir de l'opérande" entrez p. ex. EW1.
5. Dans le champ "Nombre" entrez le nombre d'opérandes à insérer.
6. Choisissez le format d'affichage dans la liste proposée.
7. Cliquez sur "OK".

La plage d'opérandes est insérée dans la table de forçage.

10.2.5.5 Limites supérieures pour la saisie de temporisations

Respectez les limites supérieures suivantes lorsque vous saisissez des temporisations :

W#16#3999 (valeur maximale en format DCB)

Exemples

Opérande	Format d'affichage	Saisie	Affichage de la valeur de forçage	Explication
T 1	DUREE_SIMATIC	137	S5TIME#130MS	Conversion en millisecondes
MW 4	DUREE_SIMATIC	137	S5TIME#890MS	Représentation possible en format DCB
MW 4	HEXADECIMAL	137	W#16#0089	Représentation possible en format DCB
MW 6	HEXADECIMAL	157	W#16#009D	Représentation impossible en format DCB, le format d'affichage DUREE_SIMATIC n'est donc pas disponible.

Nota

- Vous pouvez saisir des temporisations à la milliseconde près, mais la valeur saisie sera ajustée à une base de temps. La taille de cette base de temps dépend de celle de la valeur de temps saisie (137 donne 130 ms, les 7 ms ont été arrondies).
- Les valeurs de forçage des opérands de type de données WORD, par ex. EW 1, sont converties au format DCB. Cependant, tout profil binaire n'est pas un nombre DCB correct ! Quand la valeur saisie pour un opérande de type WORD ne peut être représentée en tant que DUREE_SIMATIC, c'est automatiquement le format par défaut qui est utilisé afin que la valeur saisie puisse être affichée.

Format DCB pour les variables au format DUREE_SIMATIC

Les valeurs des variables au format DUREE_SIMATIC sont saisies en format DCB.

Les 16 bits ont la signification suivante :

| 0 0 x x | h h h h | z z z z | e e e e |

bits 15 et 14 sont toujours à zéro.

bits 13 et 12 (marqués par xx) déterminent le multiplicateur pour les bits 0 à 11 :

00 => multiplicateur 10 millisecondes

01 => multiplicateur 100 millisecondes

10 => multiplicateur 1 seconde

11 => multiplicateur 10 secondes

bits 11 à 8 centaine (hhhh)

bits 7 à 4 dizaine (zzzz)

bits 3 à 0 unité (eeee)

10.2.5.6 Limites supérieures pour la saisie de compteurs

Respectez les limites supérieures suivantes lorsque vous saisissez des compteurs :

limites supérieures pour compteurs : C#999

W#16#0999 (valeur maximale en format DCB)

Exemples

Opérande	Format d'affichage	Saisie	Affichage de la valeur de forçage	Explication
Z 1	COMPTEUR	137	C#137	Conversion
MW 4	COMPTEUR	137	C#89	Représentation possible en format DCB
MW 4	HEXADECIMAL	137	W#16#0089	Représentation possible en format DCB
MW 6	HEXADECIMAL	157	W#16#009D	Représentation impossible en format DCB, le format d'affichage COMPTEUR n'est donc pas disponible.

Nota

- Si vous saisissez un nombre décimal pour un compteur sans caractériser la valeur par C#, elle sera automatiquement convertie au format DCB (137 donne C#137).
- Les valeurs de forçage des opérands de type de données WORD, par ex. EW 1, sont converties au format DCB. Cependant, tout profil binaire n'est pas un nombre DCB correct ! Quand la valeur saisie pour un opérande de type WORD ne peut être représentée en tant que COMPTEUR, c'est automatiquement le format par défaut qui est utilisé afin que la valeur saisie puisse être affichée.

10.2.5.7 Exemples

Exemple de saisie d'opérands dans une table de variables

Opérande autorisé :	Type de données :	Exemple (abréviations françaises) :
Entrée Sortie Memento	BOOL	E 1.0 A 1.7 M 10.1
Entrée Sortie Memento	BYTE	EB 1 AB 10 MB 100
Entrée Sortie Memento	WORD	EW 1 AW 10 MW 100
Entrée Sortie Memento	DWORD	ED 1 AD 10 MD 100
Périphérie (entrée sortie)	BYTE	PEB 0 PAB 1
Périphérie (entrée sortie)	WORD	PEW 0 PAW 1
Périphérie (entrée sortie)	DWORD	PED 0 PAD 1
Temporisations	TIMER	T 1
Compteurs	COUNTER	Z 1
Bloc de données	BOOL	DB1.DBX 1.0
Bloc de données	BYTE	DB1.DBB 1
Bloc de données	WORD	DB1.DBW 1
Bloc de données	DWORD	DB1.DBD 1

Exemple de saisie d'opérands dans la table de forçage

Opérande autorisé :	Type de données :	Exemple (abréviations françaises) :
Entrée Sortie Memento	BOOL	E 1.0 A 1.7 M 10.1
Entrée Sortie Memento	BYTE	EB 1 AB 10 MB 100
Entrée Sortie Memento	WORD	EW 1 AW 10 MW 100
Entrée Sortie Memento	DWORD	ED 1 AD 10 MD 100
Périphérie (entrée sortie)	BYTE	PEB 0 PAB 1
Périphérie (entrée sortie)	WORD	PEW 0 PAW 1
Périphérie (entrée sortie)	DWORD	PED 0 PAD 1

Nota

Pour le forçage de modules S7-300, seules les entrées et sorties sont autorisées.

Exemple de saisie d'une plage d'opérandes continue

Ouvrez une table de variables ou la table de forçage et choisissez la commande **Insertion > Plage d'opérandes** pour afficher la boîte de dialogue "Insérer une plage d'opérandes".

Durant la saisie dans la boîte de dialogue, les lignes suivantes seront ajoutées à la table de variables pour des mémentos (M) :

Opérande initial : M 3.0

Nombre : 10

Format d'affichage : BIN

Opérande	Format d'affichage
M 3.0	BIN
M 3.1	BIN
M 3.2	BIN
M 3.3	BIN
M 3.4	BIN
M 3.5	BIN
M 3.6	BIN
M 3.7	BIN
M 4.0	BIN
M 4.1	BIN

Notez comme dans le présent exemple, la désignation change après la huitième entrée dans la colonne "Opérande".

Exemples de saisie de valeurs de forçage/forçage permanent

Opérandes de type bit

Opérandes de type bit possibles	Valeurs de forçage/forçage permanent autorisées
E1.0	true
M1.7	false
A10.7	0
DB1.DBX1.1	1
E1.1	2#0
M1.6	2#1

Opérandes de type octet

Opérandes de type octet possibles	Valeurs de forçage/forçage permanent autorisées
EB 1	2#00110011
MB 12	b#16#1F
MB 14	1F
AB 10	'a'
DB1.DBB 1	10
PAB 2	12

Opérandes de type mot

Opérandes de type mot possibles	Valeurs de forçage/forçage permanent autorisées
EW 1	2#0011001100110011
MW 12	w#16#ABCD
MW 14	ABCD
AW 10	b#(12,34)
DB1.DBW 1	'ab'
PAW 2	12345 (forçage permanent impossible)
MW 3	12345
MW 5	S5t#12s340ms
MW 7	0.3s ou 0,3s
MW 9	C#123
MW 11	d#1990-12-31

Opérandes de type double mot

Opérandes de type double mot possibles	Valeurs de forçage/forçage permanent autorisées
ED 1	2#0011001100110011001100110011
MD 0	1.23e4
MD 4	1.2
AD 10	dw#16#abcdef10
AD 12	ABCDEF10
DB1.DBD 1	b#(12,34,56,78)
PAD 2	'abcd' (forçage permanent impossible)
MD 8	L# -12
MD 12	L#12
MD 16	123456789
MD 20	123456789
MD 24	T#12s345ms
MD 28	Tod#1:2:34.567
MD 32	p#e0.0

Temporisations

Opérandes de type temporisation possibles	Valeurs de forçage/ forçage permanent autorisées	Signification
T 1	0ms	Valeur de temps en millisecondes (ms)
T 12	20ms	Valeur de temps en millisecondes (ms)
T 14	12345ms	Valeur de temps en millisecondes (ms)
T 16	s5t#12s340ms	Valeur de temps égale à 12 s 340 ms
T 18	10ms	Valeur de temps égale à 10 ms
T 20	10.3s	Valeur de temps égale à 10 s 300 ms

Le forçage d'une temporisation n'influe que sur la valeur, pas sur l'état. Ainsi, il est possible de forcer la temporisation T1 à la valeur 0, mais le résultat logique pour U T1 ne change pas.

Les chaînes de caractères "s5t" et "s5time" peuvent être écrites aussi bien en minuscules qu'en majuscules.

Compteurs

Opérandes de type compteur possibles	Valeurs de forçage/forçage permanent autorisées
Z 1	0
Z 14	20
Z 16	c#123

Le forçage d'un compteur n'influe que sur la valeur, pas sur l'état. Ainsi, il est possible de forcer le compteur Z1 à la valeur 0, mais le résultat logique pour U Z1 ne change pas.

10.2.6 Edition de variables dans des tables de variables et des tables de forçage

10.2.6.1 Définition du format d'affichage

Lorsque vous saisissez un opérande, un format d'affichage par défaut lui est automatiquement attribué.

Vous pouvez modifier ce format par défaut :

1. Dans la colonne "Format d'affichage", cliquez sur la cellule correspondante de la ligne.
2. Sélectionnez alors un format parmi la liste déroulante des formats d'affichage admissibles.

10.2.6.2 Couper des zones sélectionnées et les copier dans le presse-papiers

1. Sélectionnez
une ou plusieurs lignes entières en déplaçant le pointeur de haut en bas, tout en maintenant le bouton gauche de la souris enfoncé ;
un opérande, un mnémonique ou une valeur de forçage en déplaçant le pointeur de gauche à droite, tout en maintenant le bouton gauche de la souris enfoncé.
2. Coupez la zone sélectionnée à l'aide de la commande **Edition > Couper**. La zone coupée est consignée dans le presse-papiers jusqu'à ce qu'elle y soit écrasée.

10.2.6.3 Coller le contenu du presse-papiers dans la table de variables ou dans la table de forçage

1. Positionnez le pointeur dans la table de variables à l'endroit où vous désirez coller la zone consignée dans le presse-papiers.
2. Insérez le contenu du presse-papiers à l'aide de la commande **Edition > Coller**.

10.2.6.4 Copie de zones sélectionnées dans le presse-papiers

1. Sélectionnez
une ou plusieurs lignes entières en déplaçant le pointeur de haut en bas, tout en maintenant le bouton gauche de la souris enfoncé ;
un opérande, un mnémonique ou une valeur de forçage en déplaçant le pointeur de gauche à droite, tout en maintenant le bouton gauche de la souris enfoncé.
2. Copiez la zone sélectionnée dans le presse-papiers à l'aide de la commande **Edition > Copier**.

10.2.7 Visualisation de variables

10.2.7.1 Introduction à la visualisation de variables

Vous disposez des possibilités suivantes pour visualiser des variables :

- Activez la fonction de visualisation en cliquant sur le bouton "Visualiser". Les valeurs des variables sélectionnées s'affichent alors dans la table des variables ou dans la table de forçage.

Lorsque le bouton "Paramètres étendus" est activé :

- Activez la fonction de visualisation en cliquant sur le bouton "Démarrer visualisation". Les valeurs des variables sélectionnées s'affichent alors dans la table des variables ou dans la table de forçage en fonction du mode de visualisation choisi. Si vous avez choisi le mode de visualisation cyclique, vous pouvez désactiver la fonction de visualisation en cliquant à nouveau sur le bouton "Démarrer visualisation".
- Actualisez une fois et aussitôt les valeurs des variables sélectionnées en cliquant sur le bouton "Visualisation immédiate". Les valeurs actuelles des variables sélectionnées s'affichent alors dans la table des variables ou dans la table de forçage.

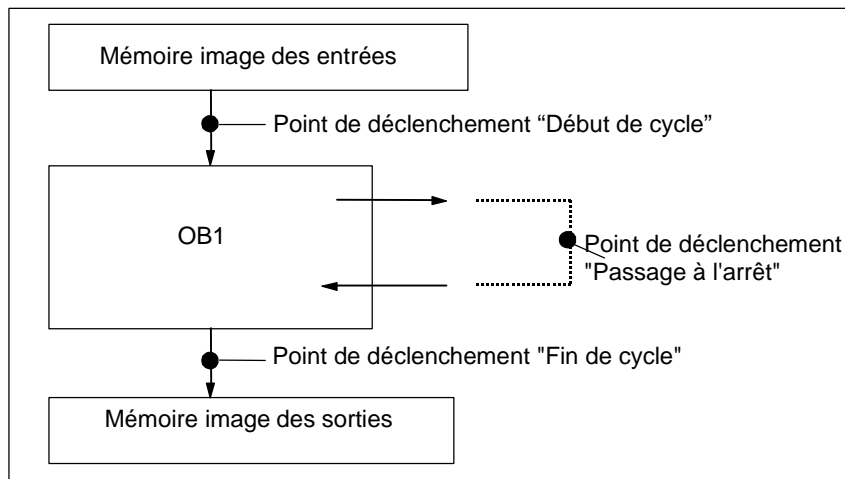
10.2.7.2 Définition du mode de visualisation

En choisissant un mode de visualisation, vous déterminez le point de déclenchement et la durée de la visualisation de variables. Vous disposez des modes suivants :

- Cyclique
- Début de cycle Unique
- Fin de cycle Unique
- Début de cycle Cyclique
- Fin de cycle Cyclique
- Passage à l'arrêt Unique
- Passage à l'arrêt Cyclique

Point de déclenchement

Le point de déclenchement "Début de cycle", "Fin de cycle" ou "Passage à l'arrêt" détermine l'instant où les variables doivent être lues dans la CPU ou bien actualisées. La figure ci-après montre la position des différents points de déclenchement.



Pour que la valeur forcée s'affiche dans la colonne "Valeur d'état", définissez "Permanent" comme point de déclenchement de la visualisation !

Visualisation immédiate

Les boutons "Visualisation immédiate" ou "Forçage immédiat" vous permettent d'actualiser les valeurs de variables sélectionnées. La tâche est exécutée une seule fois et le plus rapidement possible, sans relation avec une position précise du programme utilisateur. Ces fonctions s'utilisent principalement pour visualiser et forcer à l'état d'arrêt (STOP).

10.2.7.3 Visualisation de variables

1. Ouvrez la vue "Visualisation/Forçage" et sélectionnez dans la liste déroulante "Table de variables" ou "Table de forçage" celle des tables qui contient les variables dont vous désirez visualiser l'état.
2. Assurez-vous qu'il existe bien une liaison en ligne à la CPU.
3. Démarrez la visualisation en cliquant sur le bouton "Visualiser".
4. Vous pouvez mettre fin à la visualisation en cliquant à nouveau sur le bouton enfoncé "Visualiser".

10.2.7.4 Visualisation unique et immédiate de variables

Procédez dans l'ordre suivant :

1. Ouvrez la vue "Visualisation/Forçage" et sélectionnez dans la liste déroulante "Table de variables" ou "Table de forçage" celle des tables qui contient les variables dont vous désirez visualiser l'état.
2. Assurez-vous qu'une liaison en ligne est établie à la CPU.
3. En cliquant sur le bouton "Etendu", vous obtenez des possibilités supplémentaires pour visualiser les variables.
4. En cliquant sur le bouton "Visualisation immédiate", vous affichez une fois et aussitôt les valeurs des variables.

10.2.8 Forçage de variables

10.2.8.1 Introduction au forçage de variables

Vous disposez des possibilités suivantes pour forcer des variables :

- Activez la fonction de forçage en cliquant sur le bouton "Forçage". Désactivez la fonction de forçage en cliquant une nouvelle fois sur le bouton "Forçage".

Lorsque le bouton "Paramètres étendus" est activé :

- Activez la fonction de forçage en cliquant sur le bouton "Forçage". Le programme utilisateur affecte aux variables sélectionnées les valeurs de forçage figurant dans la table des variables, en fonction du mode de forçage choisi. Si vous avez choisi un mode de forçage cyclique, vous pouvez désactiver la fonction de forçage en cliquant à nouveau sur le bouton "Forçage".
- Actualisez une fois et aussitôt les valeurs des variables sélectionnées en cliquant sur le bouton "Forçage immédiat".

Des possibilités supplémentaires vous sont offertes par les fonctions "Forçage permanent" et "Débloquer sorties périphériques".

A noter pour le forçage

- Il n'est pas possible d'annuler le forçage (par exemple avec la commande **Edition > Annuler**).



Danger

Modifier les valeurs des variables alors que l'installation est en marche peut, en cas de défaut de fonctionnement ou d'erreurs dans le programme, entraîner des blessures corporelles graves et des dégâts matériels importants.

Assurez-vous qu'aucun état dangereux ne peut apparaître avant d'exécuter la fonction "Forçage".

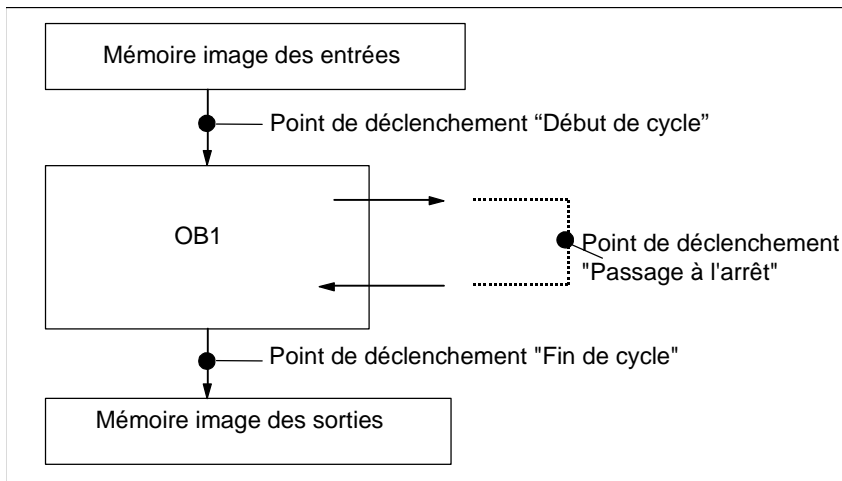
10.2.8.2 Définition du mode de forçage

En choisissant un mode de forçage, vous déterminez le point de déclenchement et la durée du forçage de variables. Vous disposez des modes suivants :

- Permanent
- Début de cycle Unique
- Fin de cycle Unique
- Début de cycle Cyclique
- Fin de cycle Cyclique
- Passage à l'arrêt Unique
- Passage à l'arrêt Cyclique

Point de déclenchement

Le point de déclenchement "Début de cycle", "Fin de cycle" ou "Passage à l'arrêt" détermine l'instant où les variables doivent être lues dans la CPU ou bien actualisées. La figure ci-après montre la position des différents points de déclenchement.



La position des points de déclenchement a les conséquences suivantes :

- Forcer des entrées n'a de sens qu'au début du cycle (c'est-à-dire au commencement du programme utilisateur OB1), car sinon la mémoire image des entrées serait de nouveau mise à jour après le forçage et donc écrasée.
- Forcer des sorties n'a de sens qu'à la fin du cycle (c'est-à-dire à la fin du programme utilisateur OB1), car sinon la mémoire image des sorties pourrait être écrasée par le programme utilisateur.

Pour que la valeur forcée s'affiche dans la colonne "Valeur d'état", définissez "Permanent" comme point de déclenchement de la visualisation.

Pour le forçage de variables, notez bien que :

- si vous choisissez le mode "Unique", un message vous signalera quand les variables sélectionnées ne peuvent pas être forcées ;
- mais avec le mode "Cyclique", ce problème ne vous sera pas signalé.

Forçage immédiat

Le bouton "Forçage immédiat" vous permet de forcer les valeurs de variables sélectionnées. La tâche est exécutée une seule fois et le plus rapidement possible, sans relation avec une position précise du programme utilisateur. Cette fonction s'utilise principalement pour forcer à l'état d'arrêt (STOP).

10.2.8.3 Forçage de variables

1. Ouvrez la vue "Visualisation/Forçage" et sélectionnez dans la liste déroulante "Table de variables" celle des tables qui contient les variables dont vous désirez visualiser l'état ou bien activez la fenêtre contenant la table de variables qui vous occupe.
2. Etablissez la liaison à la CPU de votre choix afin de pouvoir forcer les variables de la table en vigueur.
3. Sélectionnez le mode de forçage approprié.

Attention !

Sélectionner un mode de forçage **n'est pas possible** lorsqu'un forçage est en cours ! Le cas échéant, arrêtez le forçage. Il est inactif quand le bouton "Démarrer forçage" n'est pas enfoncé.

4. Entrez les valeurs de forçage pour les variables à forcer dans la colonne "Valeur de forçage" et activez la case à cocher à côté de la valeur de forçage.
5. Lancez le forçage en cliquant sur le bouton "Démarrer forçage".
Si vous avez choisi le mode de forçage cyclique, le bouton restera enfoncé.
Si vous avez choisi le mode de forçage unique, la fonction sera exécutée une fois, puis le bouton reprendra sa position initiale.
6. Si vous souhaitez imposer de nouvelles valeurs, définir un autre déclenchement ou mettre fin au forçage, cliquez à nouveau sur le bouton "Démarrer forçage" enfoncé pour le faire remonter.
Reprenez à l'étape 3 pour convenir d'un autre mode de forçage.
Reprenez à l'étape 4 pour imposer de nouvelles valeurs.

10.2.8.4 Forçage immédiat de variables

Procédez dans l'ordre suivant :

1. Ouvrez la table renfermant les variables que vous désirez forcer.
2. Entrez les valeurs de forçage pour les variables à forcer dans la colonne "Valeur de forçage" et activez les cases à cocher à côté des valeurs.
3. Cliquez sur le bouton "Forçage immédiat" pour affecter des valeurs aux variables une seule fois et immédiatement.

10.2.8.5 Forçage : initialiser la CPU à l'arrêt avec vos propres valeurs

Procédez de la manière suivante :

1. Ouvrez le panneau de commande de la CPU et mettez-la à l'état "Arrêt".
2. Dans la table de variables, écrivez la valeur de forçage souhaitée pour chaque variable et activez les cases à cocher à côté des valeurs de forçage.
3. Activez les valeurs de forçage en cliquant sur le bouton "Démarrer forçage".
4. Remettez la CPU à l'état "Marche" à l'aide du panneau de commande.

10.2.8.6 Forçage des sorties de périphérie à l'état "Arrêt" de la CPU

La fonction "Débloquer PA" désactive le blocage des sorties de périphérie (PA). Cela permet de forcer les sorties de périphérie durant l'état "Arrêt" de la CPU.

Procédez de la manière suivante :

1. Ouvrez la vue "Visualisation/Forçage" et sélectionnez dans la liste déroulante "Tables de variables" celle qui contient les sorties de périphérie à forcer.
2. Ouvrez le panneau de commande de la CPU et mettez cette dernière à l'état de fonctionnement Arrêt.
3. Dans la colonne "Valeur de forçage", saisissez les valeurs appropriées pour les sorties de périphérie à forcer et activez les cases à cocher à côté des valeurs.
4. Activez le mode "Débloquer PA" en cochant la case correspondante dans la vue "Visualisation/Forçage".
5. Forcez les sorties de périphérie en cliquant sur le bouton "Forçage immédiat".
6. Le mode "Débloquer PA" reste actif jusqu'à ce que vous retiriez la coche de la case correspondante.
7. Pour définir de nouvelles valeurs de forçage, recommencez au point numéro 3.

Nota

- Le mode "Débloquer PA" n'est possible qu'à l'état d'arrêt (STOP).
 - Les événements suivants mettent fin au mode "Débloquer PA" :
 - changement d'état de fonctionnement de la CPU (un message s'affiche),
 - suppression de la coche dans la case "Débloquer PA".
-

10.2.9 Forçage permanent de variables

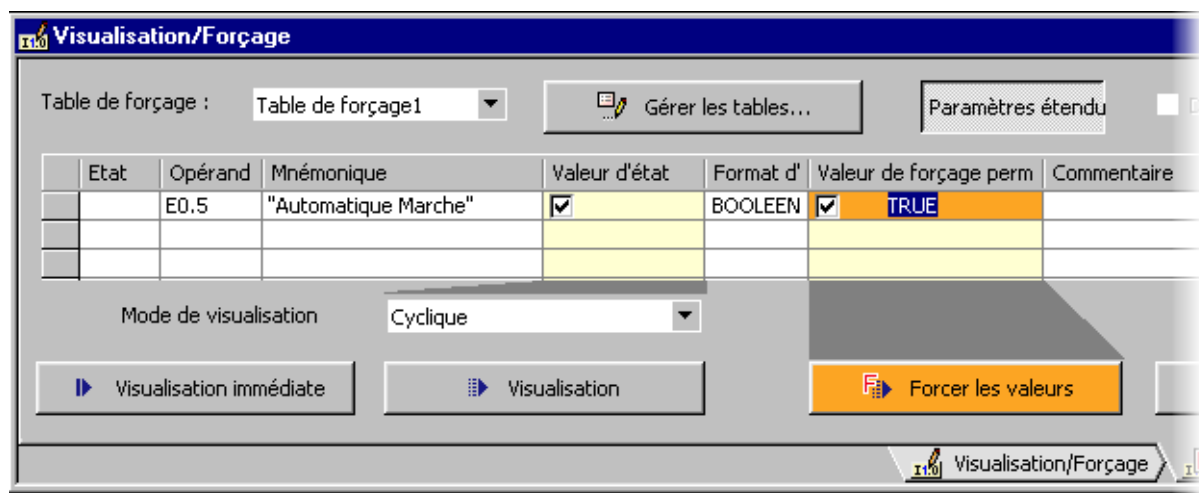
10.2.9.1 Introduction au forçage permanent de variables

Les fonctions de forçage permanent ne sont disponibles que dans la vue "Forçage permanent".

Pour afficher cette vue, cliquez deux fois sur "Visualisation/Forçage" dans la fenêtre du projet, puis sélectionnez l'onglet "Forçage permanent".

Vous pouvez affecter des valeurs fixes à des variables individuelles d'un programme utilisateur afin qu'elles ne puissent être ni modifiées ni écrasées, même par le programme utilisateur exécuté dans la CPU. Il faut évidemment que la CPU assume cette fonction.

Exemple



10.2.9.2 Mesures de sécurité pour le forçage permanent de variables



Vous devez éviter des lésions corporelles ou un dommage matériel !

Notez bien qu'une erreur de manipulation de la fonction "Forçage permanent" risque

- de mettre en danger la vie ou la santé des opérateurs,
- d'endommager la machine ou l'ensemble de l'installation.



Avertissement

- Avant de lancer la fonction de forçage permanent, assurez-vous que personne d'autre ne l'exécute simultanément sur la même CPU.
- Seul un clic sur le bouton "Annuler le forçage" peut effacer une tâche de forçage permanent ou y mettre fin. Elle ne sera pas effacée par la fermeture de la vue "Forçage permanent" ni par celle de la vue "Visualisation/Forçage".
- La commande **Edition > Annuler** ne permet pas d'annuler le forçage permanent.
- Renseignez-vous sur les différences entre forçage de variables et forçage permanent de variables.
- Aucun des boutons concernant le forçage permanent n'est disponible quand une CPU n'assume pas la fonction de forçage permanent.

Tous les modules de sorties faisant l'objet d'un forçage permanent indiquent leur valeur de forçage permanent quand le blocage des sorties est supprimé au moyen de la case d'option "débloquer PA".

10.2.9.3 Affichage des valeurs forcées de manière permanente par la CPU

1. Assurez-vous qu'une liaison en ligne est établie avec la CPU.
2. Ouvrez la vue "Visualisation/forçage" et choisissez la table "Standard" dans la liste déroulante "Table de forçage" de l'onglet "Visualiser/forcer".

10.2.9.4 Création d'une tâche de forçage permanent

1. Ouvrez la vue "Visualisation/forçage" et choisissez l'onglet "Visualisation / Forçage permanent".
2. Dans la colonne "Opérande", entrez les variables à forcer.
3. Dans la colonne "Valeur de forçage", entrez les valeurs pour les variables à forcer et activez les cases à cocher à côté des valeurs.

4. Lancez le forçage permanent en cliquant sur le bouton "Forcer les valeurs".

Résultat :

- S'il n'y a pas de tâche de forçage permanent active, les variables prennent les valeurs de forçage.
- S'il y a déjà une tâche de forçage permanent active, vous devez choisir de la remplacer ou pas.
Si cette tâche existante n'est pas de vous, veuillez consulter son auteur avant de la remplacer.

10.2.9.5 Annulation d'une tâche de forçage permanent

Pour mettre fin au forçage permanent, cliquez sur le bouton "Annuler le forçage". Si vous n'êtes pas l'auteur de la tâche de forçage permanent, veuillez consulter celui qui en est à l'origine.

Les valeurs de forçage permanent ne sont pas effacées dans la CPU par la fermeture de la fenêtre qui les affiche ni par la bascule dans une autre vue.

10.2.9.6 Différences entre forçage de variables et forçage permanent de variables

Le tableau suivant résume les différences entre forçage et forçage permanent.

Caractéristique / fonction	Forçage permanent avec CPU 318-2DP	Forçage permanent avec S7-300 (sans CPU 318-2DP)	Forçage
Mémentos (M)	oui	–	oui
Temporisations et compteurs (T, Z)	–	–	oui
Blocs de données (DB)	–	–	oui
Entrées de périphérie (PEB, PEW, PED)	oui	–	–
Sorties de périphérie (PAB, PAW, PAD)	oui	–	oui
Entrées et sorties (E, A)	oui	oui	oui
Le programme utilisateur peut écraser les valeurs de forçage/forçage permanent.	–	oui	oui
Le remplacement de la valeur de forçage permanent prend effet sans interruption.	oui	oui	–
Les variables conservent leurs valeurs après la fermeture de l'application.	oui	oui	–
Les variables conservent leurs valeurs une fois la liaison à la CPU suspendue.	oui	oui	–
Définition du déclenchement	Toujours déclenchement immédiat	Toujours déclenchement immédiat	Unique ou cyclique

Nota

- Avec la fonction "Débloquer PA", les valeurs de forçage permanent pour les sorties de périphérie concernées prennent effet aux modules correspondants, mais pas les valeurs de forçage pour les sorties de périphérie forcées de manière cyclique.

10.3 Test avec la visualisation d'état du programme

10.3.1 Test avec la visualisation d'état du programme

Vous pouvez tester votre programme en affichant, pour chaque instruction, l'état du programme (RLG, bit d'état) ou le contenu des registres correspondants. Sélectionnez les informations à afficher dans la boîte de dialogue "Paramètres" que vous ouvrirez à l'aide de la commande **Outils > Paramètres** dans la vue de l'éditeur de bloc.



Attention

Si vous effectuez le test d'une installation en marche, d'éventuels défauts de fonctionnement ou erreurs de programmation risquent d'occasionner des dommages matériels et personnels graves !

Avant d'exécuter une fonction, assurez-vous qu'aucune situation dangereuse ne peut se produire !

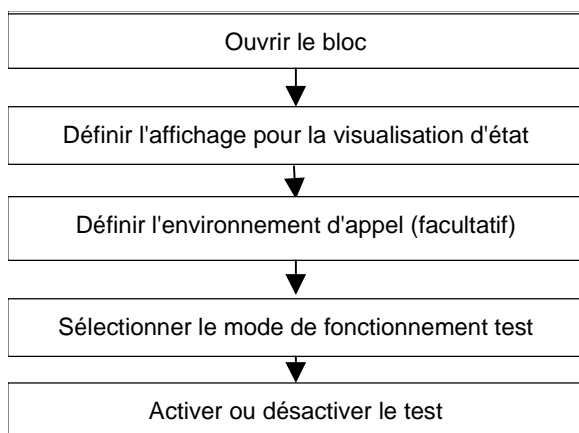
Conditions préalables

Pour pouvoir afficher l'état du programme, il faut que les conditions suivantes soient remplies :

- Vous avez enregistré le bloc sans erreurs, puis l'avez chargé dans la CPU.
- La CPU est en marche, le programme utilisateur s'exécute.
- Vous avez ouvert le bloc.

Marche à suivre de principe pour visualiser l'état du programme :

Il est fortement recommandé de ne pas appeler et tester immédiatement le programme complet, mais plutôt les blocs un par un. Ce faisant, il faut commencer par les blocs de code de niveau inférieur, c'est-à-dire ceux qui se trouvent au dernier niveau d'imbrication de la hiérarchie d'appel. Vous les appelez dans l'OB1, par exemple, et créez l'environnement à tester au moyen de la commande **Visualisation et forçage de variables**.



10.3.2 Affichage dans la visualisation d'état de programme

L'affichage de la **visualisation d'état de programme** est actualisé cycliquement. Il débute avec le réseau sélectionné.

Valeurs par défaut

- Etat satisfait : lignes continues en vert
- Etat non satisfait : lignes pointillées en bleu
- Etat inconnu : lignes continues en noir

Vous pouvez modifier ces valeurs pour le type et la couleur des lignes dans la page d'onglet "CONT/LOG" que vous affichez via la commande **Outils > Paramètres**.

Etat des éléments

- L'état d'un contact :
 - est satisfait lorsque l'opérande a la valeur "1" ;
 - n'est pas satisfait lorsque l'opérande a la valeur "0" ;
 - est inconnu lorsque la valeur de l'opérande est inconnue.
- L'état d'éléments avec sortie de validation (ENO) correspond à l'état d'un contact avec la valeur de la sortie ENO comme opérande.
- L'état d'éléments avec sortie Q correspond à l'état d'un contact avec la valeur de l'opérande.
- L'état pour des opérations CALL est satisfait lorsque le bit de résultat binaire est à 1 après l'appel.
- L'état d'une opération de saut est satisfait lorsque le saut est exécuté, c'est-à-dire lorsque la condition de saut est satisfaite.
- Les éléments avec sortie de validation (ENO) sont représentés en noir lorsque la sortie de validation n'est pas définie.

Etat des lignes

- Les lignes sont en noir lorsqu'elles n'ont pas été empruntées ou que leur état est inconnu.
- L'état des lignes commençant à la barre d'alimentation est toujours satisfait ("1").
- L'état des lignes au début de branches parallèles est toujours satisfait ("1").
- L'état des lignes après un élément est satisfait lorsque l'état de la ligne avant l'élément et l'état de l'élément sont satisfaits.
- L'état de la ligne après NOT est satisfait lorsque l'état de la ligne avant NOT n'est pas satisfait (et inversement).
- L'état de la ligne **après** la jonction de plusieurs lignes est satisfait :
 - lorsque, d'une part, l'état d'une ligne au moins **avant** la jonction est satisfait
 - et que, d'autre part, l'état de la ligne avant l'ouverture de la branche ou des branches est satisfait.

Etat des paramètres

- Les valeurs de paramètres **en gras** sont les valeurs en cours.

- Les valeurs de paramètres en écriture normale proviennent d'un cycle précédent ; il n'y a pas eu de passage par cet endroit du programme pendant le cycle en cours.

10.3.3 Etat du programme de blocs de données

Vous pouvez visualiser un bloc de données en ligne dans la vue des données.

Le bloc de données ne doit pas être modifié avant l'appel de l'état du programme. En cas de différence structurelle (déclaration) entre le bloc de données en ligne et le bloc de données hors ligne, vous pouvez directement charger le bloc de données hors ligne dans la CPU.

Le bloc de données doit se trouver dans la "vue des données", afin que les valeurs en ligne puissent être représentées dans la colonne "Valeur actuelle". Seule la partie du bloc de données visible à l'écran est actualisée. Pendant que l'état est actif, vous ne pouvez pas passer à la vue des déclarations.

Durant l'actualisation, la barre de défilement verte est visible dans la barre d'état du bloc de données et l'état de fonctionnement est affiché.

Les valeurs sont affichées dans le format du type de données respectif. Une modification du format n'est pas possible.

Lorsque vous mettez fin à l'état du programme, le contenu qui était préalablement valable s'affiche à nouveau dans la colonne des valeurs actuelles. Vous ne pouvez pas reprendre les valeurs en ligne actualisées dans le bloc de données hors ligne.

Actualisation de types de données

Tous les types de données simples sont aussi bien actualisés dans un DB global que dans toutes les déclarations (in/out/inout/stat) d'un bloc de données d'instance.

Certains types de données ne peuvent pas être actualisés. Lorsque l'état du programme est activé, les champs contenant des données non actualisées sont estompés dans la colonne "Valeur actuelle".

- Les types de données complexes DATE_AND_TIME et STRING ne sont pas actualisés.
- Dans les types de données complexes ARRAY, STRUCT, UDT, FB, SFB, seuls les éléments qui sont des types de données simples sont actualisés.
- Dans la déclaration Inout d'un bloc de données d'instance, seul le pointeur sur le type de données complexe est représenté mais ses éléments ne le sont pas. Le pointeur n'est pas actualisé.
- Les types de paramètre ne sont pas actualisés.

10.3.4 Détails sur le test dans l'état du programme

10.3.4.1 Définition de l'affichage de l'état du programme

Vous pouvez définir vous-même l'affichage de l'état du programme dans un bloc LIST, LOG ou CONT.

Procédez de la manière suivante :

1. Choisissez la commande **Outils > Paramètres**.
2. Sélectionnez l'entrée "Editeur de bloc" dans la liste déroulante située dans le haut de la boîte de dialogue "Paramètres".
3. Choisissez les options désirées pour le test. Vous pouvez afficher les champs d'état suivants :

Lorsque vous cochezvous voyez s'afficher :
Bit d'état	Bit d'état, c.-à-d. bit 2 du mot d'état
Résultat logique	Bit 1 du mot d'état ; indique le résultat d'une fonction logique ou d'une comparaison arithmétique.
Accu 1	Contenu de l'ACCU 1.
Registre d'adresse 1/2	Contenu du registre d'adresse correspondant dans le cas de l'adressage indirect (intrazone ou interzone) par registre
Accu 2	Contenu de l'ACCU 2
Registre des DB 1/2	Contenu du registre des DB du premier ou du deuxième bloc de données ouvert.
INFOS	Référence mémoire indirecte ; indication du pointeur (adresse), pas du contenu de l'adresse ; possible uniquement pour l'adressage indirect en mémoire, pas pour l'adressage indirect par registre. Contenu d'un mot de temporisation ou d'un mot de compteur lorsque l'instruction comporte les opérations correspondantes.
Mot d'état	Tous les bits d'état du mot d'état

10.3.4.2 Définition de l'environnement d'appel du bloc

Vous pouvez indiquer des conditions précises pour la visualisation de l'état du programme en définissant l'environnement d'appel. L'état du programme n'est alors enregistré que lorsque la condition de déclenchement précisée est satisfaite.

Procédez de la manière suivante :

1. Choisissez la commande **Test > Conditions d'appel**.
2. Fixez, dans la boîte de dialogue "Conditions d'appel du bloc" qui apparaît alors, vos conditions de déclenchement et confirmez par "OK".

Sélection possible	Signification
Chemin d'appel	Vous pouvez indiquer ici le chemin suivant lequel bloc à tester doit être appelé pour déclencher une visualisation d'état. Vous pouvez saisir les trois derniers niveaux d'appel précédant le bloc à tester.
Avec adresse	Désactivez cette option s'il faut annuler la condition par chemin d'appel.
Blocs de données ouverts	L'environnement d'appel est défini ici par l'indication d'un ou de deux blocs de données. La visualisation d'état est réalisée lorsque le bloc à tester a été appelé avec les blocs de données respectifs indiqués.

Définir l'environnement d'appel d'instances de bloc

Pour afficher l'état du programme d'un bloc dans une instance donnée, procédez de la manière suivante :

1. Choisissez la commande **Test > Mode de fonctionnement** et sélectionnez le "Mode test".
2. Ouvrez le bloc appelant et positionnez le pointeur sur l'instruction d'appel souhaitée (ligne CALL dans LIST ou boîte du bloc dans CONT/LOG).
3. En cliquant sur le bouton droit de la souris, choisissez la commande **Bloc appelé > Visualiser avec chemin d'appel**.

Résultat : le bloc appelé s'ouvre, l'appel est inscrit comme critère dans la condition de déclenchement du bloc et l'état est activé pour cette instance du bloc.

Les conditions de déclenchement des blocs de données existantes restent inchangées.

10.3.4.3 Choix du mode de fonctionnement pour le test

Les fonctions de test allongent le temps de cycle du programme utilisateur.

Vous pouvez influencer sur la charge du temps de cycle due aux fonctions de test en commutant entre le mode de test et le mode de processus.

Durant la phase de mise en service, vous pouvez par exemple sélectionner le mode test impliquant un temps de cycle plus long.

Durant le fonctionnement, vous pouvez choisir un temps de cycle plus court et le mode de processus influant sur l'exécution de la visualisation du programme (voir plus bas).

Choix du mode de fonctionnement pour le test

Vous avez en principe deux possibilités pour sélectionner le mode de fonctionnement. Il n'existe qu'une seule possibilité de le faire sur la CPU.

- Dans le cadre du paramétrage de la CPU dans la rubrique "Fonctionnement" des paramètres de la CPU (par exemple pour les CPU S7-300). Le paramétrage doit être chargé dans la CPU afin que le mode de fonctionnement choisi entre en vigueur.
- Online dans le cadre du test du programme si le bloc de code est ouvert via la commande de menu **Test > Mode de fonctionnement**.

Nota : Si la commutation des modes de fonctionnement est effectuée dans le cadre du paramétrage de la CPU, seul le mode de fonctionnement paramétré est affiché ici et il n'est pas commutable.

Conséquences du mode de fonctionnement paramétré

Mode de fonctionnement	Signification
Mode test	Toutes les fonctions de test peuvent être utilisées sans restrictions. D'importantes augmentations du temps de cycle de la CPU sont possibles, puisque l'état des instructions dans les boucles programmées est déterminé à chaque passage, par exemple.
Mode processus	La fonction de test "Etat du programme" est restreinte pour garantir une charge du temps de cycle la plus faible possible. <ul style="list-style-type: none"> • Ainsi, par exemple, les conditions d'appel ne sont pas autorisées. • La visualisation d'état d'une boucle programmée est interrompue à la position de retour. • Les fonctions de test "ATTENTE" et l'exécution du programme pas à pas ne sont pas possibles.

10.3.4.4 Forçage de variables dans l'état du programme

Condition préalable : le bloc en ligne est ouvert.

Les actions décrites ci-après entraînent un forçage unique et immédiat des variables sélectionnées.

Forçage de variables de type BOOL

1. Sélectionnez l'opérande que vous souhaitez forcer.
2. Choisissez la commande **Test > Forcer à 1** ou **Test > Forcer à 0**.

Forçage de variables non booléennes

1. Sélectionnez l'opérande que vous souhaitez forcer.
2. Choisissez la commande **Test > Forcer**.
3. Dans la boîte de dialogue affichée, entrez la valeur que doit prendre la variable (valeur de forçage).
4. Fermez la boîte de dialogue.

Vous pouvez également procéder de la manière suivante :

1. Positionnez le pointeur au-dessus de l'opérande que vous souhaitez forcer.
2. Cliquez sur le bouton droit de la souris et choisissez la commande de forçage correspondante dans le menu contextuel qui s'affiche.

10.3.4.5 Activation et désactivation du test avec état du programme

1. Déclenchez la visualisation de l'état du programme avec la commande **Test > Visualiser** (commande cochée) ou avec l'une des commandes du menu contextuel **Bloc appelé > Visualiser** ou **Bloc appelé > Visualiser avec chemin d'appel**.
2. Évaluez l'état du programme LIST du bloc qui s'affiche sous forme de table.
3. Vous pouvez désactiver la visualisation en sélectionnant à nouveau la commande **Test > Visualiser** (commande non cochée).

11 Diagnostic

11.1 Fonctions de diagnostic

Le diagnostic système détecte, évalue et signale les erreurs survenant au sein d'un automate programmable. Chaque CPU dispose à cet effet d'une mémoire tampon de diagnostic dans laquelle sont inscrites des informations détaillées sur tous les événements de diagnostic dans l'ordre de leur apparition.

Evénements de diagnostic

Les événements suivants provoquent des entrées dans la mémoire tampon de diagnostic :

- les erreurs internes et externes sur un module,
- les erreurs système dans la CPU,
- les changements d'état de fonctionnement (par ex. de "Marche" à "Arrêt"),
- les erreurs dans le programme utilisateur,
- le débrochage/enfichage de modules,
- les messages personnalisés inscrits au moyen de la fonction système SFC52.

Le contenu de la mémoire tampon de diagnostic est conservé en cas d'effacement général des CPU. Grâce à la mémoire tampon de diagnostic, les erreurs dans le système peuvent être évaluées même après quelque temps, en vue de déterminer la cause d'un passage à l'"Arrêt" ou de remonter la trace des événements de diagnostic individuels.

Enregistrement des données de diagnostic

Il n'est pas nécessaire de programmer l'enregistrement de données de diagnostic par le diagnostic système, car il se fait automatiquement. SIMATIC S7 propose différentes fonctions de diagnostic. Certaines d'entre elles sont intégrées dans la CPU et d'autres sont mises à votre disposition par les modules (SM).

Indication des erreurs

Les erreurs internes et externes aux modules sont signalées par des diodes électroluminescentes en face avant du module concerné. Les signalisations par DEL et leur évaluation sont décrites dans les manuels consacrés au matériel S7. Dans S7-300, les erreurs internes et externes sont réunies en une erreur groupée.

La CPU détecte les erreurs système ainsi que les erreurs dans le programme utilisateur et elle inscrit des événements de diagnostic dans la liste d'état système et dans la mémoire tampon de diagnostic. Il est possible de lire ces messages à la console de programmation.

Les modules aptes au diagnostic détectent les erreurs de module internes et externes et génèrent une alarme de diagnostic à laquelle vous pouvez réagir à l'aide d'un OB d'alarme.

11.2 Diagnostic du matériel et recherche d'erreurs

Marche à suivre

- Vérifiez d'abord que la configuration paramétrée (c'est-à-dire celle définie dans la vue "Configuration HW") est identique à la configuration chargée. L'identité ou la divergence vous seront signalées dans la vue "Comparaison HW".
- Examinez alors s'il y a des erreurs sur certains modules.
- Demandez l'affichage des erreurs de module.

Autres possibilités de diagnostic dans la vue de diagnostic

Un double-clic sur un module affiche l'état de ce module.

11.3 Comparaison des configurations 'en ligne/hors ligne/physique'

Introduction


On appelle "configuration en ligne" celle qui est chargée dans la CPU et "configuration hors ligne" celle que vous avez paramétrée.


La configuration détectée par une CPU de sa propre initiative, sans qu'aucune configuration ait été chargée, est nommée "physique".


Concept de la configuration matérielle

La vue "Comparaison HW" compare la configuration paramétrée (hors ligne) à la configuration chargée (en ligne) et à la physique. Les différences sont repérées par des symboles dans la table de configuration. Cette dernière indique les modules ou paramètres de modules qui ne concordent pas.

Par exemple, si c'est un module d'entrées TOR qui a été configuré dans une ligne, mais que le module effectivement enfiché est un module de sorties TOR, la ligne (=emplacement) en question sera repérée par le symbole "différent" :


Symbole pour "différent" : 

Si le module est certes configuré, mais qu'il manque en ligne, le symbole qui s'affichera est le suivant : 

Si le module effectivement enfiché correspond bien au module configuré, mais qu'il possède d'autres paramètres, le symbole qui s'affichera est le suivant : 

Nota

Quand il compare la physique à la configuration hors ligne ou en ligne, STEP 7 Lite ne peut pas déterminer intégralement l'identité de la physique. Le type de module sera bien établi et comparé, mais pas le numéro de référence exact.

Dans ce cas, c'est le symbole "Modules probablement identiques" qui s'affiche : 

Détail des différences trouvées lors de la comparaison matérielle

Sous la table de configuration, la "liste Delta" énumère en détail les différences entre les modules classés dans l'ordre de leurs emplacements.

Lorsque vous souhaitez comparer les paramètres de modules, effectuez un double clic sur le module correspondant dans la vue "Comparaison HW". La boîte de dialogue des paramètres, protégée en écriture, caractérise les divers paramètres au moyen de couleurs. Les valeurs des paramètres concernées sont représentées en jaune.

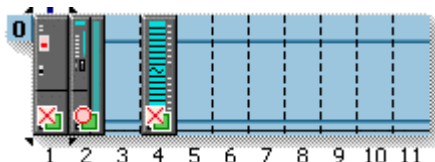
Les boutons "Différence suivante" et "Différence précédente" vous permettent de naviguer rapidement aux paramètres concernés.

11.4 La vue 'Comparaison HW'

La vue "Comparaison HW" donne deux représentations de la configuration de station en cours.

- La vue graphique montre d'une manière très proche de la réalité la disposition des modules à leurs emplacements d'enchâssement, accompagnés de symboles indiquant l'égalité ou la différence de la configuration (en ligne/hors ligne/physique).
- La table de configuration donne des renseignements détaillés sur les modules enchâssés (par ex. adresses et numéros de référence) ; des colonnes différentes sont consacrées aux configurations comparées (par ex. hors ligne – en ligne).

La vue comporte en outre des boutons permettant de basculer d'une table de comparaison à l'autre.

Zone de la vue 'Comparaison HW'					Signification																																				
					<p>Vue graphique de la configuration matérielle avec emplacement sélectionné.</p> <p>Une flèche va de l'emplacement sélectionné à l'emplacement de même valeur dans la table de configuration. S'il y a une liaison en ligne à la CPU, l'état de synchronisation est indiqué par un symbole (égal/différent).</p> <p>Le bouton "Comparaison : ..." sert à déterminer ce qui doit être comparé.</p>																																				
<table><tr><td></td><td colspan="2">Hors ligne</td><td></td><td colspan="2">En ligne</td></tr><tr><td>Empl</td><td>Module</td><td>Référence</td><td></td><td>Module</td><td>Référence</td></tr><tr><td>1</td><td>PS307 10A</td><td>6ES7 307-1KA0...</td><td></td><td></td><td></td></tr><tr><td>2</td><td>CPU314 C-...</td><td>6ES7 314-6CF0...</td><td></td><td>CPU314 C-...</td><td>6ES7 314-6CF00-0...</td></tr><tr><td>3</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>SM 331 AI2...</td><td>6ES7 331-7KB8...</td><td></td><td></td><td></td></tr></table>						Hors ligne			En ligne		Empl	Module	Référence		Module	Référence	1	PS307 10A	6ES7 307-1KA0...				2	CPU314 C-...	6ES7 314-6CF0...		CPU314 C-...	6ES7 314-6CF00-0...	3						4	SM 331 AI2...	6ES7 331-7KB8...				<p>Emplacement sélectionné dans la table de configuration.</p> <p>Selon le type de comparaison, les colonnes de la table portent un autre titre. Entre les configurations à comparer s'affichent des symboles indiquant l'état de synchronisation (égal/différent).</p>
	Hors ligne			En ligne																																					
Empl	Module	Référence		Module	Référence																																				
1	PS307 10A	6ES7 307-1KA0...																																							
2	CPU314 C-...	6ES7 314-6CF0...		CPU314 C-...	6ES7 314-6CF00-0...																																				
3																																									
4	SM 331 AI2...	6ES7 331-7KB8...																																							
<p>Liste Delta :</p> <div> Châssis 0, Emplacement 1 Module uniquement hors ligne</div> <div> Châssis 0, Emplacement 2 Les modules sont identiques</div> <div> Châssis 0, Emplacement 4 Module uniquement hors ligne</div>					<p>Liste Delta indiquant les différences.</p> <p>Elle mentionne les paramètres dont les valeurs se sont montrées différentes lors de la comparaison des configurations. Ces valeurs différentes sont affichées dans les colonnes appropriées.</p>																																				

11.5 Détecter les modules défectueux

Pour pouvoir détecter les modules défectueux, il faut établir une liaison en ligne entre la PG/le PC et la CPU.

Marche à suivre

1. Cliquez deux fois sur "Matériel" dans la vue d'ensemble du projet.
2. Sélectionnez la vue "Diagnostic HW".

La vue "Diagnostic HW" montre la configuration de station établie à partir de la CPU. Les icônes de diagnostic indiquent s'il y a des informations de diagnostic pour un module. Elles montrent l'état du module concerné ainsi que l'état de fonctionnement quand il s'agit d'une CPU.

Pour obtenir des informations de diagnostic plus détaillées sur chaque module, cliquez sur le bouton "Informations de diagnostic étendues" pour ouvrir la boîte de dialogue "Etat du module".

Mise à jour de l'affichage

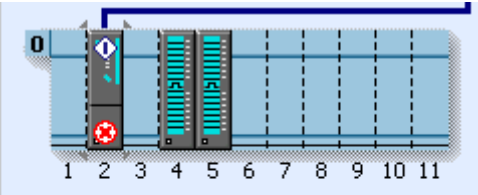







Il y a plusieurs façons de mettre à jour les icônes d'état :

- Appuyez sur la touche de fonction F5.
- Choisissez la commande **Affichage > Actualiser tout** dans la fenêtre.
- Cliquez sur le bouton "Actualiser l'affichage".

11.6 La vue 'Diagnostic HW'

La vue "Diagnostic HW" présente la configuration de station en cours sous deux formes :

- la vue graphique montre la disposition réelle des modules à leurs emplacements d'enchâssement,
- le tableau donne des informations détaillées sur les modules enchâssés (par ex. adresses et numéros de référence).

Zone de la vue 'Diagnostic HW'	Signification																																		
<div></div> <div><p>Mnémoniques</p><p> MARCHE</p><p> ARRET</p><p> ATTENTE</p><p> Erreur</p><p>Actualiser</p></div>	<p>Vue graphique de la configuration matérielle, un emplacement ou un module y est sélectionné.</p> <p>Une flèche relie le module sélectionné à l'emplacement de même valeur dans le tableau de la configuration matérielle. L'état des modules du système cible est indiqué par des icônes.</p> <p>Vous pouvez actualiser l'état des icônes au moyen du bouton "Actualiser".</p>																																		
<table><tr><th>Châssis 0</th><th>Châssis 1</th><th>Châssis 2</th><th>Châssis 3</th></tr><tr><th>Empl</th><th>Etat du modul</th><th>Module</th><th>Référence</th><th>Adresse de base</th></tr><tr><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td> Erreur</td><td>CPU 314</td><td>6ES7 314-1AE8...</td><td></td></tr><tr><td>3</td><td></td><td></td><td></td><td></td></tr><tr><td>4</td><td>O.K.</td><td>SM 32* DI...</td><td>6ES7 32* stand...</td><td>E 0</td></tr><tr><td>5</td><td>O.K.</td><td>SM 32* DO</td><td>6ES7 32* stand...</td><td>S 4</td></tr></table>	Châssis 0	Châssis 1	Châssis 2	Châssis 3	Empl	Etat du modul	Module	Référence	Adresse de base	1					2	 Erreur	CPU 314	6ES7 314-1AE8...		3					4	O.K.	SM 32* DI...	6ES7 32* stand...	E 0	5	O.K.	SM 32* DO	6ES7 32* stand...	S 4	<p>Emplacement / module sélectionné dans le tableau de la configuration matérielle.</p> <p>L'état des modules est indiqué dans la colonne "Etat du module".</p> <p>Vous pouvez accéder à différents profils support au moyen des onglets placés au bord supérieur.</p>
Châssis 0	Châssis 1	Châssis 2	Châssis 3																																
Empl	Etat du modul	Module	Référence	Adresse de base																															
1																																			
2	 Erreur	CPU 314	6ES7 314-1AE8...																																
3																																			
4	O.K.	SM 32* DI...	6ES7 32* stand...	E 0																															
5	O.K.	SM 32* DO	6ES7 32* stand...	S 4																															
<div><p>Module : CPU 314 ID système : SIMATIC</p><p>Version :</p><table><tr><th>Référence/Désignation</th><th>Composant</th><th>Version</th></tr><tr><td>6ES7 314-1AE84-0AB0</td><td>Matériel</td><td>1</td></tr><tr><td>---</td><td>Firmware</td><td>V1.0.0</td></tr></table><p>Châssis : 0 Adresse : ---</p><p>Emplacement 2 Largeur module : 1</p><p>Etat : Module défectueux</p><p>Informations de diagnostic étendues...</p></div>	Référence/Désignation	Composant	Version	6ES7 314-1AE84-0AB0	Matériel	1	---	Firmware	V1.0.0	<p>Informations détaillées sur le module sélectionné dans la vue graphique.</p> <p>Le bouton "Informations de diagnostic étendues" vous permet d'ouvrir la boîte de dialogue de l'état du module, par exemple pour lire la mémoire tampon de diagnostic.</p>																									
Référence/Désignation	Composant	Version																																	
6ES7 314-1AE84-0AB0	Matériel	1																																	
---	Firmware	V1.0.0																																	

11.7 L'état du module

11.7.1 Appel de l'état du module

Appel de l'état du module depuis le panneau de commande de la CPU

Quelle que soit la vue sélectionnée, l'état du module est visible sur le panneau de commande de la CPU.

Appel de l'état du module depuis la vue "Diagnostic du matériel"

1. Dans la fenêtre du projet, cliquez deux fois sur l'icône "Matériel".
2. Sélectionnez l'onglet "Diagnostic HW".
3. Sélectionnez un module indiqué comme défectueux.
4. Choisissez la commande **Outils > Etat du module** ou cliquez sur le bouton "Informations de diagnostic étendues".

Résultat

La boîte de dialogue "Etat du module" s'affiche pour les CPU et les modules aptes au diagnostic. Suivant les facultés de diagnostic du module en question, elle présente un nombre d'onglets plus ou moins grand. En cas de module non apte au diagnostic, vous exploitez les informations d'état à partir de la vue 'Diagnostic HW'.

Exemple : état d'une CPU

Pour une CPU, la boîte de dialogue de l'état du module affiche dans sa partie supérieure les informations suivantes :

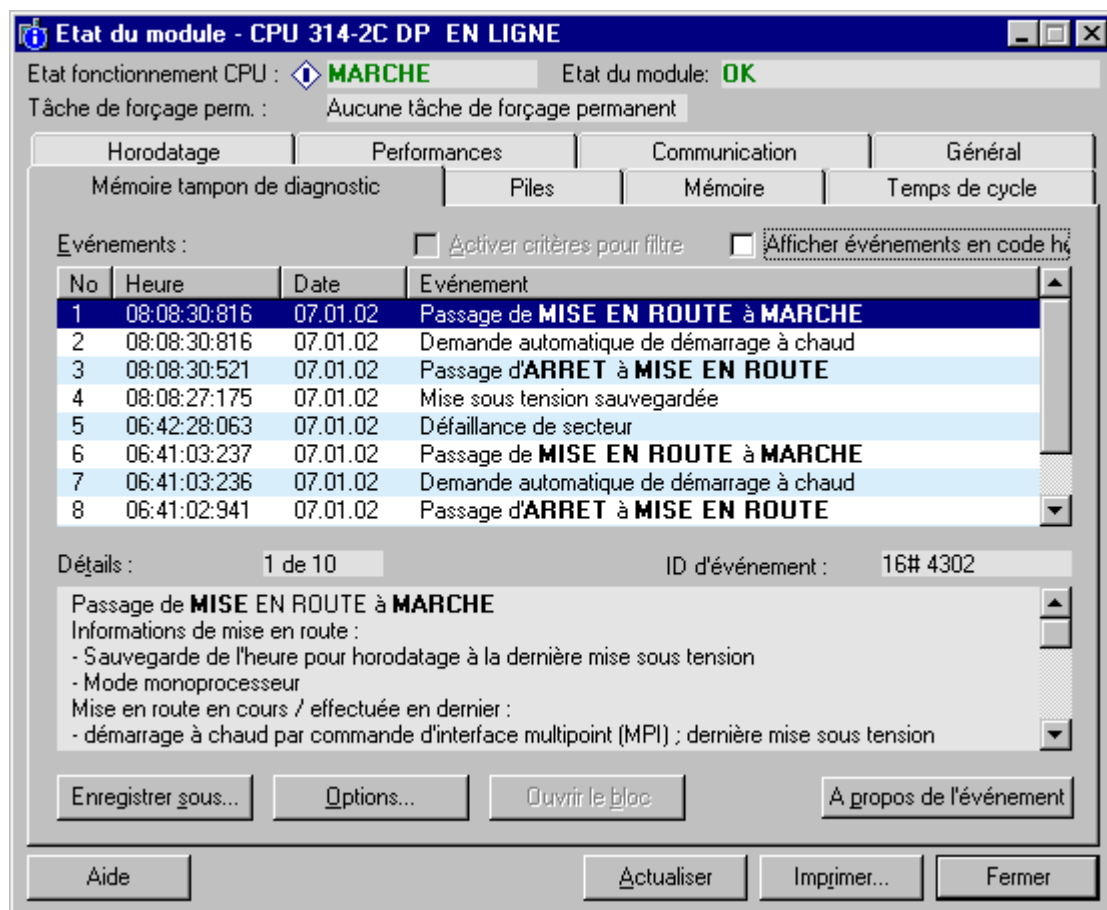
- état de fonctionnement de la CPU (par ex. RUN),
- état du module (par ex. s'il y a une erreur),
- tâche de forçage permanent (active ou inactive).

Vous trouvez en dessous les onglets de l'état du module, l'onglet "Tampon de diagnostic" étant ici au premier plan. Le nombre d'onglets diffère selon le type du module sélectionné, pour lequel l'état du module a été appelé.

L'onglet "Tampon de diagnostic" énumère les entrées (=événements) par ordre chronologique. L'entrée la plus récente se trouve toujours en haut (entrée n° 1). Les détails sur l'événement sélectionné sont dans le champ au-dessous.

Des cases à cocher servent à déterminer les paramètres pour la vue du dialogue (filtre et affichage codé hexa des événements).

Les boutons permettent d'**Enregistrer** au format texte (*.txt), de modifier les **Paramètres** (par ex. pour le filtre) et d'afficher l'**Aide** sur l'événement sélectionné, caractérisé par l'ID d'événement codée hexa. Quand l'événement renvoie à un bloc, et dans ce cas seulement, le bouton **Ouvrir le bloc** permet d'ouvrir directement ce bloc pour traitement ultérieur.



11.7.2 Fonctions d'information de l'état du module

Les fonctions d'informations sont disponibles dans la page d'onglet de même nom dans la boîte de dialogue "Etat du module". Dans chaque cas d'utilisation concret, seules les pages d'onglet significatives pour le module sélectionné s'affichent.

Fonction d'information	Informations	Emploi
Général	Données d'identification du module sélectionné, par ex. type, n° de réf., version, état, emplacement dans le châssis/profilé support.	Pour comparer les informations en ligne du module enfiché aux données du module configuré.
Mémoire tampon de diagnostic	Vue d'ensemble des événements dans la mémoire tampon de diagnostic et informations détaillées sur l'événement sélectionné.	Pour évaluer la cause du passage à l'état "Arrêt" d'une CPU et pour évaluer les événements précédents sur le module sélectionné. Grâce à la mémoire tampon de diagnostic, les erreurs dans le système peuvent être évaluées, même bien plus tard, afin de déterminer la cause d'un passage à l'état "Arrêt" ou de remonter la trace des événements de diagnostic individuels.
Alarme de diagnostic	Données de diagnostic du module sélectionné.	Pour déterminer la cause d'un défaut de module.
Mémoire	Organisation de la mémoire, charge actuelle de la mémoire de travail et de la mémoire de chargement de la CPU sélectionnée.	Pour vérifier que la mémoire de chargement d'une CPU est suffisante avant d'y transférer de nouveaux blocs ou des blocs étendus ainsi que pour comprimer le contenu de la mémoire.
Temps de cycle	Durée du cycle le plus long, du cycle le plus court et du dernier cycle de la CPU sélectionnée.	Pour contrôler le temps de cycle minimal paramétré ainsi que les temps de cycle maximal et actuel.
Horodatage	Heure actuelle, nombre d'heures de fonctionnement et informations pour la synchronisation des horloges (intervalles de synchronisation).	Pour afficher l'heure et la date d'un module et contrôler la synchronisation des horloges.
Performances	Plages d'opérandes et blocs disponibles du module sélectionné.	Avant et pendant la création d'un programme utilisateur et pour vérifier que la CPU présente les conditions requises pour l'exécution d'un programme utilisateur, par ex. quant à la taille de la mémoire image.
Performances (suite)	Affichage de tous les types de blocs disponibles dans le module sélectionné. Liste des OB, SFB et SFC pouvant être utilisés dans ce module.	Pour vérifier quels blocs standard votre programme utilisateur peut contenir ou appeler pour pouvoir s'exécuter dans la CPU sélectionnée.
Communication	Vitesses de transmission, liaisons établies, charge due à la communication et taille maximale des télégrammes sur le bus K du module sélectionné.	Pour déterminer combien de liaisons et quelles liaisons de la CPU sont possibles ou affectées.
Piles	Onglet Piles : vous ne pouvez l'ouvrir qu'à l'état "Arrêt" ou "Attente". La pile des blocs (pile B) du module sélectionné s'affiche. Vous pouvez lire en outre la pile des interruptions (pile I) et celle des données locales (pile L) et sauter à la position d'erreur dans le bloc qui a provoqué l'interruption.	Pour trouver la cause d'un passage à l'état "Arrêt" et pour corriger un bloc.

Informations supplémentaires affichées

Les informations suivantes sont toujours représentées :

- si une tâche de forçage permanent est active (seulement pour les CPU qui assument le forçage permanent),
- état de fonctionnement de la CPU concernée (par ex. "Marche", "Arrêt"),
- état du module sélectionné (par ex. Erreur, OK).

Affichage simultané de plusieurs modules

Vous pouvez interroger et afficher simultanément l'état de plusieurs modules. Pour cela, vous devez retourner au contexte de module qui vous intéresse, sélectionner un autre module et en appeler l'état. Une autre boîte de dialogue à onglets vous est alors proposée. Mais vous ne pouvez ouvrir qu'une boîte de dialogue à onglets par module.

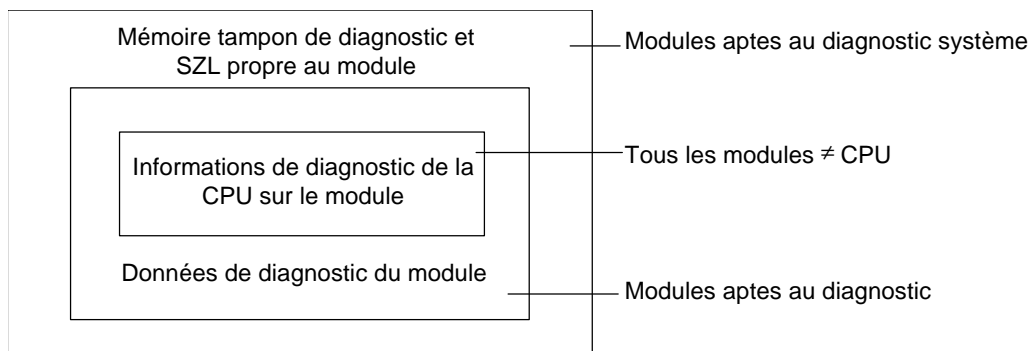
Actualisation de l'affichage de l'état du module

Chaque fois que vous changez d'onglet dans la boîte de dialogue "Etat du module", les données sont lues à nouveau dans le module. En revanche, le contenu des pages d'onglet n'est pas mis à jour automatiquement pendant l'affichage. Pour lire à nouveau les données en cours du module sans changer de page d'onglet, il suffit de cliquer sur le bouton "Actualiser".

11.7.3 Volume d'informations selon le type de module dans l'état du module

Le volume des informations susceptibles d'être évaluées et affichées dépend du module sélectionné.

Selon le volume des informations, on distingue entre modules à diagnostic système, modules à diagnostic ou modules sans diagnostic. C'est ce qu'illustre le schéma ci-dessous.



- Les modules complexes, tels que les FM351 et FM354, par exemple, possèdent des fonctions de diagnostic système. Ces modules possèdent une mémoire tampon de diagnostic et gère de manière interne une liste d'état système (SZL).
- Les modules à diagnostic sont ceux qui peuvent déclencher une alarme de processus ; il s'agit généralement de modules analogiques.
- Les modules sans diagnostic sont ceux qui ne peuvent pas déclencher une alarme de processus ; il s'agit généralement de modules TOR.

Onglets affichés

Le tableau précise quels onglets s'affichent dans la boîte de dialogue "Etat du module" pour les différents types de modules.

Onglet	CPU	Module à diagnostic système	Module à diagnostic	Module sans fonction de diagnostic
Général	oui	oui	oui	—
Mémoire tampon de diagnostic	oui	oui	—	—
Alarme de diagnostic	—	oui	oui	—
Mémoire	oui	—	—	—
Temps de cycle	oui	—	—	—
Horodatage	oui	—	—	—
Performances	oui	—	—	—
Piles	oui	—	—	—
Communication	oui	—	—	—

Outre les informations contenues dans les onglets, l'état de fonctionnement est affiché pour les modules qui en possèdent un. Quand vous interrogez l'état du module, c'est de plus l'état du point de vue de la CPU qui est indiqué (par ex. ok, erreur, module inexistant).

11.8 Diagnostic à l'état de fonctionnement STOP

11.8.1 Marche à suivre pour déterminer la cause d'un passage à l'état d'arrêt

Pour déterminer la cause d'un passage à l'arrêt de la CPU, procédez de la manière suivante :

1. Dans la vue "Diagnostic HW", sélectionnez la CPU qui est passée à l'état d'arrêt.
2. Cliquez sur le bouton "Informations de diagnostic étendues".
3. Sélectionnez l'onglet "Tampon de diagnostic".
Les dernières entrées vous permettent de déterminer la cause du passage à l'arrêt.

Exemple : erreur de programmation

L'entrée "Arrêt car OB d'erreur de programmation non chargé", par exemple, signifie que la CPU a détecté une erreur de programmation, puis a tenté de démarrer l'OB (manquant) destiné à traiter cette erreur. L'erreur de programmation est indiquée par l'entrée précédente.

1. Sélectionnez le message concernant l'erreur de programmation.
2. Cliquez sur le bouton "Ouvrir le bloc".
3. Sélectionnez la page d'onglet "Piles".

11.8.2 Contenu des piles à l'état d'arrêt

L'exploitation de la mémoire de diagnostic et du contenu des piles vous permet de déterminer la cause d'un défaut dans l'exécution d'un programme utilisateur.

Lorsque la CPU est passée à l'état d'arrêt, par exemple suite à une erreur de programmation ou à la commande d'arrêt, la pile des blocs s'affiche dans la page d'onglet "Piles" de l'état du module. Vous pouvez afficher d'autres contenus de piles grâce aux boutons "Pile des interruptions" et "Pile des données locales". Le contenu des piles vous indique quelle opération dans quel bloc a entraîné le passage à l'état "Arrêt" (STOP) de la CPU.

Contenu de la pile des blocs

La pile des blocs donne la liste de tous les blocs appelés avant le passage à l'état de fonctionnement "Arrêt" (STOP) et qui n'ont pas encore été exécutés jusqu'à la fin.

Contenu de la pile des interruptions

Un clic sur le bouton "Pile I" vous renseigne sur la position d'interruption. La pile des interruptions contient les données et les états en vigueur au moment de l'interruption, par exemple :

- contenu des accumulateurs et des registres,
- DB ouverts et leur taille,
- contenu du mot d'état,
- classe de priorité,
- bloc interrompu,
- bloc dans lequel l'exécution du programme a été poursuivie après l'interruption.

Contenu de la pile des données locales

Pour chacun des blocs énumérés dans la pile B, vous pouvez afficher les données locales correspondantes en sélectionnant le bloc et en cliquant sur le bouton "Pile L".

La pile des données locales (pile L) contient les valeurs des données locales des blocs que le programme utilisateur a utilisés jusqu'à l'interruption.

L'interprétation et l'exploitation des données locales affichées demandent de très bonnes connaissances du système. La partie avant des données affichées correspond aux variables temporaires du bloc.

11.8.3 Ouverture du bloc pour une entrée en mémoire tampon de diagnostic ou dans la pile

11.8.3.1 Ouverture du bloc pour une entrée en mémoire tampon de diagnostic

Pour les entrées en mémoire tampon de diagnostic faisant référence à la position d'une erreur (type et numéro du bloc, adresse relative), vous pouvez ouvrir le bloc ayant provoqué l'événement afin d'éliminer la cause de l'erreur.

1. Sélectionnez l'événement de diagnostic dans la liste du haut.
2. Cliquez sur le bouton "Ouvrir le bloc". Le bloc s'ouvre alors dans l'éditeur (par ex. LIST) et le pointeur est positionné à l'endroit du programme qui a causé l'événement.
3. Corrigez l'erreur dans le bloc.

Nota

La mémoire tampon de diagnostic enregistre toutes les entrées de diagnostic jusqu'au nombre maximal autorisé. Ces entrées sont conservées même lorsque vous chargez un autre programme utilisateur entre-temps.

Il est donc possible que d'anciennes entrées de diagnostic se réfèrent à des blocs n'existant plus dans la CPU. Au pire des cas, il peut exister dans la CPU un bloc de même nom, mais n'ayant rien à voir avec le message de diagnostic.

Les situations suivantes sont rarement susceptibles de se produire :

- L'événement de diagnostic est antérieur à la date de la dernière modification du bloc.
 - La boîte de dialogue "Ouvrir le bloc" s'affiche en vous avertissant que le bloc a été modifié entre temps. Il peut s'agir aussi d'un bloc de même nom, mais faisant partie d'un autre programme.
 - Vous pouvez sélectionner le bloc dans le programme correspondant et modifier hors ligne
 - Le bloc à l'origine de l'événement n'est plus dans la CPU.
 - La boîte de dialogue "Ouvrir le bloc" s'affiche en vous avertissant que le bloc référencé n'existe pas dans la CPU. Il a été effacé après enregistrement de l'entrée de diagnostic.
 - Vous pouvez le sélectionner hors ligne dans le programme correspondant et le modifier également hors ligne.
-

11.8.3.2 Ouverture du bloc dans la liste de la pile B

Procédez de la manière suivante :

4. Cliquez sur le bouton "Ouvrir le bloc". Ce dernier s'ouvre en ligne. Le curseur est positionné sur l'endroit où l'exécution du programme serait reprise après le saut quittant le bloc appelé.
5. Ouvrez le bloc hors ligne (depuis la fenêtre du projet) et effectuez les modifications dans l'éditeur de programme.

11.8.3.3 Ouverture du bloc dans la liste de la pile L

Procédez de la manière suivante :

1. Cliquez sur le bouton "Ouvrir le bloc". Ce dernier s'ouvre en ligne. Le curseur indique la position du programme qui a causé l'erreur.
2. Ouvrez le bloc hors ligne (depuis la fenêtre du projet) et effectuez les modifications dans l'éditeur de programme.

11.9 Contrôle des temps de cycle pour éviter les erreurs de temps

La page d'onglet "Temps de cycle" de l'état du module vous donne des renseignements sur les temps de cycle du programme utilisateur.

Lorsque la durée du cycle le plus long est proche du temps de surveillance, des fluctuations du temps de cycle risquent de provoquer une erreur de temps. Vous pouvez éviter cela en augmentant le temps de cycle maximal du programme utilisateur.

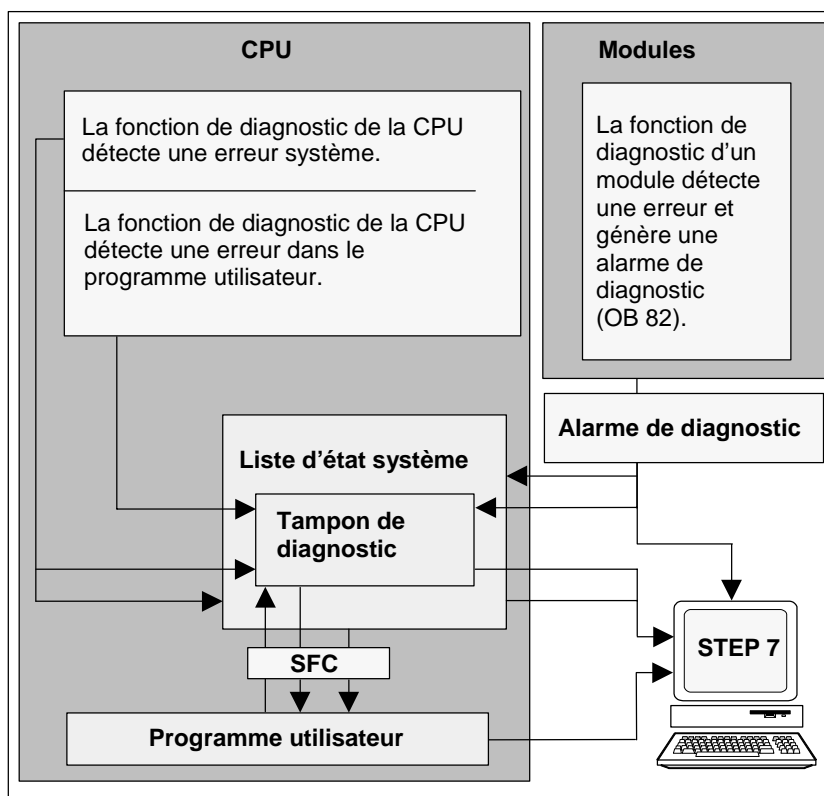
Définition du temps de cycle

Vous pouvez définir les temps de cycle minimal et maximal lors de la configuration du matériel. A cet effet, sélectionnez la CPU dans la vue "Configuration HW" et choisissez la commande **Paramètres de module** à l'aide du bouton droit de la souris. Vous pouvez saisir vos valeurs dans la rubrique "Cycle".

11.10 Transmission d'informations de diagnostic

11.10.1 Transmission d'informations de diagnostic

La figure suivante montre comment les informations de diagnostic sont transmises dans SIMATIC S7.



Lecture des informations de diagnostic

Vous pouvez lire les entrées de diagnostic dans le programme utilisateur à l'aide de la SFC51 RDSYSST ou afficher les messages de diagnostic en clair avec STEP 7 Lite.

Ces informations précisent :

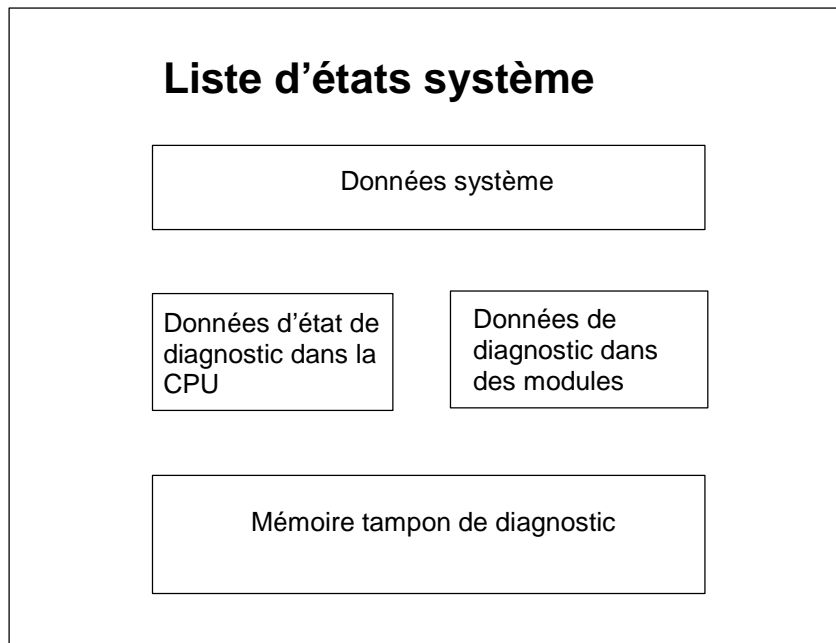
- où et quand l'erreur est apparue,
- à quel type d'événement de diagnostic appartient l'entrée (événement de diagnostic personnalisé, erreur synchrone ou asynchrone, changement d'état de fonctionnement).

11.10.2 Liste d'états système (SZL)

La liste d'état système (SZL) décrit l'état en cours de l'automate programmable : elle donne une vue d'ensemble de la configuration, du paramétrage en vigueur, des états et exécutions en cours dans la CPU et les modules associés.

Vous pouvez seulement lire les données de la liste d'état système, et non les modifier. Il s'agit en fait d'une liste virtuelle, générée uniquement sur demande.

On peut subdiviser les informations contenues dans la liste d'états système en quatre zones.



Lecture de la SZL

Il existe deux méthodes pour lire les informations de la liste d'état système :

- implicitement à partir de la console de programmation via des commandes de STEP 7 Lite (par exemple, étendue de la mémoire, données de CPU statiques, mémoire tampon de diagnostic, indications d'état),
- explicitement à partir du programme utilisateur via la fonction système SFC51 RDSYSST par indication du numéro de liste partielle souhaitée (voir aide sur les blocs).

Données système de la liste SZL

Les données système sont des caractéristiques fixes ou paramétrées d'une CPU. Le tableau suivant montre pour quels thèmes il est possible d'obtenir des informations (listes partielles de la SZL).

Domaine	Informations
Identificateur de module	Numéro de référence, identification de type et version du module
Caractéristiques de la CPU	Système d'horodatage et description de langage de la CPU
Zones de mémoire	Etendue de mémoire du module (taille de la mémoire de travail)
Zones système	Mémoire système du module (par exemple, nombre de mémentos, temporisations et compteurs, type de mémoire)
Types de blocs	Types de blocs (OB, DB, SDB, FC, FB) disponibles dans le module, nombre maximal des blocs d'un type et taille maximale d'un type de bloc
Affectation alarmes/erreurs	Affectation d'alarmes/erreurs aux OB
Etat d'alarme	Traitement et génération d'alarmes en cours
Etat des classes de priorité	OB en cours de traitement, classe de priorité verrouillée par paramétrage
Etat de fonctionnement et changement d'état de fonctionnement	Etats de fonctionnement possibles, dernier changement d'état de fonctionnement, état de fonctionnement en vigueur

Données d'état de diagnostic dans la CPU

Les données d'état de diagnostic décrivent l'état en vigueur des composants surveillés par le diagnostic système. Le tableau suivant montre pour quels thèmes il est possible d'obtenir des informations (listes partielles de la SZL).

Domaine	Informations
Données d'état de la communication	Fonctions de communication actuellement activées dans le système
Partenaires de diagnostic	Modules aptes au diagnostic déclarés à la CPU
Liste des informations de déclenchement des OB	Informations de déclenchement pour les OB de la CPU
Liste d'événements déclencheurs	Evénements déclencheurs et classes de priorité des OB
Informations d'état des modules	Informations d'état de tous les modules affectés, générant des alarmes de processus, défectueux et enfichés.

Données de diagnostic des modules

Il existe, outre la CPU, d'autres modules aptes au diagnostic (SM, CP, FM) dont les données de diagnostic sont inscrites dans la liste d'état système. Le tableau suivant montre pour quels thèmes il est possible d'obtenir des informations (listes partielles de la SZL).

Domaine	Informations
Informations de diagnostic de module	Adresse de début de module, erreurs internes/externes, erreurs de voie, erreurs de paramètres (4 octets)
Données de diagnostic de module	Toutes les données de diagnostic d'un module précis

11.10.3 Envoi de vos propres messages de diagnostic

Vous pouvez, en outre, étendre le diagnostic système standard de SIMATIC S7 à l'aide de la fonction système SFC52 WR_USMSG :

- en inscrivant vos propres informations de diagnostic (par exemple, informations sur l'exécution du programme utilisateur) dans la mémoire tampon de diagnostic,
- en envoyant des messages de diagnostic que vous avez définis à des participants déclarés (appareils de contrôle comme PG, OP, TD).

Événements de diagnostic personnalisé

Les événements de diagnostic sont répartis en classes d'événement 1 à F. Ceux que vous définissez vous-même appartiennent aux classes d'événement 8 à B. On peut les subdiviser en deux groupes :

- Les classes d'événement 8 et 9 comprennent les événements avec un numéro défini et un texte préparé que vous pouvez appeler via le numéro.
- Les classes d'événement A et B regroupent les événements avec numéro (A000 à A0FF, B000 à B0FF) et texte libres.

Envoi de messages de diagnostic à des participants

En plus d'inscrire un événement de diagnostic personnalisé dans la mémoire tampon de diagnostic, vous pouvez, à l'aide de la SFC52 WR_USMSG, envoyer le message correspondant à des visuels déclarés. A l'appel de la SFC52 avec SEND = 1, le message de diagnostic est écrit dans la mémoire tampon d'émission et automatiquement envoyé aux participants déclarés à la CPU.

Si l'envoi du message de diagnostic s'avère impossible - par exemple parce qu'aucun participant n'a été déclaré ou que la mémoire tampon d'émission est pleine -, l'événement de diagnostic personnalisé est quand même inscrit dans la mémoire tampon de diagnostic.

Création de messages avec indication d'acquittement

Procédez comme suit si vous voulez acquitter un événement de diagnostic personnalisé et enregistrer cet acquittement par programme :

- Ecrivez un 1 dans une variable de type BOOL pour un événement entrant ; écrivez 0 pour un événement sortant.
- Surveillez cette variable à l'aide du bloc SFB33 ALARM.

11.11 Mesures à prendre dans le programme pour traiter les erreurs

Lorsque la CPU détecte des erreurs dans l'exécution du programme (erreurs synchrones) ou des erreurs dans l'automate programmable (erreurs asynchrones), elle appelle l'OB d'erreur correspondant à l'erreur respective :

Erreur survenue	OB d'erreur
Erreur de temps	OB80
Erreur d'alimentation	OB81
Alarme de diagnostic	OB82
Erreur matérielle CPU	OB84
Erreur d'exécution du programme	OB85
Défaillance d'un châssis ou d'une station en périphérie décentralisée	OB86
Erreur de communication	OB87
Erreur de programmation	OB121
Erreur d'accès à la périphérie	OB122

S'il n'y a pas d'OB correspondant, la CPU passe à l'état de fonctionnement "Arrêt" (STOP). S'il y en a un, vous avez pu y stocker des instructions sur la manière de réagir à cette situation d'erreur, ce qui permet éventuellement de minimiser ou de supprimer les conséquences de l'erreur.

Marche à suivre

Création et ouverture de l'OB

1. Appelez l'état de fonctionnement de votre CPU.
2. Sélectionnez l'onglet "Performances".
3. Vérifiez dans la liste affichée que l'OB à programmer est autorisé pour cette CPU.
4. Insérez l'OB dans le dossier "Blocs" de votre programme et ouvrez-le.
5. Saisissez le programme de traitement de l'erreur.
6. Chargez l'OB dans le système cible.

Programmation des mesures de traitement d'erreur

1. Evaluer les données locales de l'OB pour déterminer avec précision la cause d'erreur. Les variables OB8x_FLT_ID ou OB12x_SW_FLT des données locales contiennent le code d'erreur. Leur signification est décrite dans le manuel de référence des fonctions système et des fonctions standard.
2. Allez dans la branche de programme qui réagit à cette erreur.

Un exemple de traitement d'alarmes de diagnostic est fourni dans l'aide de référence des fonctions système et fonctions standard sous le titre "Exemple de diagnostic de module avec le bloc SFC51 (RDSYSST)".

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.1 Exploitation du paramètre de sortie RET_VAL

Une fonction système signale par le paramètre de sortie RET_VAL (valeur en retour) si la CPU a pu l'exécuter avec ou sans erreur.

Informations d'erreur dans la valeur en retour

La valeur en retour est de type de données INT (nombre entier), le signe précisant s'il s'agit d'un entier positif ou négatif. La relation de la valeur en retour à la valeur 0 indique si une erreur s'est produite pendant le traitement de la fonction (voir aussi le tableau ci-dessous) :

- Si une erreur apparaît pendant le traitement de la fonction, la valeur en retour est inférieure à 0. Le bit de signe du nombre entier est à 1.
- Si la fonction est traitée sans erreur, la valeur en retour est supérieure ou égale à 0. Le bit de signe du nombre entier est à 0.

Traitement de la SFC par la CPU	Valeur en retour	Signe du nombre entier
Avec erreur	Inférieure à 0	Négatif (bit de signe à 1)
Sans erreur	Supérieure ou égale à 0	Positif (bit de signe à 0)

Réaction aux informations d'erreur

Quand une erreur apparaît au cours de l'exécution d'une fonction système, la SFC renvoie un code d'erreur au moyen de la valeur en retour RET_VAL.

Il faut distinguer entre :

- un code d'erreur général que toutes les SFC peuvent émettre et
- un code d'erreur spécifique qu'une SFC peut émettre selon ses fonctions spécifiques.

Transmission de la valeur de la fonction

Certaines fonctions système utilisent également le paramètre de sortie RET_VAL pour renvoyer leur résultat. La SFC64 TIME_TCK, par exemple, renvoie l'heure système lue via RET_VAL.

Vous trouverez des informations détaillées sur le paramètre RET_VAL dans l'aide sur les SFB/SFC.

11.11.2 OB d'erreur en réaction à la détection d'une erreur

Erreurs détectables

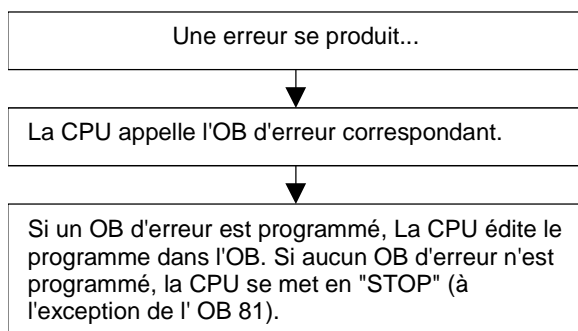
Le programme système peut détecter les erreurs suivantes :

- fonctionnement erroné de la CPU
- erreurs dans le traitement du programme système
- erreurs dans le programme utilisateur
- erreurs dans la périphérie

Selon la nature de l'erreur, la CPU passe à l'état de fonctionnement "Arrêt" (STOP) ou elle appelle un OB d'erreur.

Programmation de réactions

Vous pouvez concevoir des programmes pour réagir aux différentes sortes d'erreur et déterminer le comportement de la CPU. Vous pouvez ensuite enregistrer dans un OB d'erreur le programme écrit pour une erreur donnée. Ainsi, il sera exécuté à l'appel de cet OB d'erreur.



OB d'erreur

On distingue entre erreurs synchrones et asynchrones :

- Les erreurs synchrones peuvent être associées à une opération MC7 (par exemple, opération de chargement pour un module de signaux retiré).
- Les erreurs asynchrones peuvent être attribuées à une classe de priorité ou à l'automate programmable entier (par exemple, dépassement du temps de cycle).

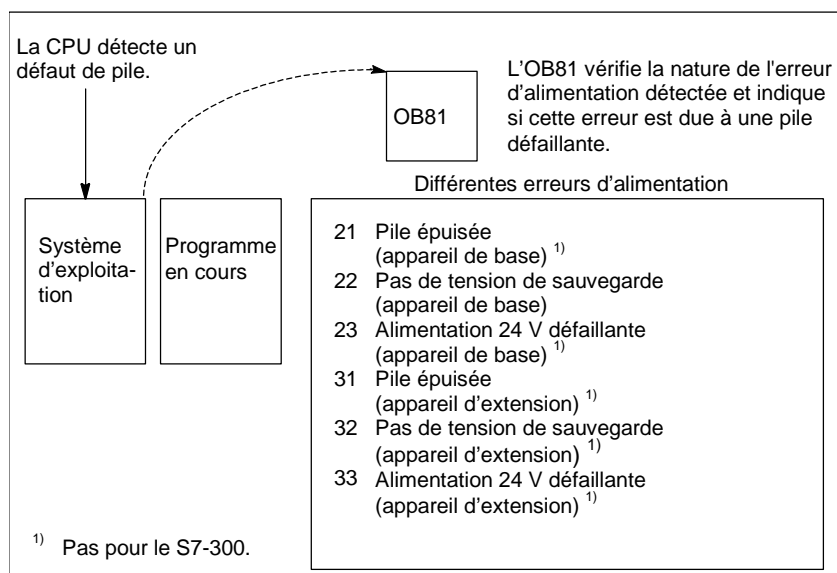
Le tableau ci-après présente les sortes d'erreur pouvant en principe apparaître. Les OB offerts par les différentes CPU sont indiqués dans le manuel *Système d'automatisation S7-300, Installation et configuration – Caractéristiques des CPU*.

Type d'erreur	Sorte d'erreur	OB	Priorité
Asynchrone	Erreur de temps	OB 80	26
	Erreur d'alimentation	OB 81	((ou 28 lorsque l'OB d'erreur est appelé dans le programme de mise en route)
	Alarme de diagnostic	OB 82	
	Erreur matérielle CPU	OB 84	
	Erreur d'exécution du programme	OB 85	
	Défaillance d'unité	OB 86	
	Erreur de communication	OB 87	
Synchrone	Erreur de programmation	OB 121	Priorité de l'OB à l'origine de l'erreur
	Erreur d'accès	OB 122	

Exemple d'utilisation de l'OB d'erreur 81

Les données locales (informations de déclenchement) de l'OB d'erreur vous permettent d'évaluer la nature de l'erreur apparue.

Quand la CPU détecte un défaut de pile, par exemple, le système d'exploitation appelle l'OB81 (voir figure).



Vous pouvez écrire un programme qui évalue l'ID de l'événement ayant déclenché l'appel de l'OB81. Vous pouvez également écrire un programme de réaction comme, par exemple, l'activation d'une sortie reliée à un voyant lumineux du poste opérateur.

Données locales de l'OB d'erreur 81

Le tableau suivant décrit les variables temporaires (TEMP) inscrites dans la table de déclaration des variables de l'OB81.

Il faut aussi que le mnémonique Defaut_pile (BOOL) soit caractérisé comme sortie (par ex. A 4.0) dans la table des mnémoniques, afin que d'autres parties du programme puissent accéder à ces données.

Décl.	Nom	Type	Description
TEMP	OB81_EV_CLASS	BYTE	Classe et code d'événement 39xx
TEMP	OB81_FLT_ID	BYTE	Code d'erreur: b#16#22 = La tension de sauvegarde manque dans l'appareil de base.
TEMP	OB81_PRIORITY	BYTE	Classe de priorité = 26/28
TEMP	OB81_OB_NUMBR	BYTE	81 = OB 81
TEMP	OB81_RESERVED_1	BYTE	Réservé
TEMP	OB81_RESERVED_2	BYTE	Réservé
TEMP	OB81_MDL_ADDR	INT	Réservé
TEMP	OB81_RESERVED_3	BYTE	Significatif uniquement pour les codes d'erreur B#16#31, B#16#32, B#16#33
TEMP	OB81_RESERVED_4	BYTE	
TEMP	OB81_RESERVED_5	BYTE	
TEMP	OB81_RESERVED_6	BYTE	
TEMP	OB81_DATE_TIME	DATE_AND_TIME	Date et heure de déclenchement de l'OB

Exemple de programme pour l'OB d'erreur 81

L'exemple de programme LIST ci-dessous montre comment lire le code d'erreur dans l'OB81.

Ce programme est construit comme suit :

- Le code d'erreur figurant dans l'OB81 (OB81_FLT_ID) est lu et comparé à l'ID de l'événement "Pile épuisée" (B#16#3921).
- Si ce code d'erreur correspond à l'ID pour "Pile épuisée", le programme saute au repère "DP" et active la sortie Defaut_pile..
- Si ce code d'erreur est différent de l'ID pour "Pile épuisée", le programme le compare à l'ID pour "Pas de tension de sauvegarde".
- Si le code d'erreur correspond à l'ID pour "Pas de tension de sauvegarde", le programme saute au repère "DP" et active la sortie Defaut_pile. Sinon, le bloc prend fin.

LIST		Description
L	B#16#21	// Comparer l'ID de l'événement // "Pile épuisée" (B#16#21)
L	#OB81_FLT_ID	// au code d'erreur de l'OB81.
==I		// Si identiques (pile vide), // sauter à DP.
SPB DP		
L	B#16#22	// Comparer l'ID de l'événement // "Pas de tension de sauvegarde" (B#16#22)
==I		// au code d'erreur de l'OB81.
SPB DP		// Si identiques, sauter à DP.
BEA		// Pas de message sur défaut de pile
DP: L	B#16#39	// Comparer le code pour événement apparaissant
L	#OB81_EV_CLASS	// au code d'erreur de l'OB81.
==I		// Si défaillance de tension de sauvegarde // ou de pile détectée,
S	Defaut_pile	// mettre à 1 Defaut_pile // (variable de la table des mnémoniques).
L	B#16#38	// Comparer le code pour événement // disparaissant
==I		// au code d'erreur de l'OB81.
R	Defaut_pile	// Remettre à 0 Defaut_pile quand éliminée.

Vous trouverez des informations détaillées sur les OB, SFB et SFC ainsi que l'explication des ID d'événement dans les aides sur les blocs correspondantes.

11.11.3 Insertion de valeurs de remplacement en cas d'erreur

Pour certaines erreurs (par ex. rupture de fil pour signal d'entrée), vous pouvez indiquer des valeurs remplaçant celles qui ne sont pas disponibles en raison de l'erreur. Il y a deux façons d'indiquer des valeurs de remplacement :

- Vous pouvez paramétrer avec STEP 7 Lite des valeurs de remplacement pour les modules de sorties paramétrables. Les modules de sorties non paramétrables ont la valeur de remplacement prédéfinie 0.
- Vous pouvez programmer des valeurs de remplacement dans les OB d'erreur à l'aide de la SFC44 RPL_VAL (uniquement pour les modules d'entrées).

Pour toutes les opérations de chargement entraînant des erreurs synchrones, vous pouvez indiquer une valeur de remplacement pour le contenu de l'accumulateur dans l'OB d'erreur.

LIST	Description
<pre> L B#16#2942 L #OB122_SW_FLT ==I SPB Qfeh L B#16#2943 <> I SPB Stop </pre>	<p>Comparer le code d'événement de l'OB122 au code d'événement (B#16#2942) pour acquittement d'une erreur de temps lors de la lecture de la périphérie. Si identiques, sauter à "ErrA".</p> <p>Comparer le code d'événement de l'OB122 au code d'événement (B#16#2943) pour erreur d'adressage (écriture depuis un module inexistant). Si différents, sauter à "Stop".</p>
<pre> ErrA: CALL "REPL_VAL" VAL : = DW#16#2912 RET_VAL : = #Erreur L #Erreur L 0 ==I BEB </pre>	<p>Repère "ErrA" : Transmettre DW#16#2912 (10010 binaire) à la SFC44 (REPL_VAL). La SFC44 charge cette valeur dans l'ACCUI, remplaçant ainsi celle qui a causé l'appel de l'OB122. Enregistrer le code d'erreur SFC dans #Erreur.</p> <p>Comparer #Erreur à 0 (si égal, pas d'erreur lors du traitement de l'OB122). Fin du bloc si pas d'erreur.</p>
<pre> Stop: CALL "STP" </pre>	<p>Repère "Stop" : Appeler la SFC46 STP pour mettre la CPU à l'état de fonctionnement "Arrêt".</p>

11.11.4 Erreur de temps (OB80)

Description

Le système d'exploitation de la CPU appelle l'OB80 en cas d'erreur de temps. Les erreurs de temps sont par exemple :

- le dépassement du temps de cycle maximal,
- l'omission d'alarmes horaires parce que l'heure a été avancée,
- un retard excessif dans le traitement d'une classe de priorité.

Programmation de l'OB80

Vous devez créer l'OB80 avec STEP 7 Lite en tant qu'objet dans votre programme. Ecrivez dans le bloc créé le programme à exécuter et chargez-le dans la CPU comme partie de votre programme utilisateur.

L'OB80 peut vous servir, par exemple :

- à exploiter ses informations de déclenchement pour constater quelles alarmes horaires ont été omises,
- à désactiver l'alarme horaire omise à l'aide de la SFC29 CAN_TINT afin qu'elle ne soit pas exécutée et que le traitement des alarmes horaires avec la nouvelle heure réglée parte d'un point net.

Si vous ne désactivez pas des alarmes horaires omises dans l'OB80, la première alarme omise sera traitée et toutes les autres ne seront pas prises en considération.

Si l'OB80 n'est pas programmé, la CPU passe à l'état de fonctionnement "Arrêt" (STOP) quand elle détecte une erreur de temps.

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.5 Erreur d'alimentation (OB81)

Description

Le système d'exploitation de la CPU appelle l'OB81 en cas de défaillance dans l'appareil de base ou dans un appareil d'extension :

- de la tension d'alimentation 24 V,
- d'une pile,
- du système de sauvegarde entier,

ou bien lorsqu'il a été remédié à cette défaillance (appel pour événement entrant et sortant).

Programmation de l'OB81

Vous pouvez créer l'OB81 avec STEP 7 Lite en tant que bloc dans votre programme. Ecrivez dans le bloc créé le programme à exécuter dans l'OB 81.

L'OB81 peut vous servir, par exemple :

- à évaluer ses informations de déclenchement pour constater quelle erreur d'alimentation est apparue,
- à déterminer le numéro du profilé support ou du châssis dont l'alimentation est défectueuse,
- à commander un voyant lumineux situé dans un poste opérateur pour signaler au personnel de maintenance qu'une pile doit être remplacée.

Contrairement à ce qui se passe pour tous les autres OB d'erreur asynchrone, la CPU ne passe pas à l'état "Arrêt" (STOP) quand une erreur d'alimentation est détectée et que l'OB81 n'a pas été programmé. Mais l'erreur est inscrite dans la mémoire tampon de diagnostic et la DEL correspondante en face avant la signale.

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.6 Alarme de diagnostic (OB82)

Description

Le système d'exploitation de la CPU appelle l'OB82 quand un module avec fonction de diagnostic pour lequel vous avez validé l'alarme de diagnostic détecte une erreur et quand cette erreur a été éliminée (appel pour événement entrant et sortant).

Programmation de l'OB82

Vous devez créer l'OB82 avec STEP 7 Lite en tant qu'objet dans votre programme. Ecrivez dans le bloc créé le programme à exécuter et chargez-le dans la CPU comme partie de votre programme utilisateur.

L'OB82 peut vous servir, par exemple :

- à exploiter ses informations de déclenchement,
- à effectuer un diagnostic précis de l'erreur apparue.

Lorsqu'une alarme de diagnostic est déclenchée, le module défectueux inscrit automatiquement 4 octets de données de diagnostic et son adresse de début dans les informations de déclenchement de l'OB d'alarme de diagnostic et dans la mémoire tampon de diagnostic. Vous apprenez ainsi sur quel module et à quel moment s'est produite l'erreur.

Vous pouvez évaluer d'autres données de diagnostic du module défectueux (voie où s'est produite l'erreur, nature de l'erreur) à l'aide d'un programme correspondant dans l'OB82. La SFC51 RDSYSST permet de lire les données de diagnostic du module et la SFC52 WR_USRMSG d'inscrire ces informations dans la mémoire tampon de diagnostic. Vous pouvez, en outre, envoyer à un appareil de contrôle déclaré le message de diagnostic que vous avez défini vous-même.

Si vous n'avez pas programmé l'OB82, la CPU passe à l'état de fonctionnement "Arrêt" (STOP) quand une alarme de diagnostic est déclenchée.

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.7 Erreur matérielle CPU (OB84)

Description

Le système d'exploitation de la CPU appelle l'OB84 quand une erreur est détectée pour l'interface au réseau MPI, au bus de communication ou au coupleur pour la périphérie décentralisée, par exemple un niveau de signal erroné sur la voie, ou quand cette erreur a été éliminée (appel pour événement entrant et sortant).

Programmation de l'OB84

Vous devez créer l'OB84 avec STEP 7 Lite en tant qu'objet dans votre programme. Ecrivez dans le bloc créé le programme à exécuter et chargez-le dans la CPU comme partie de votre programme utilisateur.

L'OB84 peut vous servir, par exemple :

- à exploiter ses informations de déclenchement,
- à envoyer un message à la mémoire tampon de diagnostic à l'aide de la fonction système SFC52 WR_USMSG.

Si l'OB84 n'est pas programmé, la CPU passe à l'état de fonctionnement "Arrêt" (STOP) en cas de détection d'une erreur matérielle CPU.

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.8 Erreur d'exécution du programme (OB85)

Description

Le système d'exploitation de la CPU appelle l'OB85

- en cas d'événement déclencheur d'un OB d'alarme qui ne peut pas être exécuté, car il n'a pas été chargé dans la CPU ;
- en cas d'erreur lors de l'accès au bloc de données d'instance d'un bloc fonctionnel système ;
- en cas d'erreur lors de la mise à jour de la mémoire image du processus (module absent ou défectueux).

Programmation de l'OB85

Vous devez créer l'OB85 avec STEP 7 Lite en tant qu'objet dans votre programme. Ecrivez dans le bloc créé le programme à exécuter et chargez-le dans la CPU comme partie de votre programme utilisateur.

L'OB85 peut vous servir, par exemple :

- à exploiter ses informations de déclenchement pour découvrir quel module est défectueux ou manque (indication de l'adresse de début du module),
- à calculer l'emplacement du module concerné à l'aide de la fonction système SFC49 LGC_GADR.

Si vous ne programmez pas l'OB85, la CPU passera à l'état de fonctionnement "Arrêt" (STOP) en cas d'erreur de classe de priorité.

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.9 Défaillance d'unité (OB86)

Description

Le système d'exploitation de la CPU appelle l'OB86 lorsqu'une défaillance d'unité est détectée, par exemple :

- en cas de défaillance de profilé support ou châssis (IM manquant ou défectueux ou câble de liaison interrompu),
- en cas de coupure de tension décentralisée d'un profilé support ou châssis,
- en cas de défaillance d'un esclave DP dans un réseau maître du système de bus PROFIBUS DP,

ou lorsque cette erreur a été éliminée (appel pour événement entrant et sortant).

Programmation de l'OB86

Vous devez créer l'OB86 avec STEP 7 Lite en tant qu'objet dans votre programme. Ecrivez dans le bloc créé le programme à exécuter et chargez-le dans la CPU comme partie de votre programme utilisateur.

L'OB86 peut vous servir, par exemple :

- à exploiter ses informations de déclenchement pour savoir quelle unité est défaillante ou manque,
- à inscrire un message dans la mémoire tampon de diagnostic et à l'envoyer à un appareil de contrôle à l'aide de la fonction système SFC52 WR_USMSG.

Si vous ne programmez pas l'OB86, la CPU passera à l'état de fonctionnement "Arrêt" (STOP) à la détection d'une défaillance d'unité.

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.10 Erreur de communication (OB87)

Description

Le système d'exploitation de la CPU appelle l'OB87 lorsqu'une erreur de communication apparaît au cours de l'échange de données via blocs fonctionnels de communication ou via communication par données globales, par exemple :

- un identificateur de télégramme erroné a été détecté lors de la réception de données globales,
- le bloc de données pour les informations d'état des données globales manque ou est trop court.

Programmation de l'OB87

Vous devez créer l'OB87 avec STEP 7 Lite en tant qu'objet dans votre programme. Ecrivez dans le bloc créé le programme à exécuter et chargez-le dans la CPU comme partie de votre programme utilisateur.

L'OB87 peut vous servir, par exemple :

- à exploiter ses informations de déclenchement,
- à créer le bloc de données qui fait défaut pour les informations d'état de la communication par données globales.

Si vous ne programmez pas l'OB87, la CPU passera à l'état de fonctionnement "Arrêt" (STOP) à la détection d'une erreur de communication.

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.11 Erreur de programmation (OB121)

Description

Le système d'exploitation de la CPU appelle l'OB121 quand une erreur de programmation apparaît, par exemple :

- des temporisations adressées manquent,
- un bloc appelé n'est pas chargé.

Programmation de l'OB121

Vous devez créer l'OB121 avec STEP 7 Lite en tant qu'objet dans votre programme. Ecrivez dans le bloc créé le programme à exécuter et chargez-le dans la CPU comme partie de votre programme utilisateur.

L'OB121 peut vous servir, par exemple :

- à exploiter ses informations de déclenchement,
- à inscrire la cause de l'erreur dans un bloc de données de signalisation.

Si vous ne programmez pas l'OB121, la CPU passera à l'état de fonctionnement "Arrêt" (STOP) à la détection d'une erreur de programmation.

Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

11.11.12 Erreur d'accès à la périphérie (OB122)

Description

Le système d'exploitation de la CPU appelle l'OB122 lorsqu'une opération STEP 7 Lite accède à une entrée ou à une sortie de module de signaux à laquelle aucun module n'était associé lors du dernier démarrage, par exemple :

- erreur d'accès direct à la périphérie (module défectueux ou manquant),
- accès à une adresse de périphérie inconnue de la CPU.

Programmation de l'OB122

Vous devez créer l'OB122 avec STEP 7 Lite en tant qu'objet dans votre programme. Ecrivez dans le bloc créé le programme à exécuter et chargez-le dans la CPU comme partie de votre programme utilisateur.

L'OB122 peut vous servir, par exemple :

- à exploiter ses informations de déclenchement,
- à appeler la fonction système SFC44 pour attribuer une valeur de remplacement à un module d'entrées afin que l'exécution du programme se poursuive avec une valeur cohérente dépendante du processus.

Si vous ne programmez pas l'OB122, la CPU passera à l'état de fonctionnement "Arrêt" (STOP) à la détection d'une erreur d'accès à la périphérie.

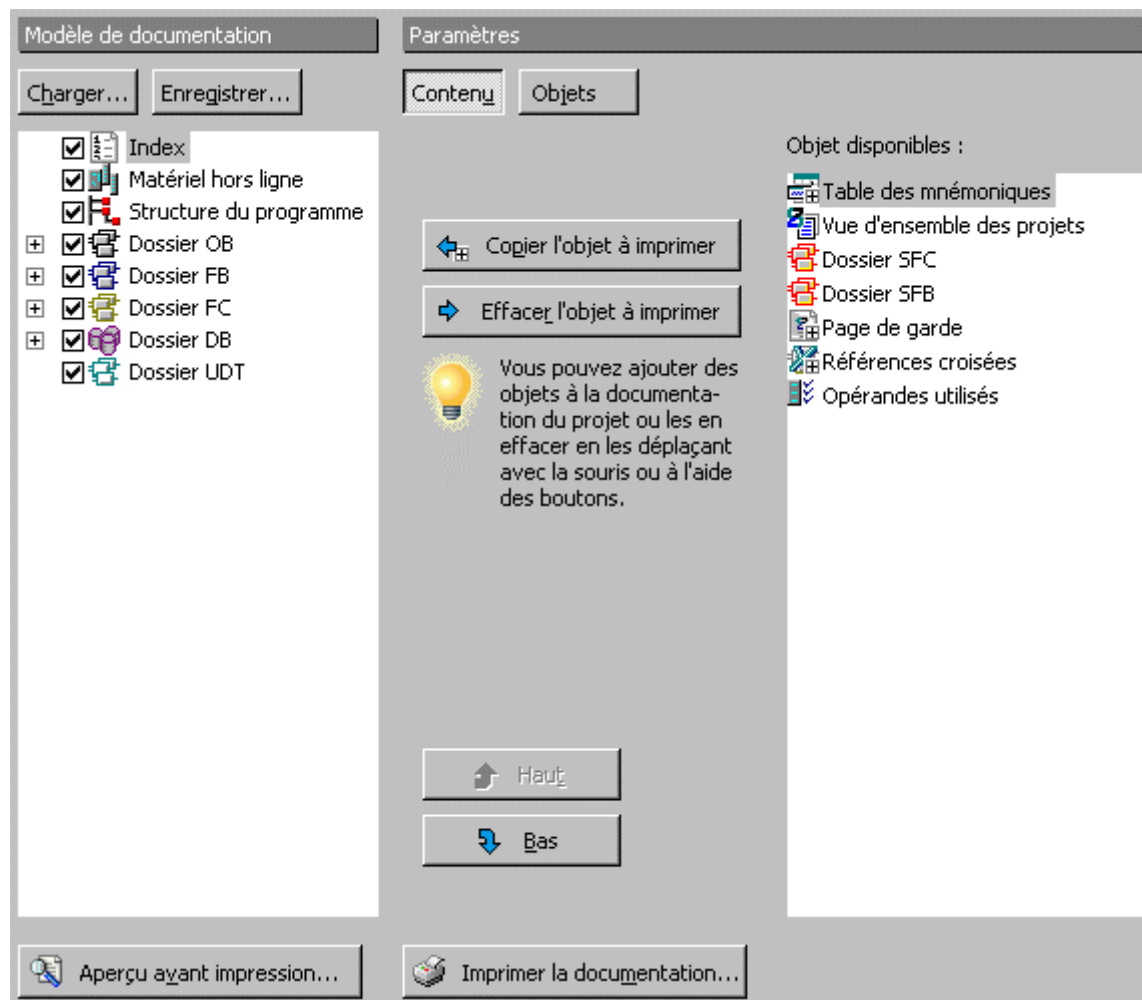
Des informations détaillées sur les OB, SFB et SFC sont données dans les aides sur les blocs correspondantes.

12 Imprimer une documentation du projet

12.1 Documentation du projet

Avec STEP 7 Lite, vous pouvez donner la forme qui vous convient à la documentation de votre projet. La vue utilisée à cet effet s'affiche lorsque vous effectuez un double clic sur l'icône "Documentation du projet".

- Dans une liste des objets à imprimer, vous sélectionnez ceux qui vous intéressent, tels que page de garde, liste des références croisées, table des mnémoniques, dossier Blocs, etc. et vous les disposez dans l'ordre qui vous convient.
- Précisez la représentation des objets individuels (avec mnémoniques ou sans, etc.) dans les "Options".
- Vous pouvez choisir la police, la taille des caractères et le style des caractères pour chaque objet en particulier. Si vous ne souhaitez pas distinguer les objets les uns des autres, utilisez le **modèle de format de police** "par défaut" ou un modèle enregistré auparavant.
- Pour la mise en page, vous pouvez choisir l'orientation "Portrait" ou "Paysage" et concevoir les en-têtes et bas de page pour chaque objet à imprimer. Si vous préférez donner le même aspect à l'ensemble de la documentation, affectez à la mise à page le **modèle de format de mise en page** "par défaut" ou un autre modèle enregistré auparavant.
- Vous pouvez consulter l'"Aperçu avant impression" pour chaque objet à imprimer, afin de vous assurer que le résultat correspond bien à ce que vous attendez.
- Quand tous les paramètres de la documentation vous conviennent, vous pouvez les enregistrer sous forme de **modèle de documentation**, afin de les réutiliser ultérieurement. Vous trouverez pratique de créer plusieurs modèles pour les différents cas d'application, par exemple pour la livraison, la mise en service ou la maintenance.
- Le bouton "Imprimer la documentation" sert à choisir l'imprimante et à la paramétrer avant de lancer l'impression.



12.2 Etablir la documentation du projet

Avec STEP 7 Lite, vous pouvez donner la forme qui vous convient à la documentation de votre projet et déterminer, au moyen d'objets à imprimer, les éléments qui la composent.

Sélectionner les objets à imprimer

Pour établir le contenu de la documentation du projet, procédez de la manière suivante :

1. Ouvrez la vue dans laquelle vous allez établir la documentation du projet en choisissant la commande de menu **Affichage > Documentation du projet**. Vous pouvez également afficher cette vue en effectuant un double clic sur "Documentation du projet" (dans la fenêtre du projet).
Vous voyez à gauche la liste des objets qui composeront votre documentation et à droite celle des objets disponibles (le bouton "Contenu" sous le titre "Paramètres" est enfoncé).
2. Vous n'avez plus qu'à faire glisser les objets qui vous intéressent de la droite vers la gauche et à les mettre dans la liste à la position où vous voulez qu'ils soient imprimés.
Une autre solution consiste à sélectionner l'objet de votre choix dans la liste de droite et à le reprendre dans la liste de gauche en cliquant sur le bouton "Déplacer l'objet à imprimer". Quelques objets (comme la page de garde) peuvent être insérés plusieurs fois dans la documentation de votre projet.

Si vous n'avez pas besoin de certains objets, vous pouvez les désactiver temporairement dans votre liste d'impression en désactivant la case d'option correspondante. Pour supprimer un objet de la liste d'impression, faites-le glisser du côté droit ou sélectionnez-le et cliquez sur le bouton "Effacer l'objet à imprimer".

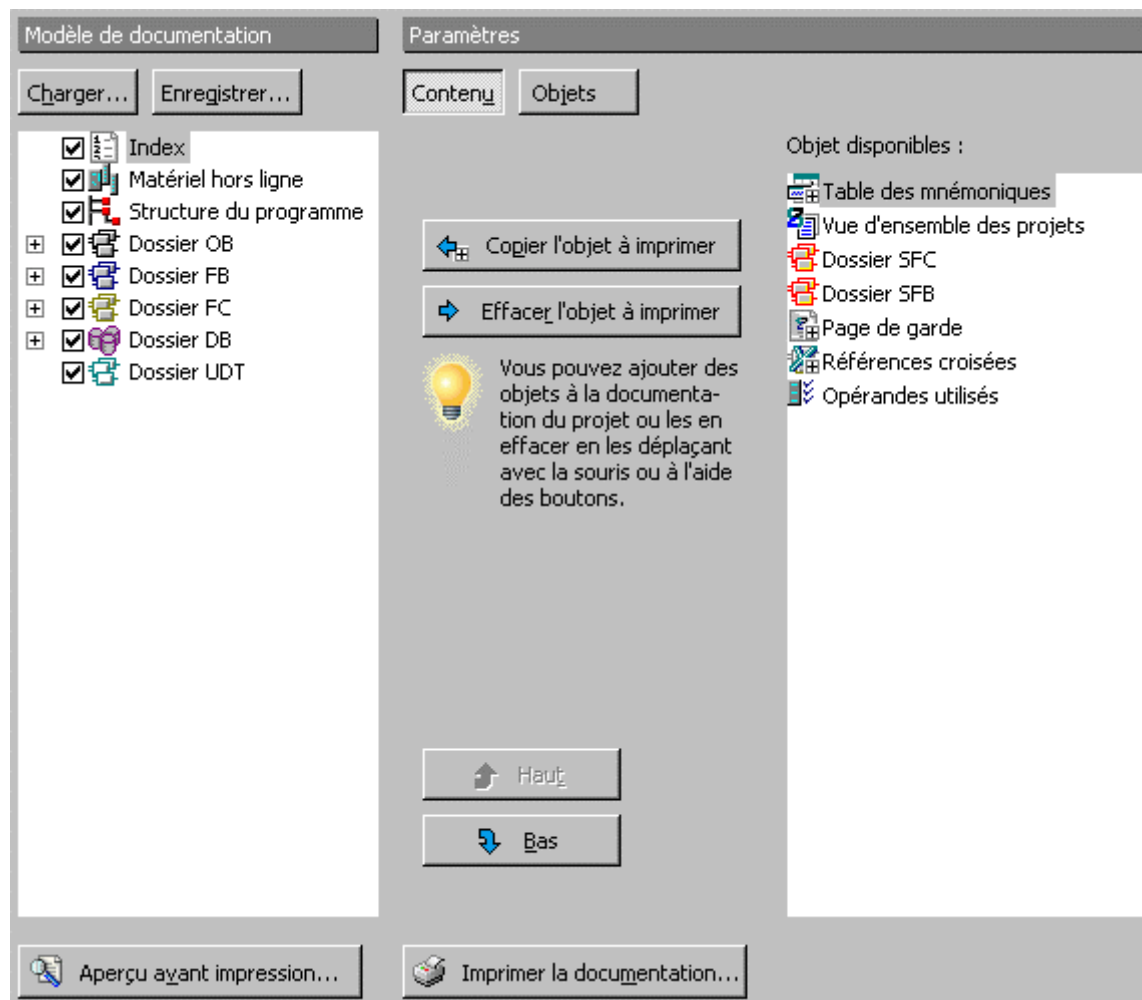
Changer l'ordre des objets à imprimer

Vous déterminez en toute liberté l'ordre des objets dans la liste d'impression (et donc dans la documentation du projet).

1. Sélectionnez l'objet que vous voulez imprimer à une autre position de la documentation.
2. Actionnez le bouton "Haut" ou "Bas".









Le sommaire représente une exception. Il ne peut être placé qu'au début ou à la fin de la documentation du projet. Vous pouvez cependant le faire précéder d'un nombre quelconque de pages de garde comme objets à imprimer.















Quand il y a plusieurs pages de garde, c'est la position du sommaire dans la liste d'impression qui détermine le premier objet mentionné dans le sommaire et numéroté. Par exemple, si le sommaire se trouve entre la 2ème et la 3ème page de garde, la 3ème sera mentionnée dans le sommaire et comprise dans la numérotation, mais les pages de garde 1 et 2 ne figureront pas dans le sommaire.



12.3 Objets à imprimer

Le tableau ci-après énumère les objets imprimables pouvant figurer dans la liste d'impression. Il indique en outre pour quels objets l'aperçu avant impression est disponible et quels objets peuvent servir plusieurs fois.

Objet à imprimer	Icône	Aperçu avant impression ?	Plusieurs emplois ?	Observation
Page de garde		Oui	Oui	Vous pouvez saisir du texte et choisir une police. L'icône avec le signe plus fait partie de la liste des objets à imprimer disponibles. Le signe plus indique que cet objet peut figurer plusieurs fois sur la liste d'impression.
Sommaire		Oui	Non	Le sommaire est imprimé au début ou à la fin de la documentation du projet. Quand il y a plusieurs pages de garde, c'est la position du sommaire dans la liste d'impression qui détermine le premier objet mentionné dans le sommaire et numéroté. Par exemple, si le sommaire se trouve entre la 2ème et la 3ème page de garde, la 3ème sera mentionnée dans le sommaire et comprise dans la numérotation, mais les pages de garde 1 et 2 ne figureront pas dans le sommaire.
Table des mnémoniques		Oui	Oui	Vous pouvez paramétrer le filtre et le tri.
Références croisées		Oui	Oui	Vous pouvez paramétrer le filtre et le tri. L'icône avec le signe plus fait partie de la liste des objets à imprimer disponibles. Le signe plus indique que cet objet peut figurer plusieurs fois sur la liste d'impression.
Opérandes utilisés		Oui	Non	-
Structure du programme		Oui	Non	La structure du programme est imprimée sous forme d'arborescence.
Matériel hors ligne		Oui	Non	Impression de la configuration matérielle
Vue du projet		Oui	Non	Les catégories sont imprimées aussi.

Objet à imprimer	Icône	Aperçu avant impression ?	Plusieurs emplois ?	Observation
Dossier Blocs		Oui	Non	<p>Amenez les objets dossier DB, dossier OB, dossier FB, dossier FC, dossier UDT, dossier SFB ou dossier SFC de la liste des objets à imprimer disponibles dans celle de vos objets à imprimer soit par glisser-lâcher, soit à l'aide du bouton "Déplacer l'objet à imprimer".</p> <p>Un signe plus dans la liste des objets à imprimer signifie que le dossier contient d'autres objets. Cliquez sur le signe plus pour afficher tous les blocs du dossier comme objets à imprimer.</p> <p>Les paramètres définis pour le dossier Blocs s'appliquent par défaut aux blocs subordonnés. Si vous voulez définir des paramètres différents pour certains blocs, activez la case "Paramètres individuels" et précisez les valeurs souhaitées.</p>
Dossier DB		Oui	Non	-
Blocs de données (DB)		Oui	Non	-
Dossier OB		Oui	Non	-
Blocs d'organisation (OB)		Oui	Non	-
Dossier FB		Oui	Non	-
Blocs fonctionnels (FB)		Oui	Non	-
Dossier FC		Oui	Non	-
Fonctions (FC)		Oui	Non	-
Dossier UDT		Oui	Non	-
Types de données (UDT)		Oui	Non	-
Dossier SFB		Oui	Non	-
Blocs fonctionnels système (SFB)		Oui	Non	-
Dossier SFC		Oui	Non	-
Fonctions système (SFC)		Oui	Non	-

12.4 Options, police et mise en page

C'est dans les onglets "Options", "Police" et "Mise en page" que vous déterminez la représentation et l'aspect des différents objets à imprimer. Sélectionnez l'objet dans la liste d'impression placée à gauche et cliquez sur l'onglet de votre choix.

Pour définir les "Options", la "Police" ou la "Mise en page" d'objets individuels, procédez de la manière suivante :

1. Cliquez sur le bouton "Objets" (sous le titre "Paramètres").
2. Sélectionnez l'objet dans la liste à gauche et choisissez l'onglet souhaité.

Options

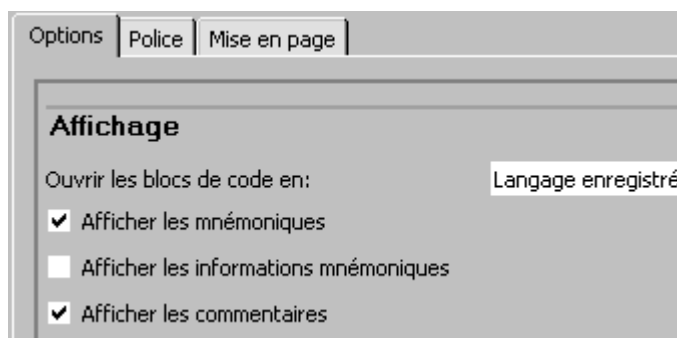
Les options s'appliquent

- au dossier des blocs et aux blocs,
- aux tables des mnémoniques,
- au matériel hors ligne,
- aux références croisées,
- aux pages de garde.

Les options définies pour le dossier des blocs sont également valables pour tous les blocs individuels du dossier, à condition que la case à cocher "Paramètres individuels" soit désactivée dans les options de ces blocs (présélection). Lorsque la case à cocher "Paramètres individuels" est activée, vous pouvez définir d'autres options.

Exemple : options disponibles pour les dossiers de blocs et les blocs

- Ouvrir les blocs de code en
Vous pouvez imprimer le bloc ou les blocs dans le langage de création ou bien choisir LIST, CONT ou LOG comme langage d'impression.
- Afficher les mnémoniques
Activez la case pour que l'objet soit imprimé avec les identificateurs symboliques de ses opérandes.
- Afficher les informations mnémoniques
Activez la case pour que l'objet soit imprimé avec les informations commentant les mnémoniques.
- Afficher les commentaires
Activez la case pour que l'objet soit imprimé avec les commentaires de ses instructions.



Police

Vous pouvez choisir la police, la taille des caractères et le style des caractères pour chaque objet à imprimer.

Ce faisant, vous pouvez distinguer entre polices proportionnelles et non proportionnelles. La police proportionnelle est utilisée dans les textes à représentation justifiée tels que le listage du code.

Un exemple de texte vous montre le résultat à escompter.

Si vous ne souhaitez pas distinguer les objets les uns des autres, utilisez le modèle de format de police "par défaut" ou un modèle enregistré auparavant.

Options Police Mise en page

Modèle de format de police

Standard Modèle de format de police...

Police proportionnelle

Arial 9 **B** *I*

Exemple de texte
Max détourné un taxi puis se rendit à l'opéra pour entendre Verdi.

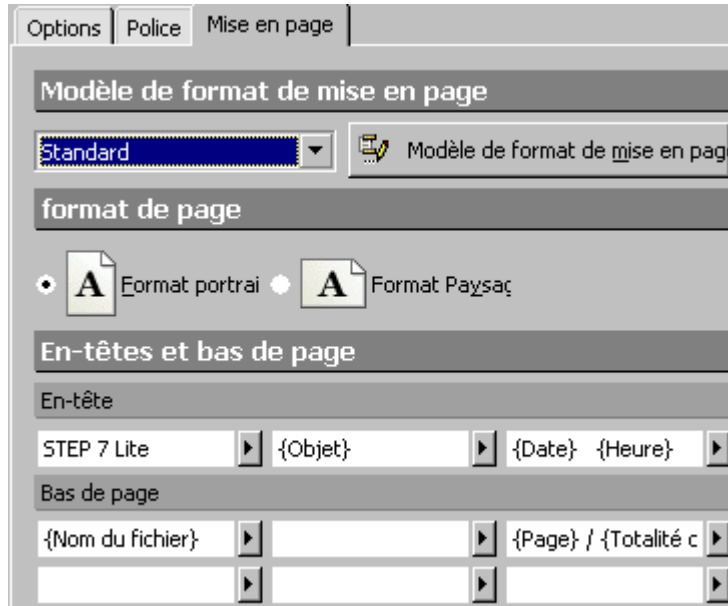
Police non proportionnelle

Courier New 9 **B** *I*

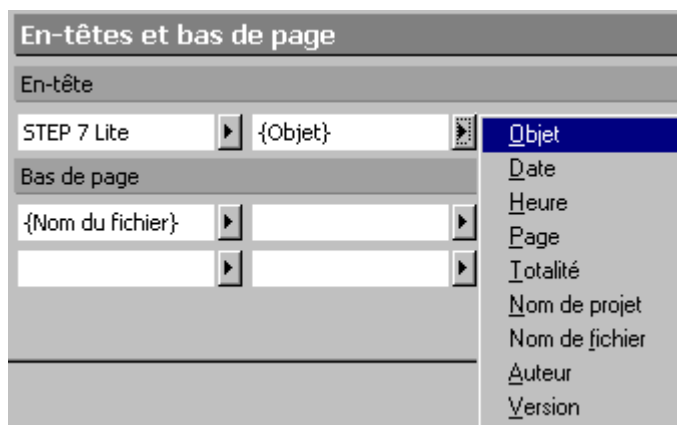
Exemple de texte
Max détourné un taxi puis se rendit à l'opéra pour entendre Verdi.

Mise en page

Dans la mise en page, vous pouvez choisir le format "Portrait" ou "Paysage" et concevoir des en-têtes et des bas de page pour chaque objet à imprimer.



La zone d'en-tête et celle du bas de page sont divisées en trois parties, l'une cadrée à gauche, la seconde centrée et la troisième cadrée à droite. L'en-tête comporte une ligne, le bas de page deux lignes. Vous pouvez en éditer les contenus ou bien les insérer à l'aide d'un menu d'insertion (bouton "Droite").



Quand vous sélectionnez un objet dans le menu d'insertion, il est inséré dans le champ de gauche. Son contenu est mis à jour dynamiquement au cours de l'impression.

Selon l'objet à imprimer, vous disposez des objets dynamiques suivants :

- Objet : nom de l'objet à imprimer
- Date : date en cours au moment de l'impression
- Heure : heure en cours au moment de l'impression
- Page : page en cours (par rapport à l'ensemble de la documentation)
- Totalité : nombre total de pages
- Nom de projet / Nom de fichier : nom du projet STEP 7 Lite
- Auteur : auteur du bloc
- Version : version du bloc

Outre la sélection d'objets du menu d'insertion, vous pouvez aussi insérer un texte statique dans un champ.

Si vous préférez donner le même aspect à l'ensemble de la documentation, affectez à la mise à page le modèle de format de mise en page "par défaut" ou un autre modèle enregistré auparavant.

12.5 Définir des modèles et les utiliser

Une fois que vous avez défini des paramètres pour la documentation du projet, vous pouvez les enregistrer dans un modèle et les réutiliser au besoin.

Modèles de documentation

Un modèle de documentation permet d'enregistrer tous les paramètres définis pour une documentation (exception faite des paramètres individuels définis pour certains blocs des dossiers de blocs).

Parmi les paramètres enregistrés dans un modèle de documentation, il y a les modèles de format de police et de mise en page créés dans le projet.

En cliquant sur le bouton "Enregistrer", vous pouvez enregistrer comme modèle les paramètres individuels que vous avez définis pour une documentation et en faire en plus le modèle par défaut à utiliser pour les nouveaux projets.

Créer un modèle de documentation à partir des paramètres en cours

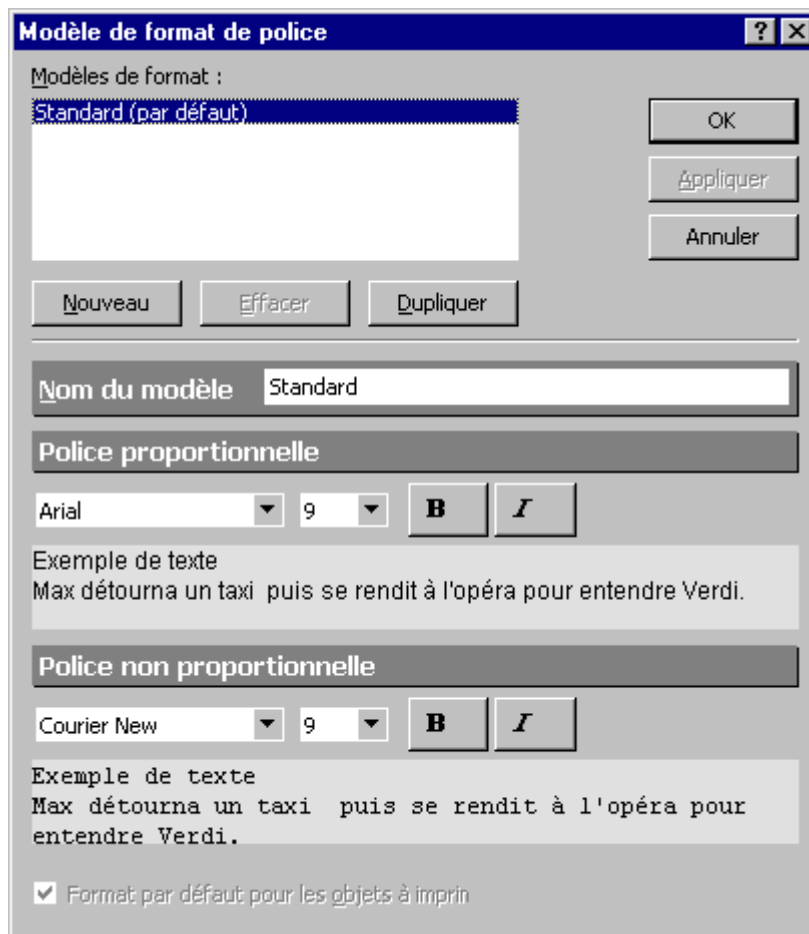
1. Dans la vue "Documentation du projet", cliquez sur le bouton "Enregistrer" (sous le titre "Modèle de documentation").
2. Dans la boîte de dialogue "Enregistre modèle de documentation", entrez un chemin et un nom pour le modèle de documentation.

Utiliser un modèle de documentation

1. Dans la vue "Documentation du projet", cliquez sur le bouton "Charger" (sous le titre "Modèle de documentation").
2. Dans la boîte de dialogue "Charger modèle de documentation", naviguez jusqu'au modèle souhaité et sélectionnez-le.

Modèle de format de police

Les modèles de format de police permettent de définir et d'enregistrer les paramètres utilisés chaque fois pour la police proportionnelle et la police non proportionnelle.



Créer un nouveau modèle de format de police

1. Cliquez sur le bouton "Objets" (sous le titre "Paramètres").
2. Sélectionnez l'onglet "Police".
3. Cliquez sur le bouton "Modèle de format de police".
4. Cliquez sur le bouton "Nouveau" et entrez un nom pour le nouveau modèle de format.
5. Choisissez une police, une taille de caractères et un style de caractères pour police proportionnelle et police non proportionnelle.
6. Si vous voulez que ces polices soient utilisées pour tous les nouveaux objets à imprimer, activez la case "Format par défaut pour les objets à imprimer".
7. Enregistrez le modèle de format de police avec le bouton "OK" ou "Appliquer". (Choisissez "Appliquer", si vous souhaitez que la boîte de dialogue reste ouverte pour pouvoir y effectuer d'autres paramétrages).

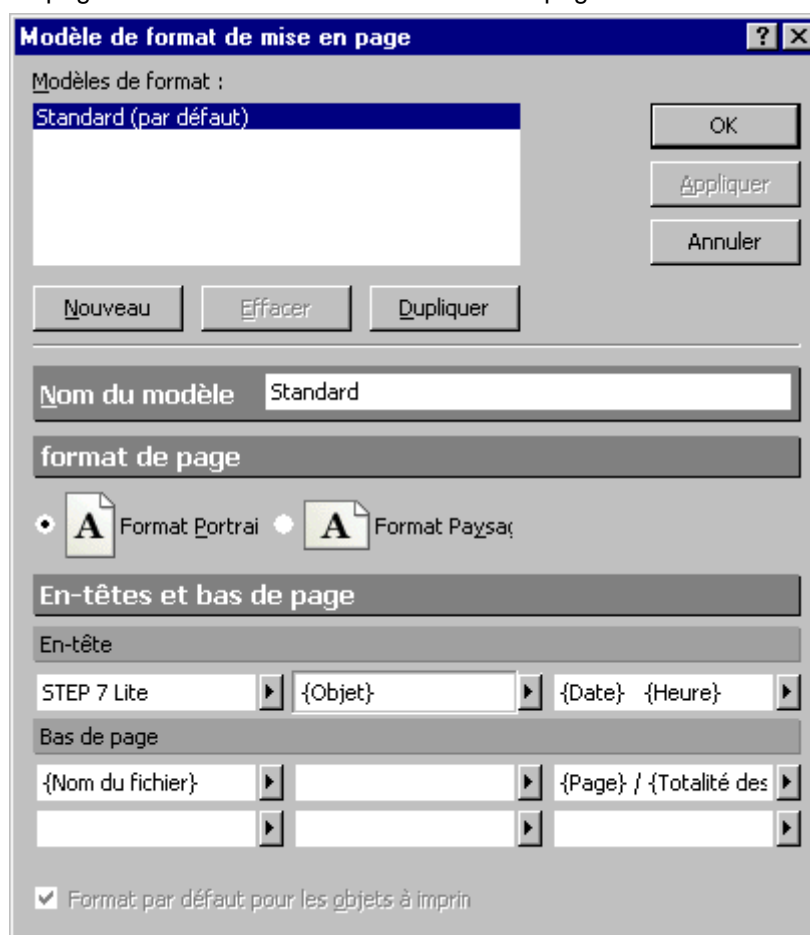
Utiliser un modèle de format de police

Le modèle défini peut être utilisé maintenant dans l'onglet "Police" pour tous les objets à imprimer. Vous pouvez sélectionner les polices dans la liste déroulante "Modèle de format de police".

Aussitôt que vous avez modifié le paramétrage d'un modèle de format de police sélectionné, le contenu de la liste déroulante change sur "Individuel".

Modèle de format de mise en page

Les modèles de format de mise en page permettent d'enregistrer les paramètres de la mise en page et le contenu des en-têtes et bas de page.



Créer un nouveau modèle de format de mise en page

1. Cliquez sur le bouton "Objets" (sous le titre "Paramètres").
2. Sélectionnez l'onglet "Mise en page".
3. Cliquez sur le bouton "Modèle de format de mise en page".
4. Cliquez sur le bouton "Nouveau" et entrez un nom pour le nouveau modèle de format.
5. Choisissez le format de page et définissez en-tête et bas de page.
6. Si vous voulez que cette mise en page soit utilisée pour tous les nouveaux objets à imprimer, activez la case "Format par défaut pour les objets à imprimer".
7. Enregistrez le modèle de format de mise en page avec le bouton "OK" ou "Appliquer". (Choisissez "Appliquer", si vous souhaitez que la boîte de dialogue reste ouverte pour pouvoir y effectuer d'autres paramétrages).

Utiliser un modèle de format de mise en page

Le modèle défini peut être utilisé maintenant dans l'onglet "Mise en page" pour tous les objets à imprimer. Vous pouvez sélectionner la mise en page dans la liste déroulante "Modèle de format de mise en page".

Aussitôt que vous avez modifié le paramétrage d'un modèle de format de mise en page sélectionné, le contenu de la liste déroulante change sur "Individuel".

12.6 Imprimer une documentation du projet

Une fois la documentation du projet définie, vous en lancez l'impression avec le bouton "Imprimer la documentation". La boîte de dialogue qui s'ouvre vous permet de choisir encore l'imprimante et de la paramétrer. Les options disponibles ici dépendent du pilote d'imprimante utilisé.

Servez-vous du bouton "Aperçu avant impression" pour vérifier à l'avance que le résultat correspond bien à ce que vous souhaitez.

Imprimer des objets individuels

Procédez de la manière suivante :

1. Ouvrez l'objet qui convient dans la fenêtre du projet pour que les informations à imprimer s'affichent à l'écran.
2. La commande **Fichier > Imprimer** permet d'imprimer l'objet individuel. La boîte de dialogue qui s'ouvre vous permet de choisir encore l'imprimante et de la paramétrer.
3. Servez-vous de la commande **Fichier > Aperçu avant impression** pour vérifier à l'avance que le résultat correspond bien à ce que vous souhaitez.

Les objets individuels sont imprimés conformément aux paramètres les concernant dans la documentation du projet (Options, Police et Mise en page). Si ces paramètres n'ont pas été définis, ils seront imprimés conformément aux paramètres par défaut.

13 Astuces et conseils

13.1 Remplacement de modules dans la table de configuration

Lorsque vous retouchez la configuration d'une station et que vous souhaitez remplacer un module, par exemple par un autre qui a un autre numéro de référence, procédez de la manière suivante :

1. Amenez le module par glisser-lâcher du catalogue sur le module ("ancien") déjà placé dans la vue graphique ou la table de la configuration matérielle.
2. Lâchez le nouveau module ; dans la mesure du possible, il adopte alors les paramètres du module déjà enfiché.

Cette façon de faire est plus rapide que celle qui consiste à effacer d'abord l'ancien module avant d'insérer le nouveau et de le paramétrer.

13.2 Test à l'aide de la table des variables

Voici une série de conseils facilitant la visualisation, le forçage et le forçage permanent de variables.

- Vous pouvez taper les mnémoniques et les opérandes dans la colonne "Mnémonique" comme dans la colonne "Opérande". L'entrée est automatiquement reportée dans la colonne appropriée.
- Pour obtenir l'affichage de la valeur forcée dans la colonne "Valeurs d'état", choisissez "Permanent" comme point de déclenchement la visualisation.
- Lorsque le bouton "Paramètres étendus" est activé :
- Pour forcer des sorties, il est judicieux de choisir le mode "Fin de cycle Cyclique" ou "Fin de cycle Unique".
- Pour forcer des entrées, il est judicieux de choisir le mode "Début de cycle Cyclique" ou "Début de cycle Unique".
- Le mode de forçage "Permanent" combine les propriétés précitées.
- Vous ne pouvez inscrire que des mnémoniques déjà définis dans la table des mnémoniques. Il faut saisir un mnémonique exactement comme il est défini dans la table. Ceux qui comportent des caractères spéciaux sont à écrire entre guillemets (ex. : "Moteur.stop", "Moteur+stop", "Moteur-stop").
- Il est possible de choisir le mode de visualisation durant la visualisation de variables.
- Pour saisir une plage d'opérandes consécutifs :
utilisez la commande **Insertion > Plage d'opérandes**.

- Pour changer de format d'affichage dans plusieurs lignes à la fois :
 - sélectionnez la partie de table dans laquelle vous voulez changer de format en faisant glisser le pointeur dessus, le bouton gauche de la souris étant enfoncé ;
 - choisissez la nouvelle représentation avec la commande **Affichage > Choisir format d'affichage** ; le format changera seulement pour celles des lignes sélectionnées qui autorisent ce changement.

13.3 Travailler sur la PG/le PC sans projet original

Si vous voulez régler des paramètres dans la CPU ou dans des modules ou bien adapter des programmes et que vous ne disposez pas d'un projet original dans la PG ou le PC (cas de dépannage), vous pouvez procéder comme suit :

1. Faites le nécessaire pour qu'il y ait une liaison en ligne entre PG/PC et CPU.
2. Dans la fenêtre en ligne du projet (onglet "CPU en ligne"), sélectionnez l'objet que vous souhaitez éditer, par exemple l'icône "CPU en ligne" si vous voulez effectuer des modifications importantes.
3. Choisissez la commande **Fichier > Charger dans PG**.
Si aucun projet n'est encore ouvert, un nouveau projet est automatiquement créé.
Tous les objets sélectionnés sont chargés dans la PG ; vous pourrez ensuite les ouvrir dans la fenêtre du projet (onglet "Projet") et les éditer.
4. Enregistrez les modifications apportées au projet (commande **Fichier > Enregistrer**).
5. Chargez les modifications dans la CPU (commande **Fichier > Charger dans CPU**).

Si vous ouvrez les blocs et la configuration en ligne (dans la fenêtre du projet, onglet "CPU en ligne"), vous ne pourrez pas les éditer.

A Annexe

A.1 Etats de fonctionnement

A.1.1 Etats de fonctionnement et changements d'état de fonctionnement

Etats de fonctionnement

Les états de fonctionnement décrivent le comportement de la CPU à n'importe quel instant. La connaissance de ces états est utile pour la programmation de la mise en route, le test de l'automate ainsi que pour le diagnostic d'erreur.

Les CPU peuvent se trouver dans les états de fonctionnement suivants :

- Arrêt
- Mise en route
- Marche
- Attente

A l'état de fonctionnement "Arrêt" (STOP), la CPU vérifie la présence de tous les modules configurés ou utilisés selon l'adressage par défaut et elle met la périphérie dans un état de base prédéfini. Le programme utilisateur n'est pas exécuté dans cet état.

A l'état de fonctionnement "Mise en route", on distingue entre les modes de mise en route "Démarrage à chaud" et "Démarrage à froid" :

- En cas de démarrage à chaud, l'exécution du programme commence au début avec une "position de base" des données système et des zones d'opérandes utilisateur (les temporisations, compteurs et mémentos non rémanents sont remis à zéro).
- En cas de démarrage à froid, la mémoire image des entrées est lue et le programme utilisateur STEP 7 Lite est exécuté en commençant par la première opération dans l'OB1 (ceci est également le cas pour le démarrage à chaud).
 - Les blocs de données créés par SFC dans la mémoire de travail sont effacés, les autres blocs de données prennent la valeur par défaut de la mémoire de chargement.
 - La mémoire image ainsi que tous les compteurs, temporisations et mémentos sont remis à zéro, qu'ils aient été paramétrés comme rémanents ou pas.

A l'état de fonctionnement "Marche" (RUN), la CPU traite le programme utilisateur, met à jour les entrées et les sorties, traite les alarmes et les messages d'erreur.

Le traitement du programme utilisateur est suspendu à l'état de fonctionnement "Attente" et vous pouvez tester le programme pas à pas. Il n'est pas possible de passer dans cet état au moyen de STEP 7 Lite.

Dans tous ces états de fonctionnement, la CPU peut communiquer via l'interface MPI.

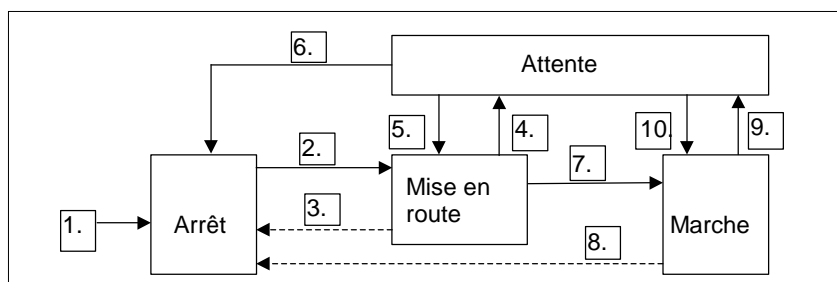
Autres états de fonctionnement

Quand la CPU n'est pas prête à fonctionner, elle se trouve dans l'un des deux états suivants :

- Sans tension : il n'y a pas de tension secteur.
- Défectueuse : une erreur irrémédiable s'est produite.
Vérifiez si la CPU est vraiment défectueuse : mettez-la en STOP et éteignez puis allumez le commutateur secteur. Si la CPU se met en route, lisez la mémoire tampon de diagnostic afin d'analyser l'erreur. Si la CPU ne se met pas en route, il faut la remplacer.

Changements d'état de fonctionnement

La figure suivante représente les états de fonctionnement et les changements d'état de fonctionnement des CPU S7-300.



Le tableau suivant donne les conditions des changements d'état de fonctionnement.

Changement	Description
1	La CPU est à l'état "Arrêt" (STOP) lors de la mise sous tension.
2	La CPU passe à l'état "Mise en route" : <ul style="list-style-type: none"> • lorsque la position RUN ou RUN-P est sélectionnée via le commutateur à clé ou la console de programmation • ou après déclenchement automatique d'un mode de mise en route par mise sous tension, • lorsque la fonction de communication "RESUME" ou "START" est exécutée. Le commutateur à clé doit se trouver dans les deux cas sur RUN ou RUN-P.
3	La CPU passe de nouveau à l'état "Arrêt" (STOP) lorsque : <ul style="list-style-type: none"> • une erreur est détectée pendant la mise en route, • elle est mise sur STOP via le commutateur à clé ou depuis la PG, • une commande d'arrêt est traitée dans l'OB de mise en route, • la fonction de communication "STOP" est exécutée.
4	La CPU passe à l'état de fonctionnement "Attente" lorsqu'un point d'arrêt est atteint dans le programme de mise en route (STEP 7 Lite ne permet pas de définir de points d'arrêt).
5	La CPU passe à l'état de fonctionnement "Mise en route" lorsque le point d'arrêt était défini dans un programme de mise en route et que la commande QUITTER ATTENTE est exécutée (fonction de test).
6	La CPU passe de nouveau à l'état "Arrêt" (STOP) lorsque : <ul style="list-style-type: none"> • elle est mise sur STOP via le commutateur à clé ou depuis la PG, • la fonction de communication "STOP" est exécutée.
7	La CPU passe à l'état "Marche" (RUN) quand la mise en route s'achève sans erreur.
8	La CPU passe de nouveau à l'état "Arrêt" (STOP) lorsque : <ul style="list-style-type: none"> • une erreur est détectée à l'état "Marche" et que l'OB correspondant n'est pas chargé, • elle est mise sur STOP via le commutateur à clé ou depuis la PG, • une commande d'arrêt est traitée dans le programme utilisateur, • la fonction de communication "STOP" est exécutée.
9	La CPU passe à l'état de fonctionnement "Attente" lorsqu'un point d'arrêt est atteint dans le programme utilisateur (STEP 7 Lite ne permet pas de définir de points d'arrêt).
10	La CPU passe à l'état de fonctionnement "Marche" lorsqu'un point d'arrêt était défini et que la commande QUITTER ATTENTE est exécutée.

Priorité des états de fonctionnement

Lorsque plusieurs changements d'état de fonctionnement sont demandés en même temps, la CPU passe à celui qui a la priorité la plus élevée. Par exemple, si le commutateur de mode se trouve sur RUN et que vous tentez de commuter la CPU à l'état "Arrêt" (STOP) depuis la PG, la CPU passera bien à l'état "Arrêt", car il a la priorité la plus élevée.

Priorité	Etat de fonctionnement
La plus élevée	Arrêt
	Attente
	Mise en route
La plus faible	Marche

A.1.2 Etat de fonctionnement "Arrêt" (STOP)

Le programme utilisateur n'est pas traité dans cet état. Toutes les sorties sont mises à des valeurs de substitution afin d'amener le processus commandé à un état sûr. La CPU vérifie

- s'il y a des problèmes de matériel (par exemple, modules non disponibles) ;
- si la CPU doit prendre les valeurs par défaut ou s'il existe des jeux de paramètres ;
- si les conditions annexes pour le comportement de mise en route programmé sont correctes ;
- s'il y a des problèmes de logiciel système.

Il est possible, à l'état "Arrêt", de recevoir des données globales et d'exécuter une communication à sens unique passive via des SFB de communication pour liaisons configurées et via des SFC de communication pour liaisons non configurées.

Effacement général

Vous pouvez effectuer un effacement général de la CPU à l'état "Arrêt", soit manuellement en positionnant le commutateur à clé sur MRES, soit à partir de la console de programmation (par exemple, avant le chargement d'un programme utilisateur).

L'effacement général remet la CPU dans son "état fondamental", ce qui signifie que :_

- Le programme utilisateur complet dans la mémoire de travail et dans la mémoire de chargement RAM ainsi que toutes les zones d'opérandes sont effacés.
- Les paramètres système ainsi que les paramètres des modules et de la CPU reprennent leur valeur par défaut. Seuls les paramètres MPI définis avant l'effacement général sont conservés.
- Lorsqu'une carte mémoire est enfichée (EPROM flash), la CPU copie le programme utilisateur de la carte mémoire dans la mémoire de travail (y compris les paramètres de CPU et de modules si les données de configuration correspondantes se trouvent également sur la carte mémoire).

La mémoire tampon de diagnostic, les paramètres MPI, l'heure et le compteur d'heures de fonctionnement ne sont pas remis à zéro.

A.1.3 Etat de fonctionnement "Mise en route"

Après la mise sous tension, un programme de mise en route est exécuté avant que la CPU commence à exécuter le programme utilisateur. Dans ce programme de mise en route, vous pouvez prédéfinir certaines valeurs pour votre programme cyclique en programmant les OB de mise en route de manière appropriée.

Il y a deux modes de mise en route : démarrage à chaud et démarrage à froid.

A l'état de fonctionnement "Mise en route" :

- le programme contenu dans l'OB de mise en route est exécuté (OB100 pour démarrage à chaud, OB102 pour démarrage à froid),
- l'exécution d'un programme déclenché par horloge et par alarme n'est pas possible,
- les temporisations sont mises à jour,
- le compteur d'heures de fonctionnement tourne,
- les sorties TOR des modules de signaux sont bloquées, mais peuvent être mises à 1 par accès direct.

Démarrage à chaud

Un démarrage à chaud est toujours autorisé à moins qu'un effacement général n'ait été demandé par le système. Seul le démarrage à chaud est possible après :

- effacement général,
- chargement du programme utilisateur à l'état de fonctionnement "Arrêt" de la CPU,
- débordement de la pile des interruptions ou de la pile des blocs,
- interruption d'un démarrage à chaud (par mise hors tension ou via le commutateur de mode),
- dépassement de la limite de temps d'interruption paramétrée pour le redémarrage.

Démarrage à chaud manuel

Un démarrage à chaud manuel peut être déclenché :

- via le commutateur de mode de fonctionnement,
- par une commande de menu provenant de la PG ou par des fonctions de communication (quand le commutateur de mode est en position RUN ou RUN-P).

Démarrage à chaud automatique

Un démarrage à chaud automatique peut être déclenché à la mise sous tension lorsque :

- la CPU n'était pas à l'arrêt lors de la mise hors tension,
- le commutateur de mode est en position RUN ou RUN-P,
- la CPU a été interrompue par coupure secteur au démarrage à chaud (indépendamment du mode de mise en route paramétré).

Démarrage à chaud automatique sans sauvegarde

Si votre CPU fonctionne sans pile de sauvegarde (car fonctionnement sans maintenance requis), elle est automatiquement soumise à un effacement général à la mise sous tension ou au retour de la tension après mise hors tension, puis un démarrage à chaud est exécuté. Le programme utilisateur doit être disponible sur carte mémoire (EPROM flash).

Zones de données rémanentes après coupure secteur

Quand vous paramétrez la CPU avec STEP 7 Lite, vous pouvez définir comme rémanents des mémentos, temporisations, compteurs et zones de blocs de données afin d'éviter de perdre des données en cas de coupure de courant. Dans ce cas, les zones paramétrées comme rémanentes sont conservées dans la RAM non volatile interne à la CPU.

Le tableau ci-après montre le comportement de rémanence des CPU S7-300 en cas de démarrage à chaud et de démarrage à froid.

Comportement de rémanence dans la mémoire de travail (avec mémoire de chargement EPROM et RAM)

Mode de mise en route (S7-300)	Opérandes	Opérandes après mise en route
Démarrage à chaud (CPU sauvegardée par pile)	Blocs de code et de données sont conservés
Démarrage à chaud (CPU sauvegardée par pile)	Mémentos, temporisations, compteurs ne sont conservés que s'ils ont été paramétrés comme rémanents, sinon remis à zéro.
Démarrage à chaud (CPU pas sauvegardée par pile)	Blocs de code et de données sont copiés de la mémoire de chargement EPROM dans la mémoire de travail. Les contenus de blocs de données paramétrés comme rémanents sont conservés ; les blocs chargés a posteriori ou générés au moyen du programme sont perdus.
Démarrage à chaud (CPU pas sauvegardée par pile)	Mémentos, temporisations, compteurs ne sont conservés que s'ils ont été paramétrés comme rémanents, sinon remis à zéro.
Démarrage à froid	Blocs de code et de données sont copiés de la mémoire de chargement EPROM dans la mémoire de travail.
Démarrage à froid	Mémentos, temporisations, compteurs sont remis à zéro (même s'ils ont été paramétrés comme rémanents).

Activités à la mise en route

Le tableau ci-après montre les activités exécutées par la CPU lors de la mise en route.

Activités dans l'ordre de leur exécution	Démarrage à chaud	Démarrage à froid
Effacer la pile des interruptions et la pile des blocs	X	X
Effacer les mémentos, temporisations et compteurs non rémanents	X	0
Effacer tous les mémentos, temporisations et compteurs	0	X
Effacer la mémoire image des sorties	X	X
Remettre à zéro les sorties des modules de signaux	X	X
Rejeter les alarmes de processus	X	X
Rejeter les alarmes temporisées	X	X
Rejeter les alarmes de diagnostic	X	X
Actualiser la liste d'état système (SZL)	X	X
Evaluer les paramètres de modules et les transmettre aux modules ou bien leur transmettre les valeurs par défaut	X	X
Exécuter l'OB de mise en route respectif	X	X
Exécuter le cycle restant (partie du programme utilisateur n'ayant pu être exécutée par suite d'une mise hors tension)	0	0
Actualiser la mémoire image des entrées	X	X
Débloquer les sorties TOR (supprimer le signal OD) après passage à l'état de fonctionnement "Marche"	X	X
X signifie est exécuté		
0 signifie n'est pas exécuté		

Abandon de la mise en route

Si des erreurs apparaissent au cours de la mise en route, elle est abandonnée et la CPU passe ou reste à l'état "Arrêt".

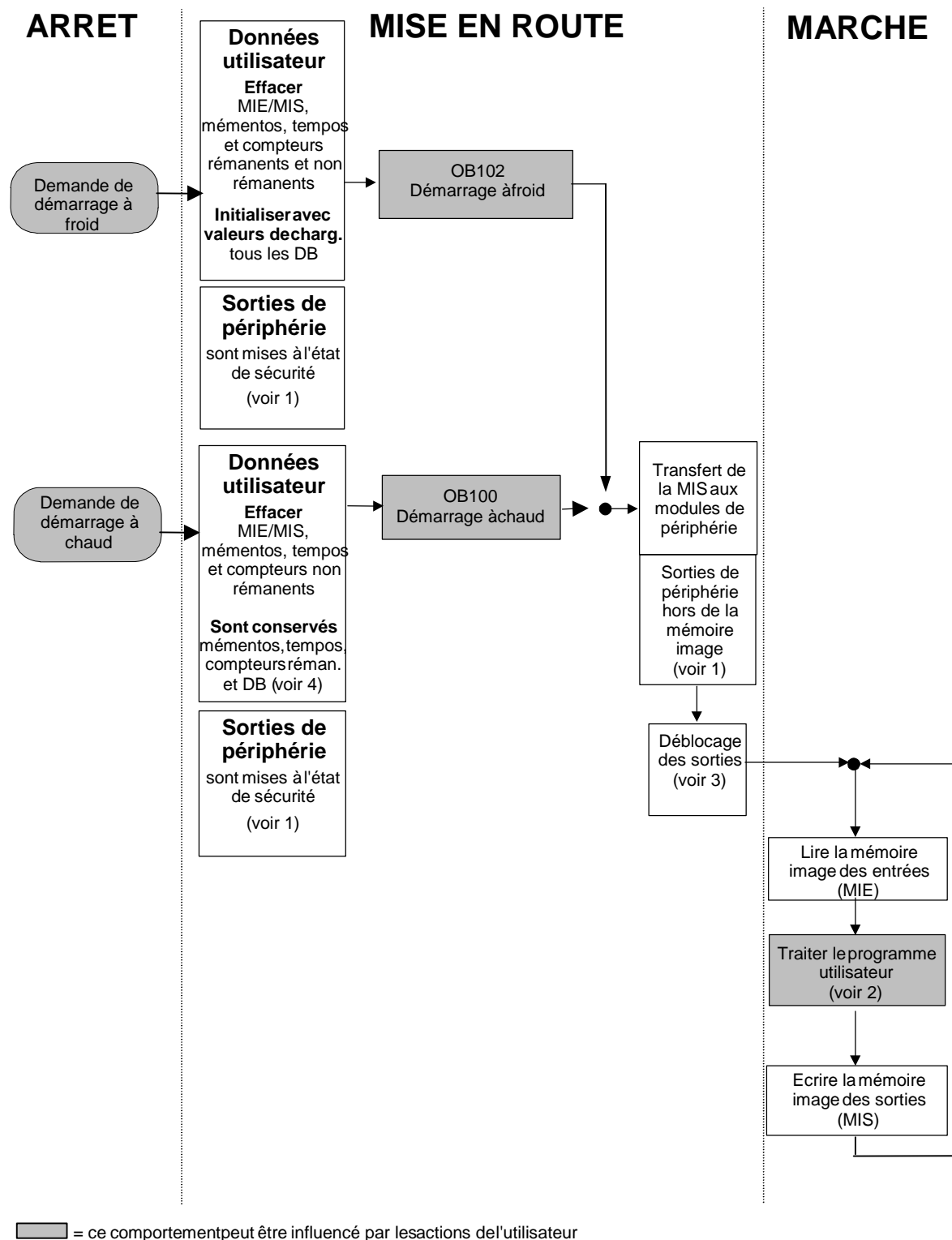
Un démarrage à chaud abandonné doit être répété.

Une mise en route n'est pas exécutée ou elle est abandonnée :

- quand le commutateur à clé de la CPU est en position STOP,
- quand un effacement général est demandé,
- quand une carte mémoire est enfichée dont l'identificateur d'application n'est pas admissible pour STEP 7 Lite (par ex. STEP 5),
- quand le programme utilisateur contient un OB que la CPU ne connaît pas ou qui a été verrouillé,
- si la CPU constate, après la mise sous tension, que tous les modules mentionnés par la table de configuration créée avec STEP 7 Lite ne sont pas effectivement enfichés (différence entre paramétrage nominal et effectif non autorisée),
- si des erreurs apparaissent lors de l'évaluation des paramètres de module.

Déroulement

La figure ci-après montre les activités de la CPU dans les états de fonctionnement "Mise en route" et "Marche" (RUN).



Légende de la figure

"Activités de la CPU dans les états de fonctionnement Mise en route et Marche"

1. Toutes les sorties de périphérie sont mises à l'état de sécurité (valeur par défaut =0) par le matériel des modules de périphérie, qu'elles soient utilisées dans le programme utilisateur au sein de la zone de la mémoire image du processus ou en dehors.

Si vous employez des modules de signaux pouvant traiter une valeur de remplacement, il est possible de paramétrer le comportement des sorties, par exemple Conserver dernière valeur.

2. Les OB d'alarme disposent également d'une mémoire image des entrées actuelle lors de leur premier appel.
3. Vous pouvez recourir aux mesures suivantes pour déterminer l'état des sorties de périphérie dans le premier cycle du programme utilisateur :
 - utiliser des modules de sorties paramétrables pour pouvoir écrire des valeurs de remplacement ou conserver la dernière valeur ;
 - donner des valeurs par défaut aux sorties dans l'OB de mise en route (OB100, OB102).
4. Dans les systèmes S7-300 sans sauvegarde, seules les zones DB configurées comme rémanentes sont conservées.

A.1.4 Etat de fonctionnement "Marche" (RUN)

A l'état "Marche" s'effectue le traitement de programme cyclique et commandé par horloge et par alarme :

- La mémoire image des entrées est lue.
- Le programme utilisateur est traité.
- La mémoire image des sorties est émise.

L'échange actif de données entre les CPU par communication par données globales (table des données globales), par SFB de communication pour les liaisons configurées et par SFC pour les liaisons non configurées n'est possible qu'à l'état de "Marche".

Le tableau ci-après indique quand l'échange de données est possible dans les différents états de fonctionnement :

Type de communication	Etat de fonctionnement de la CPU 1	Sens de l'échange de données	Etat de fonctionnement de la CPU 2
Communication par données globales	Marche	↔	Marche
	Marche	→	Arrêt/Attente
	Arrêt	←	Marche
	Arrêt	X	Arrêt
	Attente	X	Arrêt/Attente
Communication à sens unique	Marche	→	Marche
Par SFB de communication	Marche	→	Arrêt/Attente
Communication à double sens par SFB de communication	Marche	↔	Marche
Communication à sens unique	Marche	→	Marche
Par SFC de communication	Marche	→	Arrêt/Attente
Communication à double sens par SFC de communication	Marche	↔	Marche
↔ signifie échange de données possible dans les deux sens → signifie échange de données possible dans un sens seulement X signifie échange de données impossible			

Commutateur de mode de fonctionnement et état de fonctionnement

- Commutateur de mode de fonctionnement sous forme de **commutateur à clé** :
Pour les CPU possédant un commutateur à clé, vous pouvez sélectionner les positions RUN ou RUN-P.
En position RUN, aucun accès à la mémoire de chargement de la CPU n'est possible depuis votre PG/PC, à moins que lors du paramétrage de la CPU vous ayez paramétré un mot de passe pour le niveau 1 et que vous connaissiez ce mot de passe.
En position RUN-P, un accès sans restrictions est possible, dans la mesure où vous n'avez défini aucun mot de passe lors du paramétrage de la CPU.
- Commutateur de mode de fonctionnement sous forme de **commutateur à bascule** (CPU 31xC) :
Pour les CPU possédant un commutateur à bascule, vous pouvez uniquement sélectionner la position RUN et pas la position RUN-P. En position RUN un accès sans restrictions est possible, dans la mesure où vous n'avez défini aucun mot de passe lors du paramétrage de la CPU (correspond à la position RUN-P des CPU avec commutateur à clé).

A.1.5 Etat de fonctionnement "Attente"

L'état de fonctionnement "Attente" tient une place à part. Il ne sert qu'à des fins de test à la mise en route ou en marche. Dans l'état de fonctionnement "Attente" :

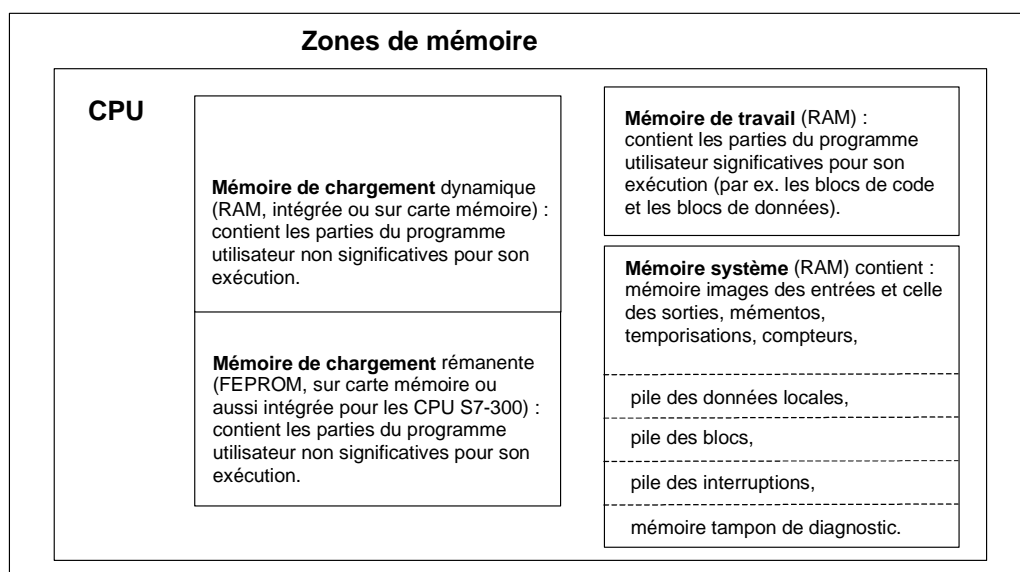
- Tous les temps sont suspendus : les temporisations et les compteurs d'heures de fonctionnement ne sont pas traités. Les temps de surveillance ainsi que les périodes de base des niveaux commandés par horloge sont arrêtés.
- L'horloge temps réel fonctionne.
- Les sorties ne sont pas libérées, mais peuvent être validées à des fins de test.
- Il est possible de commander les entrées et les sorties.
- En cas de coupure secteur et de retour de la tension, les CPU avec sauvegarde en "Attente" passent à l'état "Arrêt" et n'exécutent pas de redémarrage ni de démarrage à chaud automatique. Au retour de la tension, les CPU sans sauvegarde effectuent un démarrage à chaud automatique sans sauvegarde.
- Il est possible de recevoir des données globales et d'exécuter une communication à sens unique passive par SFB de communication pour liaisons configurées et par SFC de communication pour liaisons non configurées (voir aussi tableau à l'état de fonctionnement "Marche").

A.2 Zones de mémoire des CPU S7

A.2.1 Organisation des zones de mémoire

La mémoire des CPU S7 comporte trois zones (voir la figure ci-après) :

- La mémoire de chargement sert à enregistrer le programme utilisateur sans affectation de mnémoniques ni commentaires (ces derniers restent dans la mémoire de la console de programmation). La mémoire de chargement peut être de la mémoire vive (RAM) ou de la mémoire EPROM.
- Les blocs identifiés comme non significatifs pour l'exécution sont exclusivement chargés dans la mémoire de chargement.
- La mémoire de travail (mémoire vive intégrée) sert à enregistrer les parties du programme significatives pour l'exécution du programme. Le traitement du programme a lieu exclusivement dans la mémoire de travail et dans la mémoire système.
- La mémoire système (mémoire vive) contient les éléments de mémoire que chaque CPU met à la disposition du programme utilisateur, comme par exemple mémoire image des entrées, mémoire image des sorties, mementos, temporisations et compteurs. La mémoire système contient, en outre, la pile des blocs et la pile des interruptions.
- C'est également la mémoire système de la CPU qui fournit la mémoire temporaire (pile des données locales) allouée au programme pour les données temporaires à l'appel d'un bloc. Ces données sont valables seulement tant que le bloc est actif.



A.2.2 Mémoire de chargement et mémoire de travail

Lorsque vous chargez le programme utilisateur de la console de programmation dans la CPU, seuls les blocs de code et les blocs de données sont chargés dans la mémoire de chargement et dans la mémoire de travail de la CPU.

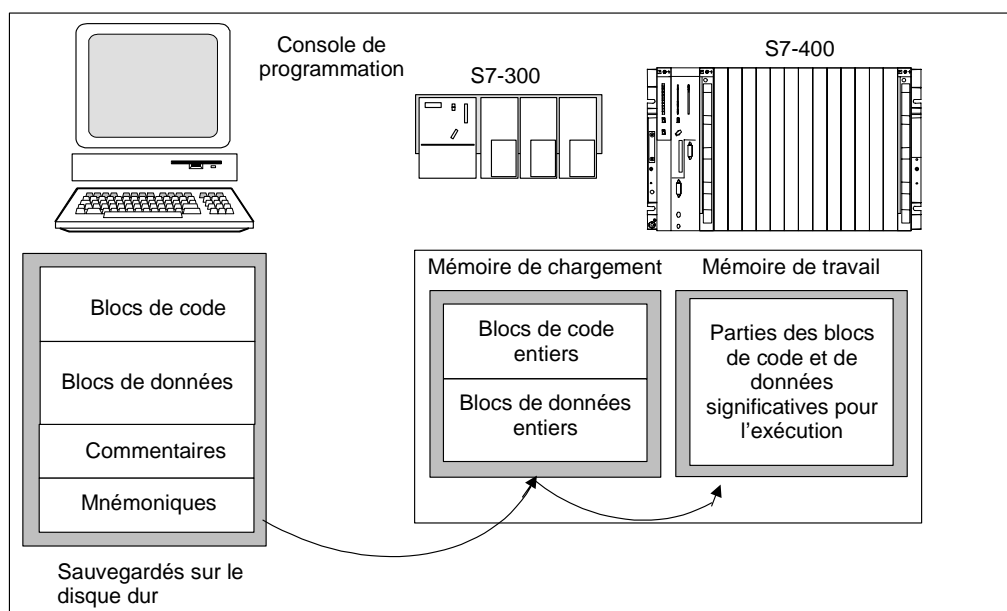
La table des mnémoniques et les commentaires de blocs restent dans la zone de mémoire de la PG.

Répartition du programme utilisateur

Afin de garantir un traitement rapide du programme utilisateur et de ne pas surcharger inutilement la mémoire de travail non extensible, seules les parties des blocs significatives pour le traitement du programme sont chargées dans la mémoire de travail.

Les parties de blocs non requises pour l'exécution du programme (par exemple, les en-têtes de blocs) restent dans la mémoire de chargement.

La figure suivante représente le chargement du programme dans la mémoire de la CPU.



Nota

Les blocs de données créés dans le programme utilisateur à l'aide de fonctions système (par ex. SFC22 CREAT_DB) sont enregistrés complètement dans la mémoire de travail par la CPU.

Certaines CPU disposent de zones gérées séparément pour le code et les données dans la mémoire de travail. Pour ces CPU, la taille et l'occupation de ces zones sont affichées sur la page d'onglet "Mémoire" de l'état du module.

Qualification de blocs de données comme "non significatifs pour l'exécution"

On peut qualifier de "non significatifs pour l'exécution" - mot-clé UNLINKED - les blocs de données programmés comme partie d'un programme LIST dans un fichier source. Lors du chargement dans la CPU, ces DB ne sont donc rangés que dans la mémoire de chargement. Si besoin est, il est possible de copier leur contenu dans la mémoire de travail à l'aide de la SFC20 BLKMOV.

Cela permet donc de gagner de la place dans la mémoire de travail, la mémoire de chargement extensible servant de mémoire intermédiaire (par exemple pour les formules : seule la prochaine formule à traiter est chargée dans la mémoire de travail).

Structure de la mémoire de chargement

Il est possible d'étendre la mémoire de chargement à l'aide de cartes mémoire. La taille maximale de la mémoire de chargement est donnée dans le manuel *Système d'automatisation S7-300, Installation et configuration – Caractéristiques de la CPU*.

Pour les CPU S7-300, la mémoire de chargement peut comporter une partie EPROM intégrée en plus de la partie RAM. Dans les blocs de données, certaines zones peuvent être déclarées comme rémanentes par paramétrage avec STEP 7 Lite (voir Zones de mémoire rémanentes des CPU S7-300).

Comportement de la mémoire de chargement pour les zones RAM et EPROM

Selon que vous choisissez une carte mémoire RAM ou EPROM pour étendre la mémoire de chargement, il peut s'ensuivre un comportement différent de cette mémoire lors du chargement, du rechargement et de l'effacement général.

Le tableau suivant représente les possibilités de chargement :

Type de mémoire	Possibilités de chargement	Type de chargement
Mémoire vive (RAM)	Chargement et effacement de blocs individuels	Liaison PG-CPU
	Chargement et effacement d'un programme entier	Liaison PG-CPU
	Rechargement de blocs individuels	Liaison PG-CPU
EPROM intégrée ou enfichable	Chargement de programmes entiers	Liaison PG-CPU
EPROM enfichable	Chargement de programmes entiers	Chargement de l'EPROM sur la PG et enfichage de la carte mémoire dans la CPU. Chargement de l'EPROM sur la CPU.

Les programmes enregistrés en mémoire vive sont perdus lorsque vous exécutez un effacement général de la CPU (MRES) ou lorsque vous retirez la CPU ou la carte mémoire RAM.

Les programmes enregistrés sur cartes mémoire EPROM ne sont pas perdus en cas d'effacement général et restent conservés même sans sauvegarde par pile (transport, copies de sûreté).

A.2.3 Mémoire système

A.2.3.1 Utilisation des zones de mémoire système

La mémoire système des CPU S7 est subdivisée en zones d'opérands (voir le tableau ci-après). En utilisant les opérations correspondantes, vous accédez dans votre programme aux données directement dans la plage d'opérands en question.

Plage d'opérands	Accès par des unités de taille suivante	Notation S7	Description
Mémoire image des entrées	Entrée (bit)	E	Au début de chaque cycle, la CPU lit les entrées provenant des modules d'entrées et enregistre ces valeurs dans la mémoire image des entrées.
	Octet d'entrée	EB	
	Mot d'entrée	EW	
	Double mot d'entrée	ED	
Mémoire image des sorties	Sortie (bit)	A	Pendant le cycle, le programme calcule les valeurs pour les sorties et les dépose dans la mémoire image des sorties. A la fin du cycle, la CPU écrit les valeurs de sortie calculées dans les modules de sorties.
	Octet de sortie	AB	
	Mot de sortie	AW	
	Double mot de sortie	AD	
Mémentos	Mémento (bit)	M	Cette zone met à disposition de l'espace mémoire pour les résultats intermédiaires calculés dans le programme.
	Octet de memento	MB	
	Mot de memento	MW	
	Double mot de memento	MD	
Temporisations	Temporisation (T)	T	Cette zone sert d'espace mémoire pour les temporisations.
Compteur	Compteur (Z)	Z	Cette zone sert d'espace mémoire pour les compteurs.
Bloc de données	Bloc de données ouvert avec AUF DB :	DB	Les blocs de données contiennent des informations pour le programme. Ils peuvent soit servir à tous les blocs de code (DB globaux), soit être associés à un FB ou à un SFB spécifique (DB d'instance).
	Bit de données	DBX	
	Octet de données	DBB	
	Mot de données	DBW	
	Double mot de données	DBD	
	Bloc de données ouvert avec AUF DB :	DI	
	Bit de données	DIX	
	Octet de données	DIB	
	Mot de données	DIW	

Plage d'opérandes	Accès par des unités de taille suivante	Notation S7	Description
	Double mot de données	DID	
Données locales	Bit de données locales	L	Cette zone fournit de l'espace mémoire aux données temporaires d'un bloc pour la durée du traitement de ce bloc. La pile L sert également à la transmission de paramètres de blocs et à l'enregistrement de résultats intermédiaires pour les réseaux CONT.
	Octet de données locales	LB	
	Mot de données locales	LW	
	Double mot de données locales	LD	
Zone de périphérie : entrées	Octet d'entrée de périphérie	PEB	Les zones de périphérie des entrées et des sorties permettent l'accès direct à des modules d'entrées et de sorties centralisés et décentralisés.
	Mot d'entrée de périphérie	PEW	
	Double mot d'entrée de périphérie	PED	
Zone de périphérie : sorties	Octet de sortie de périphérie	PAB	
	Mot de sortie de périphérie	PAW	
	Double mot de sortie de périphérie	PAD	

Vous trouverez les zones d'adresses autorisées pour votre CPU dans la description de CPU ou dans la liste d'opérations suivante :

- manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*,
- liste d'opérations *Automate programmable S7-300*.

A.2.3.2 Mémoire image des entrées/sorties

Lorsque le programme utilisateur accède aux zones d'opérandes Entrées (E) et Sorties (A), il n'interroge pas les états des signaux sur les modules TOR, mais accède à une zone dans la mémoire système de la CPU et de la périphérie décentralisée. On appelle cette zone de mémoire "mémoire image du processus".

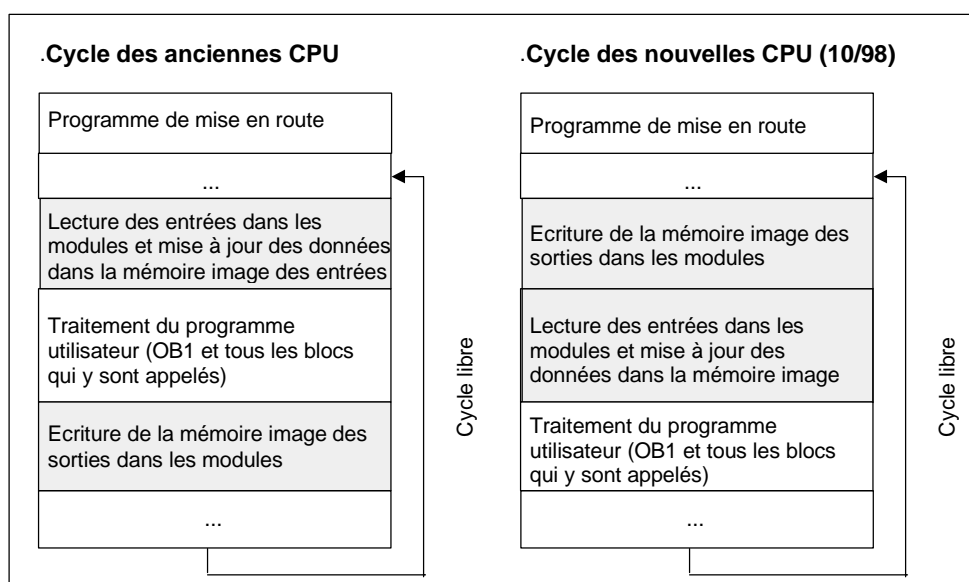
La mémoire image du processus se compose de deux parties : la mémoire image des entrées (MIE) et la mémoire image des sorties (MIS).

Condition d'accès à la mémoire image

La CPU ne peut accéder qu'à la mémoire image des modules que vous avez configurés avec STEP 7 Lite.

Mise à jour de la mémoire image du processus

Le système d'exploitation actualise la mémoire image du processus cycliquement. La figure suivante représente les étapes de traitement au cours d'un cycle ; elles ne sont pas les mêmes pour les anciennes CPU et pour celles qui sont disponibles depuis 10/98.



Avantages de la mémoire image du processus

Par rapport à l'accès direct aux modules d'entrées et de sorties, l'accès à la mémoire image du processus offre l'avantage que la CPU dispose d'une mémoire image cohérente des signaux du processus pendant l'exécution cyclique du programme. Si un état de signal change sur un module d'entrées pendant le traitement du programme, l'état de signal dans la mémoire image est conservé jusqu'à la mise à jour de la mémoire image dans le cycle suivant. En outre, l'accès à la mémoire image prend bien moins de temps que l'accès direct aux modules de signaux, car la mémoire image du processus se trouve dans la mémoire interne de la CPU.

Erreur d'accès à la périphérie lors de la mise à jour de la mémoire image

Pour la famille de CPU qui nous occupe (S7-300), la réaction prééglée à une erreur durant la mise à jour de la mémoire image est la suivante :

- pas d'inscription dans le tampon de diagnostic, pas d'appel d'OB, les octets d'entrée/sortie concernés sont mis à 0.

Avec les nouvelles CPU (à partir de 4/99), vous pouvez modifier par paramétrage la réaction aux erreurs d'accès à la périphérie :

- l'OB85 ne démarre que pour une erreur d'accès à la périphérie apparaissant et disparaissant et génère une entrée dans le tampon de diagnostic,
- pas d'appel de l'OB85 (comportement par défaut des S7-300).

Nombre de démarrages de l'OB85

En plus de la réaction paramétrée aux erreurs d'accès à la périphérie (apparaissant/disparaissant ou à chaque accès à la périphérie), la plage d'adresses d'un module a aussi une influence sur le nombre de démarrages de l'OB85.

Pour un module dont la plage d'adresses va jusqu'au double mot, l'OB85 démarre une fois : par exemple pour un module TOR possédant jusqu'à 32 entrées ou sorties ou pour un module analogique à 2 voies.

Pour les modules dont la plage d'adresses est plus grande, l'OB85 démarre autant de fois qu'il est nécessaire d'accéder à la plage avec des opérations sur double mot : par exemple deux fois pour un module analogique à 4 voies.

A.2.3.3 Pile des données locales

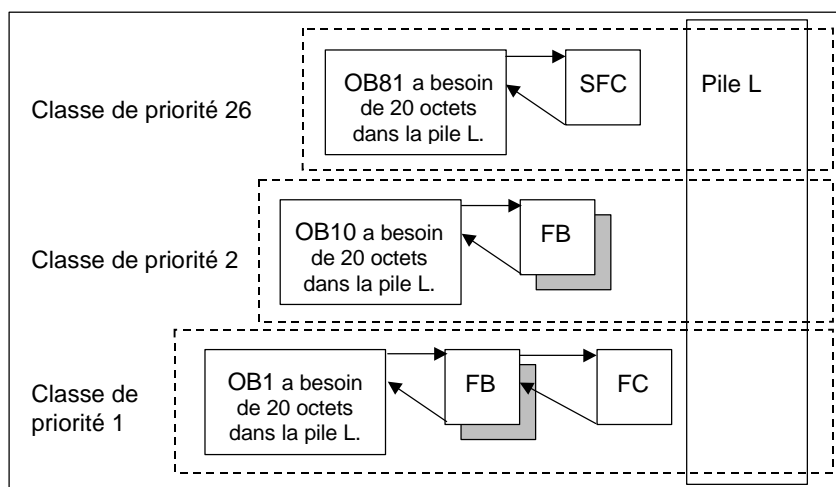
La pile L enregistre :

- les variables temporaires des données locales de blocs,
- les informations de déclenchement des blocs d'organisation,
- des informations pour la transmission de paramètres,
- des résultats intermédiaires dans les programmes CONT.

Quand vous créez des blocs d'organisation, vous pouvez déclarer des variables temporaires (TEMP) disponibles uniquement pendant l'exécution du bloc et écrasées ensuite. Les données locales doivent être initialisées avant le premier accès. Chaque bloc d'organisation nécessite, en outre, 20 octets de données locales pour ses informations de déclenchement.

La CPU possède une mémoire limitée pour les variables temporaires (données locales) des blocs en cours de traitement. La taille de cette zone de mémoire, appelée pile des données locales, dépend de la CPU. Par défaut, elle est partagée également entre les différentes classes de priorité. Ainsi, chaque classe de priorité dispose d'une zone de données locales en propre. Cela garantit que même les classes de priorité les plus élevées avec leurs OB associés ont assez de place pour leurs données locales.

La figure suivante illustre l'affectation de données locales aux classes de priorité à l'aide d'un exemple dans lequel l'OB1 est interrompu par l'OB10 dans la pile L, l'OB10 étant interrompu à son tour par l'OB81.



Avertissement

Toutes les variables temporaires (TEMP) d'un OB et des blocs qui y sont appelés sont enregistrées dans la pile L. Cette dernière peut déborder lorsque vous imbriquez trop de niveaux dans votre traitement des blocs.

Les CPU S7 passent à l'état "Arrêt" (STOP) lorsque vous dépassez la taille de pile L autorisée pour un programme.

Nous vous conseillons donc de tester la pile L (les variables temporaires) dans votre programme.

Tenez compte de la mémoire requise par les données locales des OB d'erreur synchrone.

Affectation de données locales aux classes de priorité

Pour les CPU S7-300, un nombre fixe de données locales (256 octets) est affecté à chaque classe de priorité. Vous ne pouvez pas le modifier.

A.2.3.4 Pile des interruptions

Lorsque le traitement du programme est interrompu par un OB de priorité plus élevée, le système d'exploitation enregistre, dans la pile des interruptions (pile I), le contenu des accumulateurs et des registres d'adresse ainsi que le numéro et la taille des blocs de données ouverts.

A l'achèvement du traitement du nouvel OB, le système d'exploitation charge les informations contenues dans la pile I et reprend le traitement du bloc interrompu au point où s'était produite l'interruption.

A l'état de fonctionnement "Arrêt" (STOP), vous pouvez lire la pile des interruptions à la PG à l'aide de STEP 7 Lite. Vous trouverez ainsi plus facilement la cause du passage de la CPU à l'état "Arrêt".

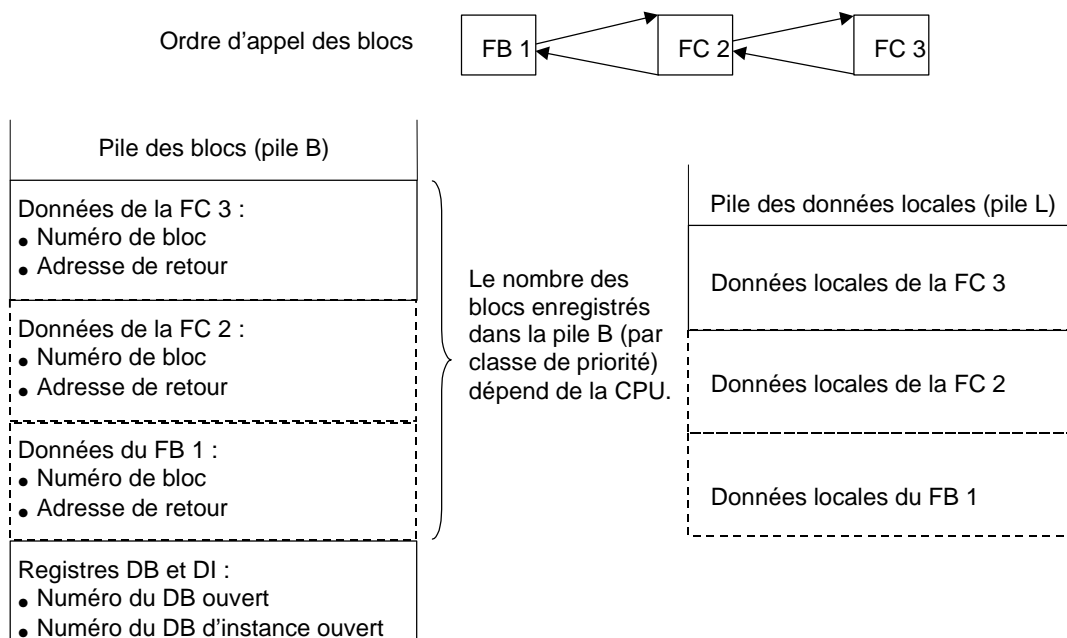
A.2.3.5 Pile des blocs

Lorsque le traitement d'un bloc est interrompu par l'appel d'un autre bloc ou par une classe de priorité plus élevée (alarme/traitement d'erreur), la pile B enregistre les données suivantes :

- numéro, type (OB, FB, FC, SFB, SFC) et adresse de retour du bloc interrompu,
- numéro des blocs de données (des registres DB et DI) ouverts au moment de l'interruption.

Ces informations permettent de poursuivre l'exécution du programme utilisateur après l'interruption.

Si la CPU est à l'état de fonctionnement "Arrêt" (STOP), vous pouvez afficher la pile des blocs sur la console de programmation à l'aide de STEP 7 Lite. Cette pile énumère tous les blocs dont le traitement n'était pas terminé au moment où la CPU est passée à l'état "Arrêt". Ces blocs sont listés dans l'ordre dans lequel leur traitement avait commencé (voir la figure ci-après).



Registres de bloc de données

Il existe deux registres de bloc de données ; ils contiennent les numéros des blocs de données ouverts.

- Le registre DB contient le numéro du bloc de données global ouvert.
- Le registre DI contient le numéro du bloc de données d'instance ouvert.

A.2.3.6 Mémoire tampon de diagnostic

Dans la mémoire tampon de diagnostic, les messages de diagnostic sont affichés dans l'ordre de leur apparition. La première entrée contient l'événement le plus récent. Le nombre des entrées dans la mémoire tampon de diagnostic dépend du module et de son état de fonctionnement en cours.

Parmi les événements de diagnostic, on trouve :

- erreur dans un module,
- erreur d'assignation du processus
- erreur système dans la CPU,
- changements d'état de fonctionnement de la CPU
- erreurs dans le programme utilisateur,
- événements de diagnostic personnalisés (via la fonction système SFC52).

A.2.3.7 Exploitation de la mémoire tampon de diagnostic

Une partie de la liste d'état système est constituée de la mémoire tampon de diagnostic. Des informations détaillées y sont inscrites sur les événements de diagnostic système et de diagnostic personnalisé, dans l'ordre de leur apparition. Les informations inscrites en cas d'événement de diagnostic système sont identiques aux informations de déclenchement transmises au bloc d'organisation approprié.

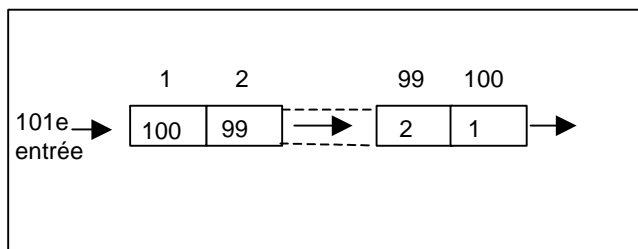
Il n'est pas possible d'effacer les entrées dans la mémoire tampon de diagnostic ; son contenu est conservé même après un effacement général.

La mémoire tampon de diagnostic permet :

- en cas d'arrêt de l'installation, d'évaluer les derniers événements précédant le passage à l'arrêt (STOP) et de trouver la cause de l'arrêt,
- de déceler plus rapidement les causes d'erreur et d'améliorer ainsi la disponibilité de l'installation,
- d'évaluer et d'optimiser le comportement dynamique de l'installation.

Organisation de la mémoire tampon de diagnostic

La mémoire tampon de diagnostic est conçue comme mémoire circulante pour un nombre maximal d'entrées dépendant du module. Lorsque ce nombre maximal est atteint, un nouvel événement écrase le plus ancien et toutes les entrées sont déplacées en conséquence. Ainsi, l'entrée la plus récente est toujours à la première place. La mémoire tampon de diagnostic de la CPU 314 S7-300, par exemple, peut contenir 100 entrées :



Le nombre d'entrées affichées dans la mémoire tampon de diagnostic dépend du module et de son état de fonctionnement en cours. Pour certaines CPU, la longueur de la mémoire tampon de diagnostic est paramétrable.

Contenu de la mémoire tampon de diagnostic

La zone de liste **supérieure** énumère tous les événements de diagnostic qui se sont produits, avec les informations suivantes :

- numéro d'ordre de l'entrée (l'événement le plus récent a le numéro 1) ;
- heure et date de l'événement de diagnostic : c'est l'heure et la date du module qui sont indiqués si ce dernier possède une horloge ; pour pouvoir utiliser correctement ces indications horaires, il est donc important de régler la date et à l'heure du module et de les vérifier de temps à autre ;
- texte abrégé de l'événement.

La zone de liste **inférieure** affiche des informations supplémentaires sur l'événement sélectionné dans la zone supérieure. Ce sont, par exemple :

- numéro de l'événement,
- désignation de l'événement,
- changement d'état de fonctionnement causé par l'événement de diagnostic,
- renvoi à la position de l'erreur dans un bloc (type et numéro de bloc et adresse relative) qui a causé l'inscription de l'événement,
- événement arrivant ou partant,
- informations complémentaires spécifiques à l'événement.

En cliquant sur le bouton "A propos de l'événement", vous pouvez afficher des informations complémentaires sur l'événement sélectionné dans la liste.

Vous trouverez des explications sur les ID d'événement dans l'aide sur les fonctions système et les blocs fonctionnels système (Sauts dans les descriptions des langages et les aides sur les blocs).

Enregistrement du contenu dans un fichier de texte

Le bouton "Enregistrer sous" dans la page d'onglet "Mémoire tampon de diagnostic" de la boîte de dialogue "Etat du module" permet d'enregistrer le contenu du tampon de diagnostic sous forme de texte ASCII.

Lecture de la mémoire tampon de diagnostic

Vous pouvez afficher le contenu de la mémoire tampon de diagnostic sur la PG/le PC via la page d'onglet "Mémoire tampon de diagnostic" de la boîte de dialogue "Etat du module", ou en effectuer la lecture dans un programme à l'aide de la SFC51 RDSYSST.

Dernière entrée avant arrêt

Vous pouvez demander que la dernière entrée de la mémoire tampon de diagnostic avant le passage de l'état "Marche" (RUN) à l'état "Arrêt" (STOP) soit automatiquement envoyée à un appareil de contrôle déclaré (PG, OP, TD, par exemple). Cela permet de localiser et de corriger plus rapidement la cause du passage à l'état "Arrêt".

A.2.3.8 Zones de mémoire rémanentes des CPU S7-300

Rémanence générale

La mémoire tampon de diagnostic, les paramètres MPI et le compteur d'heures de fonctionnement sont de manière générale rémanents. Ces données sont conservées aussi bien après une coupure de courant qu'après un effacement général.

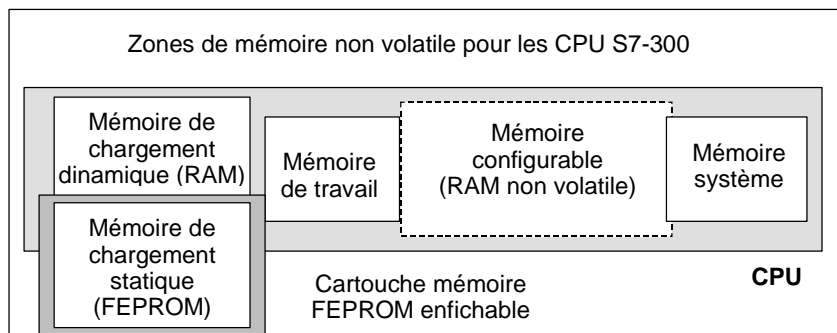
Sauvegarde des données dans les CPU 31x

Lorsque la mémoire (mémoire de chargement dynamique (RAM), mémoire de travail et mémoire système) d'une CPU S7-300 n'est pas sauvegardée, toutes les données enregistrées dans ces zones sont perdues en cas de coupure de courant. Les CPU S7-300 fournissent toutefois des moyens pour conserver programme et données.

- Vous pouvez sauvegarder, à l'aide d'une pile, toutes les données se trouvant en mémoire de chargement, en mémoire de travail et dans certaines parties de la mémoire système.
- Vous pouvez enregistrer votre programme dans l'EPROM (soit sous forme de carte mémoire, soit intégrée dans la CPU ; voir le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*)
- Vous pouvez mémoriser un volume de données dépendant de la CPU dans une zone de mémoire vive non volatile (NVRAM).

Mémoire vive non volatile

Votre CPU S7-300 comporte une zone de mémoire vive non volatile (NVRAM ; voir la figure ci-après). Si vous avez rangé votre programme dans l'EPROM de la mémoire de chargement, vous pouvez aussi enregistrer certaines données - en cas de coupure de courant ou de passage de l'état de fonctionnement "Arrêt" (STOP) à l'état "Marche" (RUN) – grâce à la configuration appropriée.



A cet effet, vous réglez votre CPU de manière à enregistrer les données suivantes en mémoire vive non volatile :

- informations enregistrées dans un DB (utile uniquement si vous avez aussi stocké votre programme dans une EPROM de la mémoire de chargement),
- valeurs de temporisations et de compteurs,
- informations enregistrées dans des mémentos.

Chaque CPU vous permet de sauvegarder un nombre précis de temporisations, de compteurs et de mémentos. Elle met en outre à disposition un nombre spécifique d'octets autorisant l'enregistrement des données stockées dans des DB.

Utilisation d'une pile de sauvegarde

Avec une pile de sauvegarde, la mémoire de chargement et la mémoire de travail deviennent rémanentes en cas de coupure de courant. Si votre configuration implique la sauvegarde en mémoire vive non volatile des temporisations, compteurs et mémentos, ces informations sont également conservées, indépendamment de la sauvegarde par pile.

Configuration des données de la mémoire vive non volatile

Vous déterminez les zones de mémoire rémanentes lors de la configuration de la CPU avec STEP 7 Lite.

La taille de mémoire pouvant être configurée en mémoire vive non volatile dépend de la CPU ; vous ne pouvez pas sauvegarder plus de données que le volume précisé pour votre CPU.

Sauvegarde des données des CPU 31xC ("CPU compactes")

La mémoire de chargement de la CPU 31xC est totalement intégrée à une carte mémoire Micro (MMC). Sa taille correspond exactement à celle de la carte mémoire MMC.

Le chargement de programmes et le fonctionnement des CPUs 31xC sont uniquement possibles lorsque la carte mémoire MMC est enfichée.

Objets rémanents :

- programme utilisateur dans la mémoire de chargement (sur la carte MMC),
- mementos, temporisations et compteurs paramétrés comme rémanents (dans la mémoire système),
- contenu des blocs de données (enregistrés sur la carte MMC par la mémoire de travail en cas de coupure de courant).

A.3 Types de données et de paramètre

A.3.1 Introduction aux types de données et de paramètre

Il faut indiquer le type de données pour toutes les données utilisées dans un programme utilisateur. On distingue entre :

- les types de données simples mis à votre disposition par STEP 7 Lite,
- les types de données complexes que vous pouvez créer en combinant des types de données simples
- et les types de paramètre avec lesquels vous définissez des paramètres à transmettre à des FB ou à des FC.

Généralités

Les opérations LIST, LOG ou CONT utilisent des objets de données de taille définie. Les opérations combinatoires sur bit, par exemple, utilisent des bits. Les opérations de chargement et de transfert LIST ainsi que les opérations de transfert LOG et CONT utilisent des octets, mots et doubles mots.

Un bit est un chiffre binaire "0" ou "1". Un octet comporte 8 bits, un mot 16 bits et un double mot 32 bits.

Les opérations arithmétiques utilisent également des octets, des mots ou des doubles mots. Dans ces opérandes de niveau octet, mot ou double mot, vous pouvez coder des nombres de formats différents, comme par exemple des nombres entiers et des nombres à virgule flottante.

Si vous utilisez l'adressage symbolique, vous définissez des mnémoniques et leur affectez un type de données (voir le tableau suivant). Les différents types de données possèdent plusieurs options pour le format et diverses représentations de nombre.

Le présent chapitre ne décrit que certaines des notations possibles pour les nombres et les constantes. Le tableau suivant indique les formats de nombres et de constantes qui ne seront pas abordés en détail.

Format	Taille en bits	Représentation des nombres
Hexadécimal	8, 16 et 32	B#16#, W#16# et DW#16#
Binaire	8, 16 et 32	2#
Date CEI	16	D#
Durée CEI	32	T#
Heure	32	TOD#
Caractère	8	'A'

A.3.2 Types de données simples

Chaque type de données simple a une longueur définie. Le tableau ci-après présente les types de données simples.

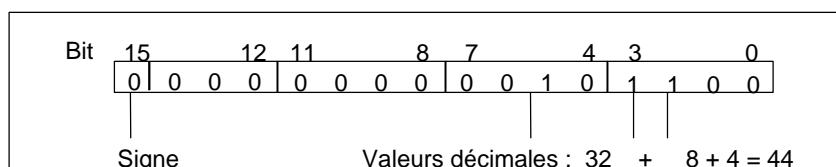
Type et description	Taille en bits	Options pour format	Plage et représentation des nombres (valeur mini à valeur maxi)	Exemple
BOOL (bit)	1	Texte booléen	TRUE/FALSE	TRUE
BYTE (octet)	8	Nombre hexadécimal	B#16#0 à B#16#FF	L B#16#10 L byte#16#10
WORD (mot)	16	Nombre en binaire pur Nombre hexadécimal BCD Nbre décimal non signé	2#0 à 2#1111_1111_1111_1111 W#16#0 à W#16#FFFF C#0 à C#999 B#(0,0) à B#(255,255)	L 2#0001_0000_0000_0000 L W#16#1000 L word#16#1000 L C#998 L B#(10,20) L byte#(10,20)
DWORD (double mot)	32	Nombre en binaire pur Nombre hexadécimal Nbre décimal non signé	2#0 à 2#1111_1111_1111_1111_1111_1111_1111_1111 DW#16#0000_0000 à DW#16#FFFF_FFFF B#(0,0,0,0) à B#(255,255,255,255)	2#1000_0001_0001_1000_1011_1011_0111_1111 L DW#16#00A2_1234 L dword#16#00A2_1234 L B#(1, 14, 100, 120) L byte#(1,14,100,120)
INT (entier)	16	Nombre décimal signé	-32768 à 32767	L 1
DINT (nbre entier de 32 bits)	32	Nombre décimal signé	L#-2147483648 à L#2147483647	L L#1
REAL (nombre à virgule flottante)	32	IEEE nombre à virgule flottante	Limite supérieure : $\pm 3.402823e+38$ 0 Limite inférieure : $\pm 1.175495e-38$	L 1.234567e+13
S5TIME (durée SIMATIC)	16	Durée S7 en pas de 10 ms (valeur par défaut)	S5T#0H_0M_0S_10MS à S5T#2H_46M_30S_0MS et S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (durée CEI)	32	Durée CEI en pas de 1 ms, entier signé	-T#24D_20H_31M_23S_648MS à T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS
DATE (date CEI)	16	Date CEI en pas de 1 jour	D#1990-1-1 à D#2168-12-31	L D#1994-3-15 L DATE#1994-3-15
TIME_OF_DAY (heure)	32	Heure en pas de 1 ms	TOD#0:0:0.0 à TOD#23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3
CHAR (caractère)	8	Caractères ASCII	'A','B' etc.	L 'E'

A.3.2.1 Format du type de données INT (entiers de 16 bits)

Un nombre entier comporte un signe précisant s'il s'agit d'un entier positif ou négatif. La place occupée par un nombre entier (16 bits) dans la mémoire est d'un mot. Le tableau suivant représente la plage d'un nombre entier (16 bits).

Format	Plage
Entier (16 bits) :	-32 768 à +32 767

La figure suivante représente le nombre entier +44 sous forme de nombre en binaire pur.

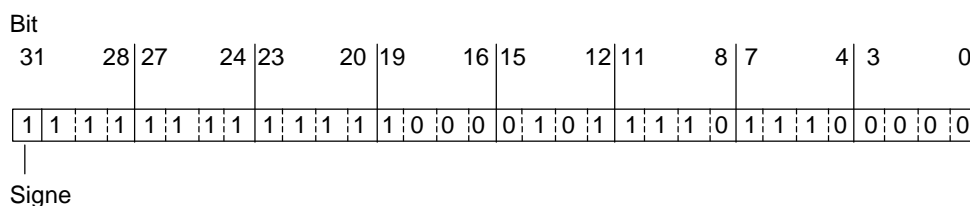


A.3.2.2 Format du type de données DINT (entiers de 32 bits)

Un nombre entier comporte un signe précisant s'il s'agit d'un entier positif ou négatif. La place occupée par un nombre entier (32 bits) dans la mémoire est de deux mots. Le tableau suivant représente la plage d'un nombre entier (32 bits).

Format	Plage
Nombre entier (32 bits) :	-2 147 483 648 à +2 147 483 647

La figure suivante représente le nombre entier -500 000 comme nombre en binaire pur. Dans le système binaire, la forme négative d'un nombre entier est représentée comme complément à deux du nombre entier positif. Vous obtenez le complément à deux d'un nombre entier en inversant les états de signaux de tous les bits, puis en additionnant +1 au résultat.



A.3.2.3 Format du type de données REAL (nombres à virgule flottante)

La représentation générale d'un nombre à virgule flottante est "nombre = $m * b^{\text{exposant } E}$ ". La base "b" et l'exposant "E" sont des nombres entiers, la mantisse "m" un nombre rationnel.

Ce type de représentation des nombres permet de représenter de très grandes et de très petites valeurs dans un espace limité. Le nombre limité de bits pour la mantisse et l'exposant permet de représenter une vaste plage de nombres.

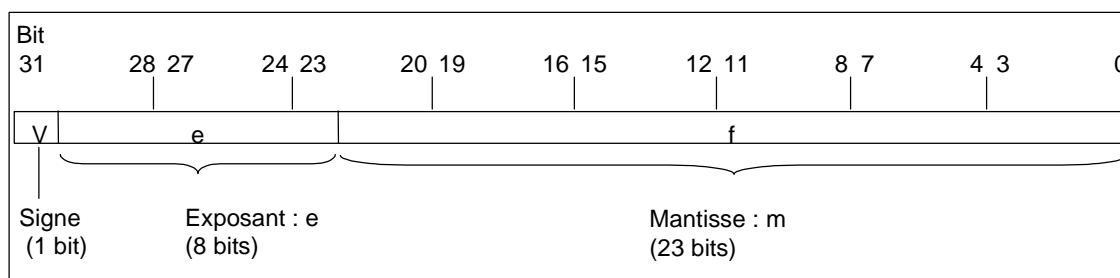
L'inconvénient, c'est la précision limitée du calcul. Pour calculer la somme de deux nombres, par exemple, les exposants doivent être adaptés par décalage de la mantisse (virgule flottante) (addition des mantisses de deux nombres possédant le même exposant).

Format virgule flottante dans STEP 7 Lite

Dans STEP 7 Lite, les nombres à virgule flottante correspondent au format de base de simple largeur, comme décrit dans la norme ANSI/IEEE Std 754-1985, *IEEE Standard for Binary Floating-Point Arithmetic*. Ils sont formés des composants suivants :

- le signe s,
- l'exposant e augmenté d'une constante (Bias = +127) : $e = E + \text{Bias}$,
- la partie fractionnaire de la mantisse m.
La partie entière de la mantisse n'est pas indiquée, car elle est toujours égale à 1 dans la plage de nombres valide.

Ces trois composants occupent au total un double mot (32 bits) :



Le tableau suivant illustre la valeur de chaque bit dans le format virgule flottante.

Composant du nombre à virgule flottante	Numéro de bit	Valeur
Signe s	31	
Exposant e	30	2 exposant 7
...
Exposant e	24	2 exposant 1
Exposant e	23	2 exposant 0
Mantisse m	22	2 exposant -1
...
Mantisse m	1	2 exposant -22
Mantisse m	0	2 exposant -23

Les trois composants **s**, **e** et **m** définissent la valeur d'un nombre représenté dans ce format par la formule :

Nombre = $1.m \times 2^{\text{exposant} (e - \text{Bias})}$

sachant que :

- $e : 1 \leq e \leq 254$
- Bias : Bias = 127. Ceci permet d'éviter un signe supplémentaire pour l'exposant.
- s : pour un nombre positif, $s = 0$ et pour un nombre négatif, $s = 1$.

Plage de valeurs des nombres à virgule flottante

Avec le format virgule flottante représenté ci-dessus :

- le nombre à virgule flottante le plus petit = $1.0 \times 2^{\text{exposant} (1-127)} = 1.0 \times 2^{\text{exposant} (-126)} = 1.175\,495\text{E}-38$ et
- le nombre à virgule flottante le plus grand = $2 \times 2^{\text{exposant} (-23)} \times 2^{\text{exposant} (254-127)} = 2 \times 2^{\text{exposant} (-23)} \times 2^{\text{exposant} (+127)} = 3.402\,823\text{E}+38$

Le nombre zéro est représenté par $e = m = 0$; $e = 255$ et $m = 0$ signifie "infini".

Format	Plage ¹⁾
Nombres à virgule flottante selon la norme ANSI/IEEE	$-3.402\,823\text{E}+38$ à $-1.175\,495\text{E}-38$ et 0 et $+1.175\,495\text{E}-38$ à $+3.402\,823\text{E}+38$

Le tableau suivant représente l'état logique des bits du mot d'état pour les résultats d'opérations sur des nombres à virgule flottante se trouvant hors de la plage admise.

Résultat dans la plage invalide	BI1	BI0	DEB	DM
$-1.175494\text{E}-38 < \text{résultat} < -1.401298\text{E}-45$ (nombre négatif) Dépassement bas	0	0	1	1
$+1.401298\text{E}-45 < \text{résultat} < +1.175494\text{E}-38$ (nombre positif) Dépassement bas	0	0	1	1
Résultat $< -3.402823\text{E}+38$ (nombre négatif) Dépassement haut	0	1	1	1
Résultat $> 3.402823\text{E}+38$ (nombre positif) Dépassement haut	1	0	1	1
Pas de nombre à virgule flottante valide ou opération invalide (valeur d'entrée hors de la plage de valeurs admise)	1	1	1	1

Attention pour les opérations mathématiques :

On obtient par exemple le résultat "Pas de nombre à virgule flottante valide" lorsqu'on tente d'extraire la racine carrée de -2 . Dans le cas d'opérations mathématiques, vous devez donc toujours évaluer d'abord les bits d'état avant de poursuivre le calcul avec le résultat.

Attention pour le "Forçage de variables" :

Lorsqu'on inscrit les valeurs pour les opérations sur nombres à virgule flottante dans des doubles mots de memento, par exemple, il est possible de modifier ces valeurs avec des modèles binaires quelconques. Chaque modèle binaire ne représente cependant pas un nombre valide !

Précision dans le calcul sur nombres à virgule flottante



Avertissement

Les calculs volumineux sur des nombres présentant des ordres de grandeur très différents (plusieurs 10^{aines} de puissances) peuvent donner des résultats imprécis

Dans STEP 7 Lite, la précision des nombres à virgule flottante est de 6 décimales. Lorsque vous indiquez des constantes à virgule flottante, vous êtes donc limité à 6 décimales au plus.

Nota

La précision de calcul de 6 décimales signifie par exemple que l'addition du nombre1 + nombre2 = nombre1, lorsque nombre1 est supérieur à nombre2 * 10 exposant y, avec $y > 6$:

$$100\,000\,000 + 1 = 100\,000\,000.$$

Exemples de nombres représentés en format virgule flottante

La figure suivante montre le format en virgule flottante des valeurs décimales suivantes :

- 10,0
- p (3,141593)
- racine carrée de 2 ($p2 = 1,414214$)

Le nombre **10.0** du premier exemple résulte de son format virgule flottante (représentation en hexa : 4120 0000) de la manière suivante :

$$e = 2 \text{ exposant } 1 + 2 \text{ exposant } 7 = 2 + 128 = 130$$

$$m = 2 \text{ exposant } (-2) = 0,25$$

Ce qui donne : $1.m * 2^{\text{exposant } (e - \text{Bias})} = 1.25 * 2^{\text{exposant } (130 - 127)} = 1.25 * 2^{\text{exposant } 3} = 10.0.$

Valeur décimale 10,0

Valeur hexadécimale

	4	1	2	0	0	0	0	0								
Bits	31	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0

0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Signe de la
mantisse : s
(1 bit)

Exposant : e
(8 bits)

$$e = 27 + 21 = 130$$

$$1.f_2e\text{-bias} = 1,25_23 = 10,0$$

$$[1,25_2(130-127) = 1,25_23 = 10,0]$$

Mantisse ou fraction : f
(23 bits)

$$f = 2-2 = 0.25$$

Valeur décimale 3,141593

Valeur hexadécimale

	4		0		4		9		0		F		D		C	
Bits	31	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0

0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Signe de la mantisse : s
(1 bit)

Exposant : e
(8 bits)

Mantisse ou fraction : f
(23 bits)

Valeur décimale 1,414214

Valeur hexadécimale

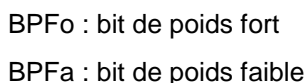
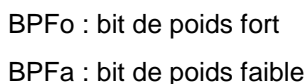
	3			F		B		5		0		4		F		7	
Bits	31	28	27	24	23	20	19	16	15	12	11	8	7	4	3	0	

0	0	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	0	0	0	1	0	0	1	1	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Signe de la mantisse : s
(1 bit)

Exposant : e
(8 bits)

Mantisse ou
fraction : f
(23 bits)



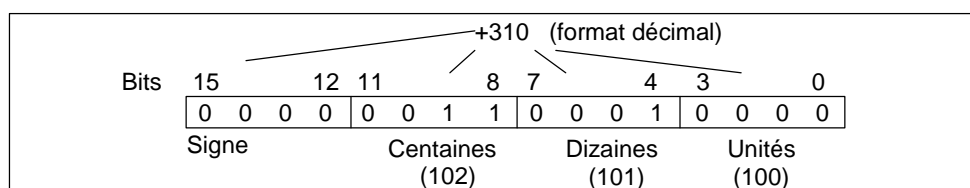
A.3.2.6 Format des types de données WORD et DWORD pour les nombres décimaux codés binaire

Dans la représentation décimale codée binaire (DCB), un nombre décimal est représenté par des groupes de chiffres binaires (bits). Un groupe de 4 bits représente un chiffre ou le signe d'un nombre décimal. Les groupes de 4 bits forment un mot (16 bits) ou un double mot (32 bits). Les quatre bits de poids fort indiquent le signe du nombre ("1111" signifie moins et "0000" plus). Les instructions comportant des opérandes décimaux codés binaires n'exploitent que le bit de poids fort (15 en format mot, 31 en format double mot). Le tableau suivant indique le format et la plage des deux types de nombre DCB.

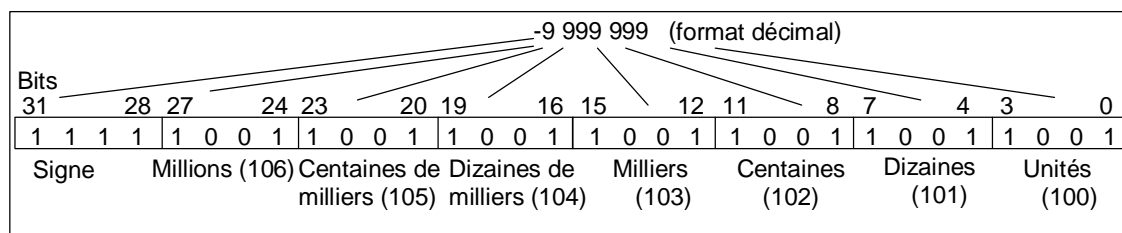
Format	Plage
Mot (16 bits, nombre DCB à 3 positions signé)	–999 à +999
Double mot (32 bits, nombre DCB à 7 positions signé)	–9 999 999 à +9 999 999

Les figures suivantes donnent des exemples de nombre décimal codé binaire dans les formats suivants :

- Format mot

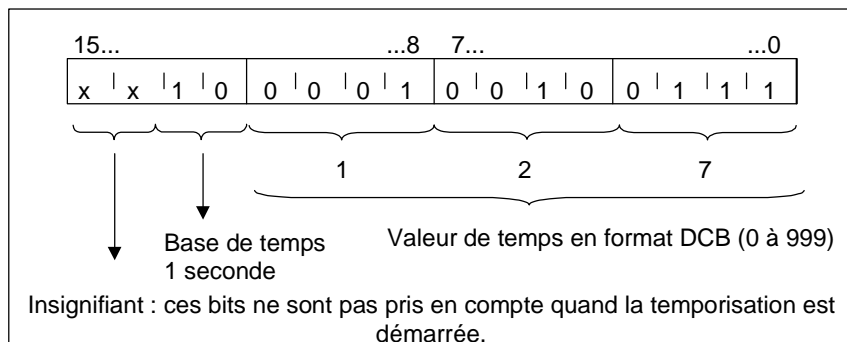


- Format double mot



A.3.2.7 Format du type de données S5TIME (durée)

Lorsque vous saisissez la durée avec le type de données S5TIME, vos entrées sont enregistrées en format DCB. La figure suivante indique le contenu de l'opérande de temporisation pour une valeur de temps égale à 127 et une base de temps d'1 s.



Quand vous utilisez le type de données S5TIME, vous saisissez une valeur de temps comprise entre 0 et 999 et vous spécifiez une base de temps (voir tableau ci-après). La base de temps indique l'intervalle selon lequel une temporisation diminue la valeur de temps d'une unité jusqu'à ce qu'elle soit "0".

Base de temps pour S5TIME

Base de temps	Code binaire pour la base de temps
10 ms	00
100 ms	01
1 s	10
10 s	11

Vous pouvez charger une valeur de temps prédéfinie en utilisant la syntaxe suivante :

- L¹ W#16#wxyz
 - avec : w = base de temps (c'est-à-dire intervalle de temps ou résolution),
 - xyz = valeur de temps en format DCB.
- L¹ S5T#aH_bbM_ccS_dddMS
 - avec : a = heures, bb = minutes, cc = secondes et ddd = millisecondes.
 - La base de temps est automatiquement sélectionnée et la valeur est arrondie au nombre inférieur le plus proche avec cette base de temps.

Vous pouvez saisir une valeur de temps de 9 990 secondes ou 2H_46M_30S au maximum.

¹ = indiquer L uniquement en programmation LIST

A.3.2.8 Format du type de données TIME

Type de données	Longueur (bits)	Format
TIME	32	Durée avec signe : + ou - jours, heures, minutes, secondes, millisecondes

Exemples de format (valeurs limites supérieure et inférieure)

Maxi	T#+24d20h31m23s647ms
Mini	T# -24d20h31m23s648ms

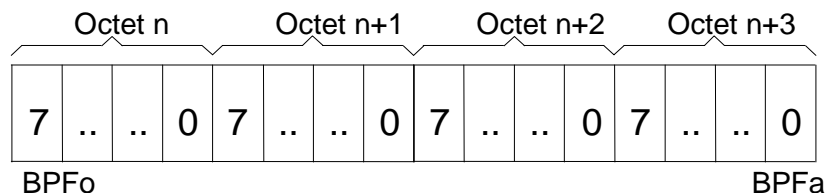
Il n'est pas nécessaire d'indiquer toutes les unités (ex. : T#5h10s est correct).

Lorsque vous indiquez une seule unité, la valeur absolue en jours, heures et minutes ne doit pas dépasser les valeurs limites supérieure et inférieure.

T# -65535 et T#+65535 sont les valeurs limites supérieure et inférieure pour les secondes et les millisecondes.

Lorsque vous indiquez plusieurs unités de temps, elles ne doivent pas dépasser les valeurs suivantes :

- 23 pour les heures,
- 59 pour les minutes,
- 59 pour les secondes,
- 999 pour les millisecondes.



BPFO : bit de poids fort

BPFa : bit de poids faible

Nota

Le type de données TIME est enregistré en tant que complément à deux sous forme de nombre entier en millisecondes avec signe.

A.3.3 Types de données complexes

Les types de données complexes définissent des groupes de données comportant plus de 32 bits ou des groupes de données composés à partir d'autres types de données.

STEP 7 Lite autorise les types de données complexes suivants :

- DATE_AND_TIME,
- STRING,
- ARRAY (tableau),
- STRUCT (structure),
- UDT (types de données utilisateur),
- FB et SFB.

Le tableau ci-après décrit les types de données complexes. Vous définissez les structures et les tableaux soit dans la déclaration des variables du bloc de code, soit dans un bloc de données.

Type de données	Description
DATE_AND_TIME DT	Définit une zone de 64 bits (8 octets). Ce type de données enregistre en format décimal codé binaire.
STRING	Définit un ensemble de 254 caractères au maximum (type de données CHAR). La zone réservée à une chaîne de caractères est par défaut de 256 octets : c'est la mémoire nécessaire à l'enregistrement de 254 octets et d'un en-tête de 2 octets. Vous pouvez préciser aussi le nombre de caractères à enregistrer dans la chaîne et réduire ainsi l'espace utilisé en mémoire (ex. : string[9] 'Siemens').
ARRAY	Définit un agrégat multidimensionnel d'un même type de données (soit simple, soit complexe). Par ex. "ARRAY[1..2,1..3] OF INT" correspond à un tableau de nombres entiers de format 2 x 3. Vous accédez aux données enregistrées dans un tableau via l'indice (ex. : [2,2]). Un tableau peut comporter 6 dimensions au maximum ; l'indice peut être un nombre entier quelconque (de -32768 à 32767).
STRUCT	Définit un agrégat de types de données quelconques combinés. Vous pouvez, par exemple, définir un tableau de structures ou une structure contenant structures et tableaux.
UDT	Un UDT vous permet d'organiser des volumes de données importants et de simplifier la saisie des types de données lorsque vous voulez créer des blocs de données ou déclarer des variables dans la table de déclaration des variables. Dans STEP 7 Lite, vous pouvez combiner des types de données simples et complexes et créer ainsi votre propre type de données. Les UDT ont un nom propre et peuvent donc être utilisés plusieurs fois.
FB, SFB	Déterminent la structure du bloc de données d'instance associé et permettent la transmission de données d'instance pour plusieurs appels de FB dans un DB d'instance.

Les types de données complexes sont stockés par alignement sur les limites de mots (WORD aligned).

A.3.3.1 Format du type de données DATE_AND_TIME (date et heure)

Lorsque vous saisissez la date et l'heure avec le type de données DATE_AND_TIME (DT), vos entrées sont enregistrées dans 8 octets en format DCB. Le type de données DATE_AND_TIME englobe la plage suivante :

DT#1990-1-1-0:0:0.0 à DT#2089-12-31-23:59:59.999

Les exemples ci-après donnent les syntaxes possibles pour saisir la date et l'heure du jeudi, 25 décembre 1993, 8 heures 01 et 1,23 seconde. Les deux formats suivants sont possibles :

- DATE_AND_TIME#1993-12-25-8:01:1.23
- DT#1993-12-25-8:01:1.23

Vous disposez des fonctions standard CEI (International Electrotechnical Commission) suivantes pour traiter le type de données DATE_AND_TIME :

- Conversion de la date et de l'heure au format DATE_AND_TIME

FC3 : D_TOD_DT

- Extraction de la date du format DATE_AND_TIME

FC6 : DT_DATE

- Extraction du jour de la semaine du format DATE_AND_TIME

FC7 : DT_DAY

- Extraction de l'heure du format DATE_AND_TIME

FC8 : DT_TOD

Le tableau suivant présente le contenu des octets contenant les informations de date et l'heure. L'exemple montre la date et l'heure pour jeudi 25 décembre 1993, 8 heures 01 et 1,23 seconde.

Octet	Contenu	Exemple
0	Année	B#16#93
1	Mois	B#16#12
2	Jour	B#16#25
3	Heures	B#16#08
4	Minutes	B#16#01
5	Secondes	B#16#01
6	Les deux chiffres de poids fort de MSEC	B#16#23
7 (4BPFo)	Les chiffres de poids faible de MSEC	B#16#0
7 (4BPFa)	Jour de la semaine 1 = dimanche 2 = lundi ... 7 = samedi	B#16#5

La plage autorisée pour le type de données "DATE_AND_TIME" est :

- mini : DT#1990-1-1-0:0:0.0
- maxi : DT#2089-12-31-23:59:59.999

	Plage de valeurs	Code DCB
Année	1990 – 1999 2000 – 2089	90h – 99h 00h – 89h
Mois	1 – 12	01h – 12h
Jour	1 – 31	01h – 31h
Heures	00 – 23	00h – 23h
Minutes	00 – 59	00h – 59h
Secondes	00 – 59	00h – 59h
Millisecondes	0 – 999	000h – 999h
Jour de la semaine	dimanche – samedi	1h – 7h

A.3.3.2 Format du type de données STRING

Une chaîne de caractères (STRING) englobe un ensemble de 254 caractères au maximum (type de données CHAR). La zone réservée par défaut à ce type de données est de 256 octets (254 octets pour les caractères et 2 octets pour l'en-tête du type de données). Vous pouvez réduire l'espace requis en mémoire par une variable de ce type en précisant, à la suite du mot-clé STRING, le nombre de caractères à enregistrer dans la chaîne.

Type de données	Longueur (octets)	Format
STRING[n] Ou STRING	n+2	Chaîne de caractères ASCII de longueur quelconque. n indique la longueur de la chaîne. La longueur maximale est de 254 caractères. En l'absence d'indication de longueur, la longueur par défaut est de 254 caractères.

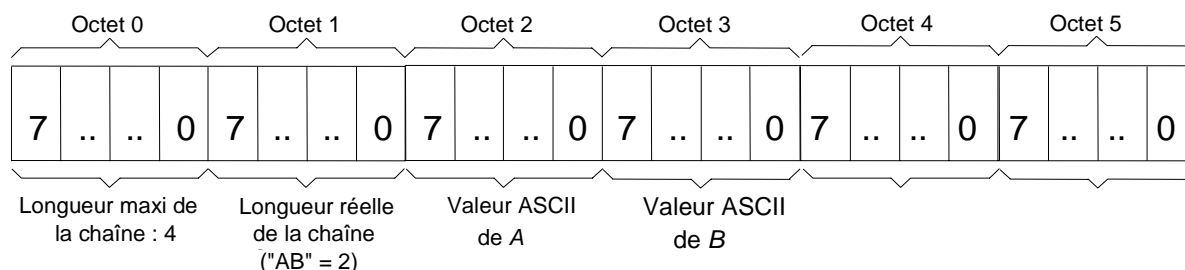
Type de données	Exemples pour le format
STRING[2]	'ab'
STRING[55]	'La chaîne de caractères doit comporter au maximum 55 caractères.'

Nota

Vous devez écrire la chaîne de caractères entre apostrophes.

La figure suivante montre la suite des octets pour la saisie du type de données STRING[4] avec la valeur initiale 'AB'.

Exemple



L'utilisateur doit initialiser les données locales dynamiques de type STRING avant leur première utilisation, par exemple par une séquence LIST de la forme :

```
LAR1 P#lokal_string_var // lokal_string_var est déclaré dans VAR_TEMP en tant
                          //que STRING[200]
L    200                // Longueur STRING précisée ci-dessus
T    LB [AR1, P#0.0]    // Inscrire dans l'octet de longueur maxi de la chaîne
L    5                  // Longueur réelle de la chaîne
T    LB [AR1, P#1.0]    // Inscrire la longueur réelle de la chaîne
```

Nota

Si le contenu d'une chaîne de caractères est modifié par le programme utilisateur, il faut écrire aussi dans l'octet "Longueur réelle" ou l'actualiser afin que la PG puisse afficher la chaîne de caractères.

Si vous avez défini une variable temporaire de type STRING, il faut écrire la longueur définie dans l'octet "Longueur maxi" avant d'utiliser la variable dans le programme utilisateur.

A.3.3.3 Format du type de données ARRAY

Un tableau (= ARRAY) est un type de données complexe avec 6 dimensions au plus. Tous les éléments d'un tableau peuvent être n'importe quels types de données (à l'exception des types de paramètre), mais tous doivent être du même type. Les tableaux ne peuvent pas être imbriqués et doivent comporter au moins deux éléments.

Exemple : "ARRAY [1..2,1..3] OF INT" définit un tableau en format 2 x 3 composé d'entiers.

Vous accédez aux données au moyen de l'indice ("[m,n]"), sachant qu'il faut respecter $1 \leq m \leq 2$ et $1 \leq n \leq 3$.

L'indice peut être n'importe quel nombre entier. Dans la déclaration, les limites du tableau doivent être telles qu'il comprenne en tout 65535 éléments au plus. Les valeurs limites d'une dimension (par ex. x1 et x2) peuvent être négatives, nulles ou positives, x2 doit cependant être supérieure à x1.

Comment indiquer les dimensions

Exemples : une dimension : ARRAY[x1..x2]

ARRAY[-2..-1]

ARRAY[1..0]

ARRAY[1..2]

Les dimensions supplémentaires sont séparées par des virgules.

Exemple : trois dimensions : ARRAY[x1..x2, y1..y2, z1..z2]

Edition de la liste des variables

Les ARRAY peuvent être utilisés dans le programme. Vous pouvez indiquer des valeurs initiales dans la colonne d'initialisation.

Valeurs initiales

Vous initialisez les éléments au moyen d'une liste de valeurs séparées par des virgules. Vous pouvez utiliser un facteur de répétition, par exemple "4(10)" ("affecte la valeur 10 aux quatre éléments suivants"), pour affecter des valeurs initiales dans un tableau. Les éléments dépourvus de valeur initiale prennent la valeur par défaut zéro.

Adressage symbolique

Les instructions de votre programme peuvent accéder aux valeurs du tableau grâce au nom de la variable.

Exemple

L	#tableau[3]	Charger dans l'ACCU1 la valeur contenue dans l'élément 3 de l'ARRAY nommé "tableau".
---	-------------	--

A.3.3.4 Format du type de données STRUCT

Une structure (STRUCT) est un type de données complexe pouvant présenter jusqu'à 8 niveaux d'imbrication. Les éléments d'une structure peuvent être n'importe quels types de données autorisés. Le type de données STRUCT doit comporter au minimum deux composants entre STRUCT et END_STRUCT. Vous pouvez déclarer une structure dans la table de déclaration des variables d'un bloc de code ou dans un type de données utilisateur (UDT).

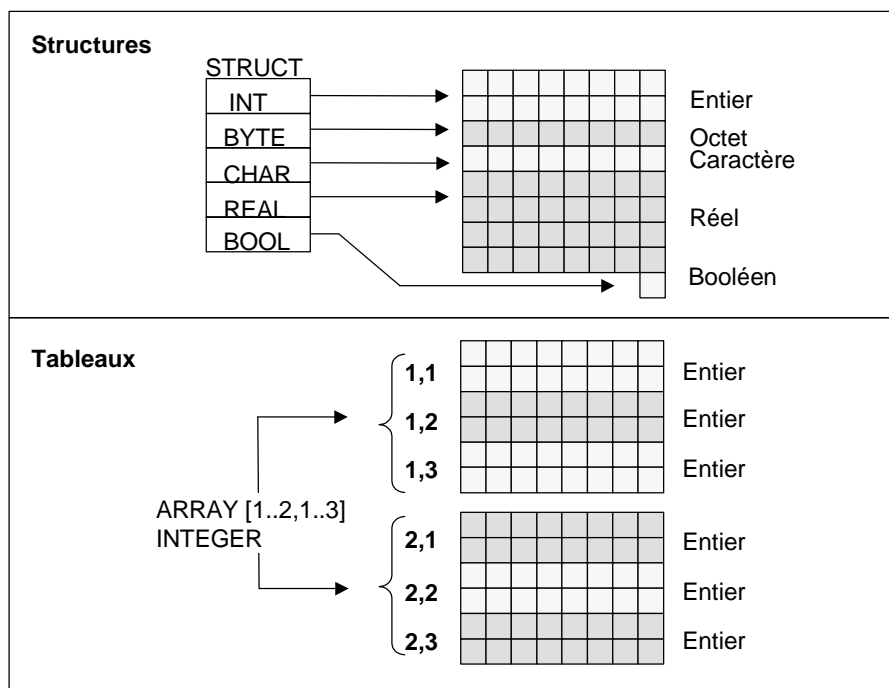
Dans le programme utilisateur, vous pouvez accéder aux différentes variables d'une structure au moyen de <nom_structure.nom_variable>. Une structure contenue dans une autre structure ne compte que comme un composant.

A.3.3.5 Utilisation de types de données complexes

Vous pouvez créer de nouveaux types de données en combinant des types de données simples et complexes pour obtenir les types de données complexes suivants :

- tableau (ARRAY) : agrégat de données de même type ;
- structure (STRUCT) : agrégat de données de types différents ;
- chaîne (STRING) : tableau à une dimension de 254 caractères (type de données CHAR) au maximum ; une chaîne ne peut être transmise que comme entité complète et sa longueur doit être la même pour les paramètres formel et effectif du bloc ;
- date et heure (DATE_AND_TIME) : année, mois, jour, heures, minutes, secondes, millisecondes et jour de la semaine.

La figure ci-après montre comment les tableaux et structures organisent des types de données en une zone de stockage d'informations. Vous pouvez définir un tableau ou une structure soit dans un DB, soit dans la table de déclaration des variables d'un FB, d'une FC ou d'un OB.



A.3.3.6 Utilisation de tableaux pour l'accès aux données

Tableaux

Un tableau correspond à un agrégat de données de même type (simple ou complexe). Il n'est pas possible de définir un tableau de tableaux. Lorsque vous définissez un tableau :

- vous lui donnez un nom ;
- vous déclarez son type à l'aide du mot-clé ARRAY ;
- vous indiquez sa taille à l'aide d'un indice ; vous saisissez le premier et le dernier nombre pour chaque dimension (jusqu'à 6) dans le tableau ; vous indiquez l'indice entre crochets, chaque dimension étant séparée par une virgule, le premier et dernier nombre de chaque dimension par deux points ; voici, par exemple, comment définir un tableau tridimensionnel :

[1..5,-2..3,30..32]

- vous indiquez le type des données à enregistrer dans le tableau.

Exemple 1

La figure ci-après montre un tableau de trois nombres entiers. Vous accédez aux données rangées dans le tableau à l'aide de l'indice, c'est-à-dire du nombre entre crochets. L'indice du deuxième nombre entier est, par exemple, Temp_fonct[2].

Un indice peut être n'importe quelle valeur entière, même négative (-32768 à 32767). Il aurait également été possible de définir le tableau de la figure ci-après comme ARRAY [-1..1]. L'indice du premier entier serait alors Temp_fonct[-1], celui du deuxième entier Temp_fonct[0] et celui du troisième Temp_fonct[1].

Adresse	Nom	Type	Val. init.	Commentaire
0.0		STRUCT		
+0.0	Temp oper	ARRAY [1..3]		
*2.0		INT		
=3.0		END STRUCT		

Temp_fonct =
 ARRAY [1..3] INTEGER

{

1

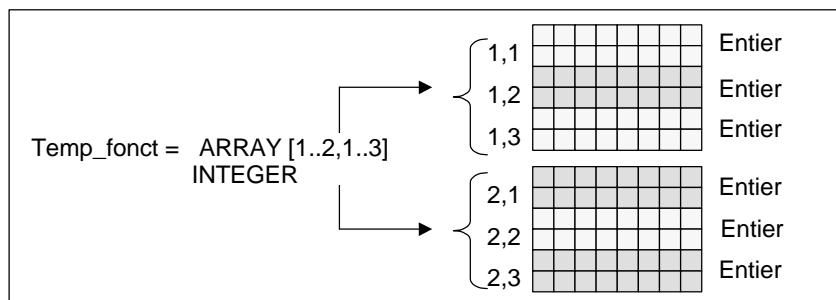
2

3

Temp_fonct [1]
 Temp_fonct [2]
 Temp_fonct [3]

Exemple 2

Un tableau peut également décrire un agrégat multidimensionnel de types de données. La figure ci-après montre un tableau bidimensionnel de nombres entiers.



Vous accédez aux données de ce tableau à l'aide de l'indice. Pour l'exemple, le premier nombre entier est Temp_fonct[1,1], le troisième Temp_fonct[1,3], le quatrième Temp_fonct[2,1] et le sixième Temp_fonct[2,3].

Un tableau peut avoir jusqu'à six dimensions (six indices). Par exemple, la variable Temp_fonct pourrait être définie comme tableau à six dimensions de la manière suivante :

ARRAY [1..3, 1..2, 1..3, 1..4, 1..3, 1..4]

L'indice du premier élément de ce tableau est Temp_fonct[1,1,1,1,1,1] et celui du dernier élément Temp_fonct[3,2,3,4,3,4].

Création d'un tableau

La définition d'un tableau se fait lors de la déclaration de données dans un DB ou dans la table de déclaration des variables. Pour déclarer un tableau, vous entrez le mot-clé ARRAY suivi de sa taille entre crochets :

[limite inférieure..limite supérieure]

Pour un tableau multidimensionnel, vous indiquez les autres limites inférieures et supérieures en séparant les dimensions par une virgule. Dans la figure ci-après, on déclare un tableau 2 x 3.

Adresse	Nom	Type	Val. init.	Commentaire
0.0		STRUCT		
+0.0	Chaleur 2x3	ARRAY[1..2, 1..3]		
*2.0		INT		
=6.0		END STRUCT		

Saisie de valeurs initiales pour un tableau

Vous pouvez affecter une valeur initiale à chaque élément des tableaux que vous créez. Il existe deux méthodes pour saisir les valeurs initiales :

- Saisie de valeurs individuelles : vous indiquez pour chaque élément du tableau une valeur autorisée (pour le type de données du tableau) ; indiquez les valeurs dans l'ordre des éléments, par exemple [1,1], en séparant les éléments par une virgule.
- Indication d'un facteur de répétition : pour des éléments qui se suivent et doivent prendre la même valeur initiale, vous pouvez préciser le nombre d'éléments (facteur de répétition x) et leur valeur initiale ; le format de saisie d'un facteur de répétition est $x(y)$, x étant le facteur de répétition et y la valeur à répéter.

Pour l'exemple de la figure ci-dessus, vous pouvez indiquer la valeur initiale des six éléments en entrant : 17, 23, -45, 556, 3342, 0. Mais pour leur donner à tous la valeur initiale 10, il vous suffirait d'indiquer 6(10). Vous pourriez également donner une valeur individuelle aux deux premiers éléments et la valeur nulle aux quatre autres en précisant : 17,23,4(0).

Accès aux données d'un tableau

Vous accédez aux données d'un tableau par l'indice de l'élément concerné dans le tableau. L'indice est combiné au mnémonique du tableau.

Exemple : si le tableau déclaré dans la figure ci-dessus commence au premier octet du DB20 (Moteur), vous accédez au deuxième élément du tableau à l'aide de l'adresse suivante :

Moteur.Chaleur_2x3[1,2]

Utilisation de tableaux comme paramètres

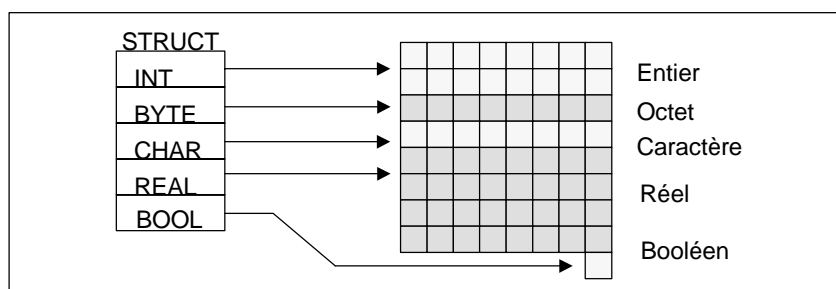
Vous pouvez transmettre des tableaux comme paramètres. Lorsque vous déclarez un paramètre ARRAY dans la déclaration des variables, vous devez transmettre le tableau complet, et non pas des éléments individuels. Il est toutefois possible d'affecter un élément de tableau à un paramètre lorsque vous appelez un bloc si cet élément correspond au type de données du paramètre.

Les tableaux que vous transmettez comme paramètres ne doivent pas nécessairement avoir le même nom (ou même avoir un nom), mais il faut qu'ils soient tous deux - paramètre effectif et paramètre formel - organisés de manière identique. Ainsi, un tableau 2x3 de nombres entiers ne peut-il être transmis comme paramètre que si le paramètre formel du bloc définit un tableau 2x3 de nombres entiers et si le paramètre effectif fourni dans l'opération d'appel est également un tableau 2x3 de nombres entiers.

A.3.3.7 Utilisation de structures pour l'accès aux données

Structures

Une structure correspond à un agrégat de données de types différents (toute combinaison de types de données simples ou complexes, y compris tableaux et structures). Cela permet de regrouper des données selon la logique de votre processus. Cela permet également de transmettre des paramètres comme une entité de données, plutôt que sous la forme d'éléments distincts. La figure ci-après montre une structure constituée d'un nombre entier, d'un octet, d'un caractère, d'un nombre à virgule flottante et d'une valeur booléenne.



Une structure peut être imbriquée jusqu'à huit niveaux (par exemple, une structure de structures contenant des tableaux).

Création d'une structure

La définition d'une structure se fait lors de la déclaration de données dans un DB ou dans la déclaration des variables d'un bloc de code.

Dans la figure ci-après, on déclare une structure *lot_1* constituée des éléments suivants : un nombre entier (pour la quantité), un octet (pour les données brutes), un caractère (pour le code de commande), un nombre à virgule flottante (pour la température) et un memento booléen (pour signaler l'achèvement).

Adresse	Nom	Type	Val. init.	Commentaire
0.0	lot 1	STRUCT		
+0.0	quantite	INT	100	
+2.0	donnees brutes	BYTE		
+4.0	code commande	CHAR		
+6.0	temperature	REAL	120	
+8.1	fin	BOOL	FALSE	
=10.0		END STRUCT		

Affectation de valeurs initiales à une structure

Pour affecter une valeur initiale à chaque élément d'une structure, vous indiquez une valeur autorisée pour le type de données et le nom de chaque élément. Vous pourriez affecter les valeurs initiales suivantes à l'exemple de la figure ci-dessus :

Quantité	=	100
Données brutes	=	B#(0)
Code de commande	=	'Z'
Température	=	120
Achèvement	=	False

Enregistrement des données et accès aux données dans une structure

Vous accédez aux éléments individuels d'une structure. Vous pouvez utiliser l'adresse symbolique - *lot_1.temperature*, par exemple - ou l'adresse absolue sous laquelle est rangé l'élément. Si, par exemple, *lot_1* est enregistré dans le DB20 à partir de l'octet 0, l'adresse absolue de *quantité* est *DB20.DBW0* et celle de *temperature* est *DB20.DBD6*.

Utilisation de structures comme paramètres

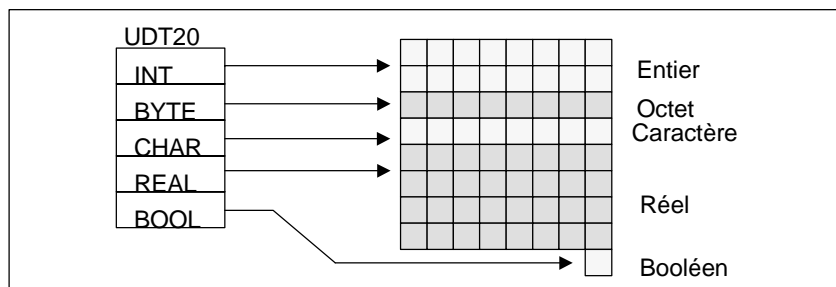
Vous pouvez transmettre des structures comme paramètres. Lorsque vous déclarez un paramètre STRUCT dans la déclaration des variables, vous devez transmettre une structure d'organisation identique. Il est possible d'affecter un élément de structure à un paramètre lorsque vous appelez un bloc si cet élément correspond au type de données du paramètre.

Lorsque vous transmettez des structures comme paramètres, elles doivent être toutes deux - paramètre effectif et paramètre formel - organisées de manière identique : elles doivent avoir les mêmes types de données dans le même ordre.

A.3.3.8 Utilisation de types de données utilisateur pour l'accès aux données

Types de données utilisateur

Les types de données utilisateur (user data type, UDT) peuvent combiner des types de données simples et complexes. Vous pouvez attribuer un nom aux UDT et les utiliser plusieurs fois. La figure ci-après montre la structure d'un type de données utilisateur constitué d'un nombre entier, d'un octet, d'un caractère, d'un nombre à virgule flottante et d'une valeur booléenne.



Au lieu d'entrer tous les types de données individuellement ou sous forme de structure, il vous suffit d'indiquer "UDT20" comme type de données et STEP 7 Lite allouera automatiquement l'espace nécessaire en mémoire.

Création d'un type de données utilisateur

Vous définissez les UDT dans STEP 7 Lite. La figure suivante montre un UDT composé des éléments suivants : un nombre entier (pour la quantité), un octet (pour les données brutes), un caractère (pour le code de commande), un nombre à virgule flottante (pour la température) et un memento booléen (pour signaler l'achèvement). Vous pouvez affecter un mnémonique à cet UDT dans la table des mnémoniques (donnees_process, par exemple).

Adresse	Nom	Type	Val. init.	Commentaire
0.0	lot 1	STRUCT		
+0.0	quantite	INT	100	
+2.0	donnees brutes	BYTE		
+4.0	code commande	CHAR		
+6.0	temperature	REAL	120	
+8.1	fin	BOOL	FALSE	
=10.0		END STRUCT		

Après avoir créé un UDT, vous pouvez l'utiliser comme un type de données, par exemple comme si pour une variable vous déclariez le type de données *UDT200* dans un DB (ou dans la table de déclaration des variables d'un FB).

La figure suivante montre un DB avec la variable *donnees_processus_1* de type de données UDT200. Vous indiquez uniquement *UDT200* et *donnees_processus_1*. Les autres champs sont créés à la compilation du DB.

Adresse	Nom	Type	Val. init.	Commentaire
0.0		STRUCT		
+6.0	donnees_proc 1	UDT200		
=6.0		END STRUCT		

Affectation de valeurs initiales à un type de données utilisateur

Pour affecter une valeur initiale à chaque élément d'un UDT, vous indiquez une valeur autorisée pour le type de données et le nom de chaque élément. Vous pourriez affecter les valeurs initiales suivantes à l'exemple de la figure ci-dessus :

Quantité	=	100
Données brutes	=	B#16#0
Code de commande	=	'Z'
Température	=	1.200000e+002
Achèvement	=	False

Lorsque vous déclarez une variable comme étant d'un type de données utilisateur, les valeurs initiales pour cette variable seront les valeurs entrées à la création de l'UDT.

Enregistrement des données et accès aux données dans un type de données utilisateur

Vous accédez aux éléments individuels d'un UDT. Vous pouvez utiliser l'adresse symbolique - *lot_1.temperature*, par exemple - ou l'adresse absolue sous laquelle est rangé l'élément. Si, par exemple, *lot_1* est enregistré dans le DB20 à partir de l'octet 0, l'adresse absolue de *quantité* est *DB20.DBW0* et celle de *temperature* est *DB20.DBD6*.

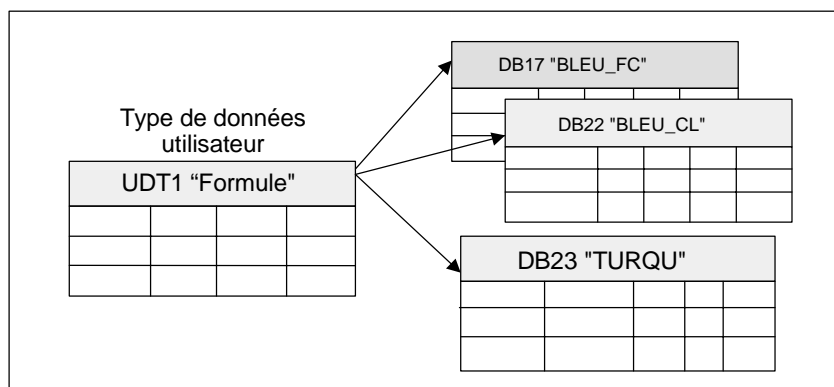
Utilisation de types de données utilisateur comme paramètres

Vous pouvez transmettre des variables de type de données UDT comme paramètres. Lorsque vous déclarez un paramètre comme UDT dans la déclaration des variables, vous devez transmettre un UDT dont les éléments de données ont une organisation identique. Il est également possible d'affecter un élément d'UDT à un paramètre lorsque vous appelez un bloc si cet élément correspond au type de données du paramètre.

Avantages des DB associés à un UDT

Vous pouvez, à l'aide des UDT que vous avez créés, générer de nombreux blocs de données ayant la même organisation de données. Vous pouvez adapter ces blocs de données à chaque tâche en saisissant des valeurs effectives différentes.

Si, par exemple, vous organisez un UDT pour une formule (par ex. pour mélanger des couleurs), vous pouvez associer à cet UDT plusieurs DB contenant à chaque fois d'autres indications de quantités.



L'organisation de l'UDT conditionne celle du bloc de données associé.

A.3.4 Types de paramètre

En plus des types de données simples et complexes, vous pouvez définir des types de paramètre pour des paramètres formels devant être transmis entre blocs. STEP 7 Lite dispose des types de paramètre ci-après.

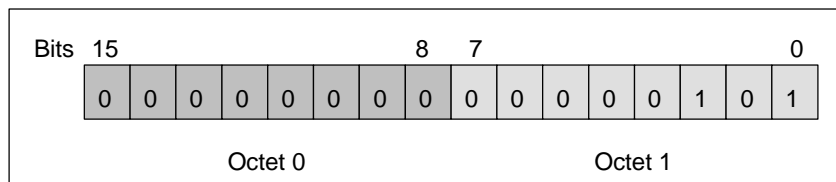
- **TIMER** ou **COUNTER** : identifient une temporisation ou un compteur précis devant être utilisé lors du traitement. Le paramètre effectif que vous fournissez à un paramètre formel de type **TIMER** ou **COUNTER** doit être une temporisation ou un compteur : vous indiquez un **T** ou un **Z** suivi d'un nombre entier positif.
- **BLOCK** : identifie un bloc précis devant être utilisé comme entrée ou comme sortie. La déclaration du paramètre détermine le type de bloc (**FB**, **FC**, **DB**, etc.) à utiliser. Si vous indiquez un paramètre effectif pour un paramètre formel de type **BLOCK**, ce doit être une adresse de bloc. Exemple : "FC101" en adressage absolu ou "Soupape" en adressage symbolique.
- **POINTER** : référence l'adresse d'une variable. Un pointeur contient une adresse au lieu d'une valeur. Lorsque vous indiquez un paramètre effectif pour un paramètre formel de type **POINTER**, ce doit être l'adresse. Dans STEP 7 Lite, vous pouvez préciser un pointeur en format de pointeur ou simplement comme adresse (par ex. M50.0). Exemple de format de pointeur pour l'adressage de données commençant à M 50.0 : P#M50.0
- **ANY** : s'utilise lorsque le type de données du paramètre effectif est inconnu ou lorsqu'on peut faire appel à un type de données quelconque. Vous trouverez de plus amples informations sur le paramètre **ANY** dans les rubriques Format du type de paramètre **ANY** ou Utilisation du type de paramètre **ANY**.

Un type de paramètre peut également être un type de données utilisateur (UDT). Vous trouverez de plus amples informations sur les UDT dans la rubrique Utilisation de types de données utilisateur pour l'accès aux données.

Paramètre	Taille	Description
TIMER	2 octets	Identifie une temporisation précise que le programme dans le bloc de code appelé doit utiliser. Format : T1
COUNTER	2 octets	Identifie un compteur précis que le programme dans le bloc de code appelé doit utiliser. Format : Z10
BLOCK_FB BLOCK_FC BLOCK_DB BLOCK_SDB	2 octets	Identifie un bloc précis que le programme dans le bloc de code appelé doit utiliser. Format : FC101 DB42
POINTER	6 octets	Identifie l'adresse. Format : P#M50.0
ANY	10 octets	Utilisé lorsque le type de données du paramètre effectif est inconnu. Format: P#M50.0 BYTE 10 format ANY pour P#M100.0 WORD 5 types de données L#1COUNTER 10 format ANY pour types de paramètre

A.3.4.1 Format des types de paramètre BLOCK, COUNTER et TIMER

STEP 7 Lite enregistre les types de paramètre BLOCK, COUNTER et TIMER sous forme de nombres binaires dans un mot (32 bits). La figure suivante montre le format de ces types de paramètre.



Le nombre autorisé de blocs, temporisations et compteurs dépend de votre modèle de CPU S7. Pour plus de renseignements à ce sujet ainsi que sur le nombre maximal de blocs disponibles, consultez les fiches techniques de votre CPU dans le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*.

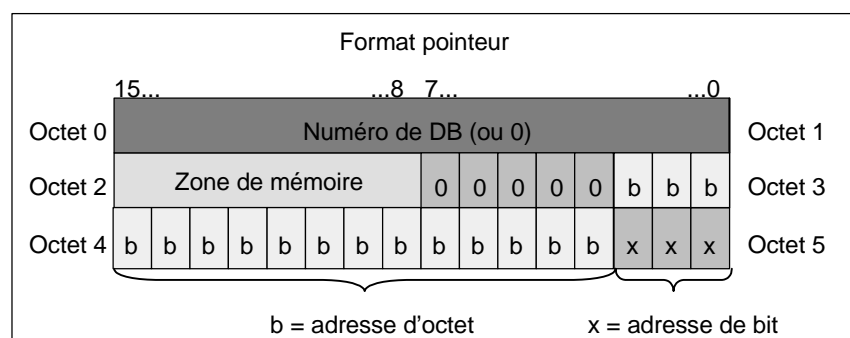
Ce type vous permet de transmettre au bloc de code le numéro d'une temporisation, d'un compteur, d'un bloc fonctionnel, d'un bloc de données, d'un bloc de données système ou d'une fonction qu'il doit utiliser pour le traitement. Pour affecter une valeur à ce paramètre formel déclaré, vous indiquez comme paramètre effectif un "T", "Z", "FB", "DB", "SDB" ou "FC" suivi d'un nombre entier positif.

Exemple

```
Call FB 10, DB110(
:
Entree_Var_Counter:= Z5,      // FB10 utilisera le
                                // compteur 5 pour le traitement
:
);
```

A.3.4.2 Format du type de paramètre POINTER

STEP 7 Lite enregistre le type de paramètre POINTER dans 6 octets (48 bits). La figure suivante montre la nature des données enregistrées dans chaque octet.



Le type de paramètre POINTER enregistre les informations suivantes :

- Numéro de DB (ou 0, lorsque les données ne sont pas enregistrées dans un DB)
- Zone de mémoire dans la CPU (le tableau suivant indique les codes hexadécimaux des zones de mémoire pour le type de paramètre POINTER)

Code hexadécimal	Zone de mémoire	Description
b#16#81	E	Zone de mémoire des entrées
b#16#82	A	Zone de mémoire des sorties
b#16#83	M	Zone de mémoire des mémentos
b#16#84	DB	Bloc de données
b#16#85	DI	Bloc de données d'instance
b#16#86	L	Données locales (pile L)
b#16#87	V	Données locales précédentes

- Adresse des données (en format octet.bit)

STEP 7 Lite propose le format pointeur : p#zone de mémoire octet.bit_adresse. (Si le paramètre formel a été déclaré comme type de paramètre POINTER, il vous suffit d'indiquer la zone de mémoire et l'adresse ; STEP 7 Lite convertit automatiquement votre entrée en format pointeur.) Les exemples suivants montrent comment vous saisissez le type de paramètre POINTER pour les données commençant à M50.0 :

- P#M50.0
- M50.0 (si le paramètre formel a été déclaré comme POINTER)

Exemple

```
call FB 10, DB110(
:
Entree_Var_Addr:= P#M20.0,    // FB10 utilisera l'adresse du
                             // memento 20.0 pour le traitement
:
);
```


A.3.4.3 Utilisation du type de paramètre POINTER

Un pointeur est utilisé pour adresser un opérande. L'avantage de ce type d'adressage est que vous pouvez modifier de manière dynamique l'opérande de l'instruction durant l'exécution du programme.

Pointeur pour l'adressage indirect en mémoire

Les instructions utilisant l'adressage indirect en mémoire sont composées d'une opération, d'un identificateur d'opérande et d'un décalage (qui doit être indiqué entre crochets).

Exemple de pointeur en format double mot :

L	P#8.7	Charger la valeur du pointeur dans l'ACCU 1
T	MD2	Transférer le pointeur dans MD2
U	E [MD2]	Interroger l'état de signal à l'entrée E 8.7
=	A [MD2]	et affecter l'état de signal à la sortie A 8.7

Pointeur pour l'adressage intrazone et interzone

Les instructions utilisant ce type d'adressage sont composées d'une opération et des éléments suivants : identificateur d'opérande, identificateur de registre d'adresse, décalage.

Le registre d'adresse (AR1/2) et le décalage doivent être indiqués ensemble entre crochets.

Exemple d'adressage intrazone

Le pointeur ne contient aucune indication de zone de mémoire :

L	P#8.7	Charger la valeur du pointeur dans l'ACCU 1
LAR1		Charger le pointeur de l'ACCU 1 dans AR1
U	E [AR1, P#0.0]	Interroger l'état de signal de l'entrée E 8.7 et
=	A [AR1, P#1.1]	affecter l'état de signal à la sortie A 10.0

Le décalage 0.0 n'a pas d'effet. La sortie 10.0 se calcule à partir de 8.7 (AR1) plus le décalage 1.1. Le résultat est 10.0 et non pas 9.8, voir le format du pointeur.

Exemple d'adressage interzone

Dans l'adressage interzone, la zone de mémoire est précisée dans le pointeur (dans l'exemple, E ou A).

L	P# E8.7	Charger la valeur du pointeur et l'identificateur de zone dans l'ACCU 1
LAR1		Charger la zone de mémoire E et l'adresse 8.7 dans AR1
L	P# A8.7	Charger la valeur du pointeur et l'identificateur de zone dans l'ACCU 1
LAR2		Charger la zone de mémoire A et l'adresse 8.7 dans AR2
U	[AR1, P#0.0]	Interroger l'état de signal de l'entrée E 8.7 et
=	[AR2, P#1.1]	affecter l'état de signal à la sortie A 10.0.

Le décalage 0.0 n'a pas d'effet. La sortie 10.0 se calcule à partir de 8.7 (AR2) plus 1.1 (décalage). Le résultat est 10.0 et non pas 9.8, voir format du pointeur.

A.3.4.4 Bloc pour modifier le pointeur

A l'aide du bloc-exemple FC3 "Déplacement de pointeur", il est possible de modifier l'adresse de bit ou l'adresse d'octet d'un pointeur. Le pointeur à modifier est transmis à la variable "Pointeur" dans l'appel de la FC (vous pouvez utiliser des pointeurs interzone et intrazone en format double mot).

Le paramètre "bit-octet" vous permet de modifier l'adresse de bit ou l'adresse d'octet du pointeur (0 : adresse de bit, 1 : adresse d'octet). La variable "valeur_inc" (en format entier) indique la valeur qui doit être additionnée ou soustraite au contenu de l'adresse. Vous pouvez également indiquer des nombres négatifs pour décrémenter l'adresse.

Quand c'est l'adresse de bit qui est modifiée, un transfert dans l'adresse d'octet est effectué (également pour la décrémentation), par exemple :

- P#M 5.3, bit_octet = 0, valeur_inc = 6 => P#M 6.1 ou
- P#M 5.3, bit_octet = 0, valeur_inc = -6 => P#M 4.5.

La fonction n'a pas d'effet sur l'information de zone du pointeur.

La FC corrige le débordement haut/bas du pointeur. Dans ce cas, le pointeur n'est pas modifié et la variable côté sortie "RET_VAL" (traitement d'erreur possible) est mise à "1" (jusqu'au prochain traitement correct de la FC3). Ceci est le cas lorsque :

- l'adresse de bit est sélectionnée et valeur_inc >7 ou <-7,
- l'adresse de bit ou l'adresse d'octet est sélectionnée et la modification aurait pour conséquence une adresse d'octet "négative",
- l'adresse de bit ou l'adresse d'octet est sélectionnée et la modification aurait pour conséquence une adresse d'octet trop élevée.

Exemple de bloc pour modifier le pointeur dans LIST

```

FUNCTION FC 3: BOOL
  TITLE =Déplacement de pointeur
  //La FC3 peut être utilisée pour modifier des pointeurs.
  AUTHOR : AUT1CS1
  FAMILY : INDADR
  NAME : ADRPOINT
  VERSION : 0.0

  VAR_INPUT
    Bit_octet : BOOL ;           //0 : adresse de bit, 1 : adresse d'octet
    Valeur_inc : INT ;           //Incrément (si valeur négative =>
                                //décrémentation/si valeur positive
                                //=> incrémentation)

  END_VAR

  VAR_IN_OUT
    Pointeur : DWORD ;          //Pointeur à utiliser
  END_VAR

  VAR_TEMP
    Valeur_inc1 : INT ;          //Incrément de valeur intermédiaire
    Pointeur1 : DWORD ;          //Pointeur de valeur intermédiaire
    Val_int : DWORD ;            //Variable auxiliaire
  END_VAR

  BEGIN
  NETWORK
  TITLE =
  //Le bloc corrige automatiquement les changements qui modifient les
  //informations de zone du pointeur ou qui conduisent à des
  //pointeurs "négatifs" !
    SET      ;                   //Mettre le RLG à 1 et
    R        #RET_VAL;           //remettre le débordement à zéro
    L        #Pointeur;          //Affecter le pointeur temporaire
    T        #Pointeur1;         //de valeur intermédiaire
    L        #Valeur_inc;        //Affecter l'incrément temporaire
    T        #Valeur_inc1;       //de valeur intermédiaire
    U        #Bit_octet;         //lorsque =1, alors opération sur
l'adresse d'octet
    SPB Octet;                   //Saut au calcul de l'adresse d'octet
    L        7;                  //Si valeur incrément > 7,
    L        #Valeur_inc1;
    <I        ;
    S        #RET_VAL;           //alors mettre RET_VAL à 1 et
    SPB      Fin;                //sauter à la fin
    L        -7;                 //Si valeur incrément < -7,
    <I        ;
    S        #RET_VAL;           //alors mettre RET_VAL à 1 et
    SPB      Fin;                //sauter à la fin
    U        L 1.3;              //si bit 4 de la valeur = 1
                                //(valeur_inc négative)
    SPB      neg;                //alors sauter à la soustraction
                                //des adresses de bit

```

```

L      #Pointeur1;      //Charger informations d'adresse
                        //du pointeur
L      #Valeur_inc1;    //et additionner l'incrément
+D      ;
SPA     test;           //Sauter au test de résultat négatif
neg:    L #Pointeur1;    //Charger informations d'adresse
                        //du pointeur
L      #Valeur_inc1;    //Charger l'incrément
NEGI    ;              //Effectuer la négation
                        //de la valeur négative,
-D      ;              //soustraire la valeur
SPA     test;           //et sauter au test
Octet:  L 0;            //Début de la modification
                        //de l'adresse d'octet
L      #Valeur_inc1;    //Si incrément >=0, alors
<I      ;
SPB     pos;            //sauter à l'addition, sinon
L      #Pointeur1;      //charger informations d'adresse
                        //du pointeur,
L      #Valeur_inc1;    //charger l'incrément,
NEGI    ;              //effectuer la négation
                        //de la valeur négative,
SLD     3;              //décaler l'incrément de 3 positions
                        //vers la gauche,
-D      ;              //soustraire la valeur
SPA     test;           //et sauter au test
pos:    SLD 3;           //décaler l'incrément de 3 positions
                        //vers la gauche
L      #Pointeur1;      //charger informations d'adresse
                        //du pointeur
+D      ;              //additionner l'incrément
test:   T #Valeur_int;   //Transférer les calculs de résultat
                        //dans Valeur_int,
U      L 7.3;           //Si adresse d'octet invalide
                        //trop grande ou
S      #RET_VAL;        //négative), alors mettre RET_VAL à 1
SPB     Fin;            //et sauter à la fin,
L      #Valeur_int;     //sinon transférer le résultat
T      #Pointeur;       //dans le pointeur
Fin:    NOP 0;
END_FUNCTION

```

A.3.4.5 Format du type de paramètre ANY

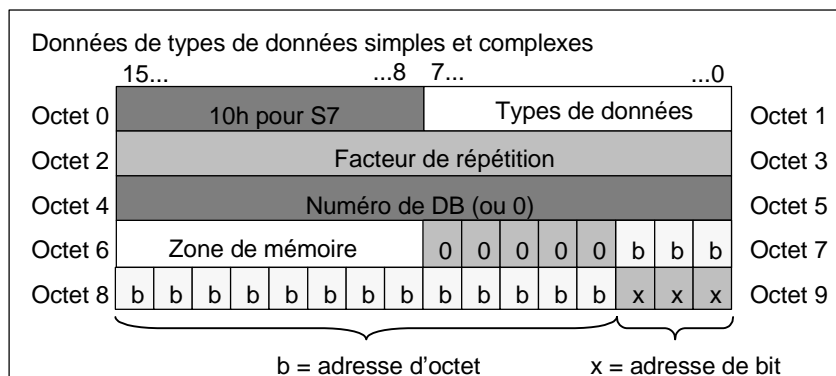
STEP 7 Lite enregistre les données du type de paramètre ANY dans 10 octets. En définissant un paramètre de type ANY, vous devez veiller à ce que les 10 octets soient tous occupés, car le bloc appelé évalue le contenu entier du paramètre. Si vous spécifiez un numéro de DB dans l'octet 4, par exemple, vous devez aussi indiquer explicitement la zone de mémoire dans l'octet 6.

STEP 7 Lite gère les données de types simples et complexes autrement que celles des types de paramètre.

Format ANY pour les types de données

Pour les types de données simples et complexes, STEP 7 Lite enregistre les données suivantes :

- types de données,
- facteur de répétition,
- numéro de DB,
- zone de mémoire dans laquelle les informations sont enregistrées,
- adresse de début des données.



Le facteur de répétition désigne une quantité du type de données identifié qui est à transmettre par le type de paramètre ANY. Vous pouvez ainsi indiquer une zone de données et également utiliser des tableaux et structures en liaison avec le type de paramètre ANY. STEP 7 Lite caractérise les tableaux et structures comme nombre de types de données (à l'aide du facteur de répétition). Pour transmettre 10 mots, par exemple, vous devez entrer la valeur 10 pour le facteur de répétition et la valeur 04 pour le type de données.

L'adresse est enregistrée dans le format octet.bit, l'adresse d'octet étant enregistrée dans les bits 0 à 2 de l'octet 7, dans les bits 0 à 7 de l'octet 8 et dans les bits 3 à 7 de l'octet 9.

L'adresse de bit est enregistrée dans les bits 0 à 2 de l'octet 9.

Dans le cas du pointeur zéro de type de données NIL, tous les octets ont la valeur 0 à partir de l'octet 1.

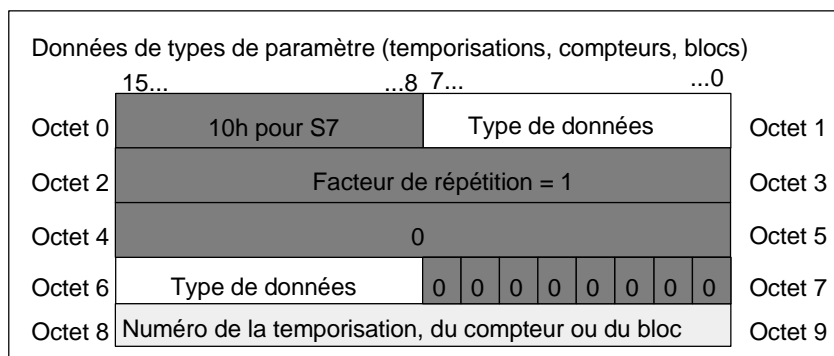
Les tableaux suivants indiquent le codage des types de données ou des zones de mémoire pour le type de paramètre ANY.

Codage des types de données		
Code hexadécimal	Type de données	Description
b#16#00	NIL	Pointeur zéro
b#16#01	BOOL	Bits
b#16#02	BYTE	Octets (8 bits)
b#16#03	CHAR	Caractères (8 bits)
b#16#04	WORD	Mots (16 bits)
b#16#05	INT	Entiers (16 bits)
b#16#06	DWORD	Mots (32 bits)
b#16#07	DINT	Entiers (32 bits)
b#16#08	REAL	Nombres à virgule flottante (32 bits)
b#16#09	DATE	Date
b#16#0A	TIME_OF_DAY (TOD)	Heure
b#16#0B	TIME	Temporisation
b#16#0C	S5TIME	Type de données S5TIME
b#16#0E	DATE_AND_TIME (DT)	Date et heure (64 bits)
b#16#13	STRING	Chaîne de caractères

Codage des zones de mémoire		
Code hexadécimal	Zone	Description
b#16#81	E	Zone de mémoire des entrées
b#16#82	A	Zone de mémoire des sorties
b#16#83	M	Zone de mémoire des mementos
b#16#84	DB	Bloc de données
b#16#85	DI	Bloc de données d'instance
b#16#86	L	Données locales (pile L)
b#16#87	V	Données locales précédentes

Format ANY pour les types de paramètre

Pour les types de paramètre, STEP 7 Lite enregistre le type de données et l'adresse des paramètres. Le facteur de répétition est toujours égal à 1. Les octets 4, 5 et 7 sont toujours à 0. Les octets 8 et 9 indiquent le numéro de la temporisation, du compteur ou du bloc.



Le tableau suivant indique le codage des types de données pour le type de paramètre ANY avec types de paramètre.

Code hexadécimal	Type de données	Description
b#16#17	BLOCK_FB	Numéro de FB
b#16#18	BLOCK_FC	Numéro de FC
b#16#19	BLOCK_DB	Numéro de DB
b#16#1A	BLOCK_SDB	Numéro de SDB
b#16#1C	COUNTER	Numéro de compteur
b#16#1D	TIMER	Numéro de temporisation

Exemple

```

Call FB 10, DB110(
:
Entree_Var_Any:= MW100,      // FB10 utilisera ici un mot (MW100)
                             // pour le traitement
:
);
:
Call FB 10, DB110(
:
Entree_Var_Any:= M1.3,      // FB10 utilisera ici un bit (M1.3)
                             // pour le traitement
:
);

```

A.3.4.6 Utilisation du type de paramètre ANY

Vous pouvez définir, pour un bloc, des paramètres formels acceptant des paramètres effectifs de n'importe quel type de données. Cela s'avère surtout utile lorsque le type de données du paramètre effectif fourni lors de l'appel du bloc est inconnu ou peut varier (et lorsque tout type de données est acceptable). Dans la déclaration des variables du bloc, vous déclarez le paramètre comme type de données ANY. Vous pourrez ainsi lui affecter un paramètre effectif d'un type de données quelconque dans STEP 7 Lite.

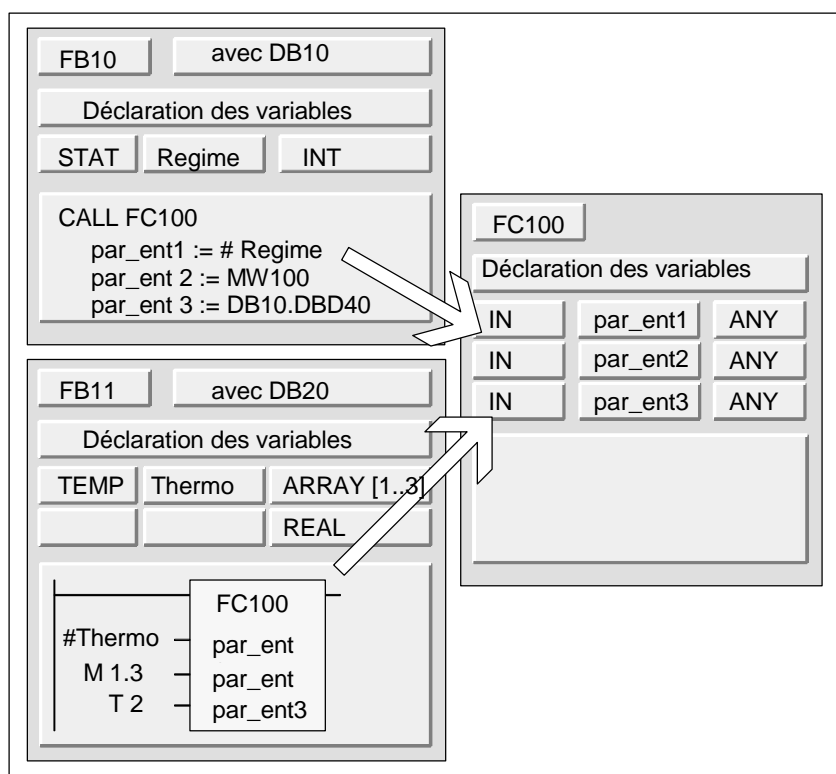
STEP 7 Lite alloue 80 bits de mémoire à une variable de type ANY. Lorsque vous affectez un paramètre effectif à un tel paramètre formel, STEP 7 Lite code l'adresse de départ, le type de données et la longueur du paramètre effectif dans ces 80 bits. Le bloc appelé analysera ces 80 bits de données enregistrées pour le paramètre ANY afin d'obtenir les renseignements nécessaires au traitement supplémentaire.

Transmission d'un paramètre effectif à un paramètre ANY

En déclarant un paramètre formel de type de données ANY, vous pouvez lui affecter un paramètre effectif de n'importe quel type de données. Vous pouvez indiquer des paramètres effectifs de types de données suivants dans STEP 7 Lite :

- types de données simples : vous indiquez l'adresse absolue ou le mnémonique du paramètre effectif ;
- types de données complexes : vous entrez le mnémonique correspondant (par ex. pour tableau ou structure) ;
- temporisations, compteurs et blocs : vous précisez leur numéro (par ex. T1, Z20 ou FB6) ;

La figure ci-après montre comment transmettre des données à une FC avec des paramètres de type ANY.



Dans cet exemple, FC100 a trois paramètres (*par_ent1*, *par_ent2* et *par_ent3*) qui ont été déclarés comme types de données ANY.

- Lorsque le bloc fonctionnel FB10 appelle la fonction FC100, il transmet un nombre entier (variable statique "Regime"), un mot (MW100) et un double mot du DB10 (DB10.DBD40).
- Lorsque le bloc fonctionnel FB11 appelle la fonction FC100, il transmet un tableau de nombres réels (variable temporaire "Thermo"), une valeur booléenne (M 1.3) et une temporisation (T2).

Indication d'une zone de données pour un paramètre ANY

Vous pouvez non seulement affecter des opérandes individuels à un paramètre ANY (par ex. MW100), mais aussi indiquer une zone de données. Vous devez utiliser, à cet effet, la notation de constante ci-après pour préciser la quantité de données à transmettre :

p# code-zone octet.bit type-données facteur-répétition

Pour l'élément *type-données*, vous pouvez indiquer en notation de constante tous les types de données simples ainsi que le type de données DATE_AND_TIME. Si le type de données n'est pas BOOL, il faut préciser l'adresse de bit 0 (x.0). Le tableau ci-après présente des exemples de notation constante pour indiquer les zones de mémoire à transmettre à un paramètre ANY.

Paramètre effectif	Description
p# M 50.0 BYTE 10	Indique 10 octets dans la zone de mémoire "Mémentos" : de MB50 à MB59
p# DB10.DBX5.0 S5TIME 3	Indique 3 unités de données de type S5TIME enregistrées dans DB10 : de DB octet 5 à DB octet 10
p# A 10.0 BOOL 4	Indique 4 bits dans la zone de mémoire "Sorties" : de A 10.0 à A 10.3

Exemple d'utilisation du type de paramètre ANY

L'exemple suivant montre comment vous pouvez copier une zone de mémoire de 10 octets en utilisant le type de paramètre ANY et la fonction système SFC20 BLKMOV.

LIST	Signification
FUNCTION FC 10: VOID	
VAR_TEMP	
Source : ANY;	
Destination :	
ANY;	
END_VAR	
BEGIN	//Charger l'adresse de début du pointeur ANY dans AR1.
LAR1 P#Source;	
	//Charger l'ID de syntaxe et
L B#16#10;	//la transférer dans le pointeur ANY.
T LB[AR1,P#0.0];	
	//Charger le type de données octet et
L B#16#02;	//le transférer dans le pointeur ANY.
T LB[AR1,P#1.0];	
	//Charger 10 octets et
L 10;	//les transférer dans le pointeur ANY.
T LW[AR1,P#2.0];	
	//La source est DB22, DBB11
L 22;	
T LW[AR1,P#4.0];	
L P#DBX11.0;	
T LD[AR1,P#6.0];	
	//Charger l'adresse de début du pointeur ANY dans AR1.
LAR1 P#Destination;	
	//Charger l'ID de syntaxe et
L B#16#10;	//la transférer dans le pointeur ANY.
T LB[AR1,P#0.0];	
	//Charger le type de données octet et
L B#16#02;	//le transférer dans le pointeur ANY.
T LB[AR1,P#1.0];	
	//Charger 10 octets et
L 10;	//les transférer dans le pointeur ANY.
T LW[AR1,P#2.0];	
	//La destination est DB33, DBB202
L 33;	
T LW[AR1,P#4.0];	
L P#DBX202.0;	
T LD[AR1,P#6.0];	
	//Appel de la fonction système Blockmove
CALL SFC 20 (
SRCBLK := Source,	//Exploitation du bit RB et du MW 12
RET_VAL := MW 12,	
DSTBLK := destin.	
);	
END_FUNCTION	

Blocs fonctionnels système (SFB) et fonctions système (SFC)

A.3.4.7 Affectation de types de données aux données locales de blocs de code

STEP 7 Lite limite les types de données - simples, complexes et paramètres - pouvant être affectés aux données locales d'un bloc dans la déclaration des variables.

Types autorisés pour les données locales d'un OB

Le tableau ci-après présente les restrictions (—) s'appliquant aux données locales des blocs d'organisation OB. Un OB ne pouvant être appelé, il ne peut pas avoir de paramètres (entrée, sortie ou entrée/sortie). Un OB n'ayant pas de DB d'instance, vous ne pouvez pas déclarer de variables statiques. Les variables temporaires d'un OB peuvent être de type de données simple, complexe ou ANY.

Les affectations valides sont indiquées par le symbole ●.

Type de déclaration	Types de données simples	Types de données complexes	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrée	—	—	—	—	—	—	—
Sortie	—	—	—	—	—	—	—
Entrée/sortie	—	—	—	—	—	—	—
Statique	—	—	—	—	—	—	—
Temporaire	●(1)	●(1)	—	—	—	—	●(1)
(1) Méorisé dans la pile L de l'OB							

Types autorisés pour les données locales d'un FB

Le tableau ci-près présente les restrictions (—) s'appliquant aux données locales des blocs fonctionnels FB. Les FB disposant d'un DB d'instance, les restrictions sont moins nombreuses. S'il n'y en a pas pour la déclaration de paramètres d'entrée, vous ne pouvez toutefois déclarer aucun type de paramètre pour les paramètres de sortie et seuls les types de paramètre POINTER et ANY sont autorisés pour les paramètres d'entrée/sortie. Vous pouvez déclarer des variables temporaires de type ANY, tous les autres types de paramètre étant interdits.

Les affectations valides sont indiquées par le symbole ●.

Type de déclaration	Types de données simples	Types de données complexes	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrée	●	●	●	●	●	●	●
Sortie	●	●	—	—	—	—	—
Entrée/sortie	●	●(1)(3)	—	—	—	●	●
Statique	●	●	—	—	—	—	—
Temporaire	●(2)	●(2)	—	—	—	—	●(2)
(1) Méorisé dans le DB d'instance comme renvoi (pointeur 48 bits)							
(2) Méorisé dans la pile L du FB							
(3) Les types STRING peuvent être définis seulement de la longueur par défaut.							

Types autorisés pour les données locales d'une FC

Le tableau ci-près présente les restrictions (—) s'appliquant aux données locales des fonctions FC. Une FC n'ayant pas de DB d'instance, vous ne pouvez pas déclarer de variables statiques. Les types de paramètre POINTER et ANY sont valables pour les paramètres d'entrée, de sortie et d'entrée/sortie. Vous pouvez également déclarer des variables temporaires de type de paramètre ANY.

Les affectations valides sont indiquées par le symbole ●.

Type de déclaration	Types de données simples	Types de données complexes	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrée	●	●(2)	●	●	●	●	●
Sortie	●	●(2)	—	—	—	●	●
Entrée/sortie	●	●(2)	—	—	—	●	●
Temporaire	●(1)	●(1)	—	—	—	—	●(1)
(1) Méorisé dans la pile L de la FC							
(2) Les types STRING peuvent être définis seulement de la longueur par défaut.							

A.3.4.8 Types de données autorisés pour la transmission de paramètres

Règles de transmission des paramètres entre blocs

Lorsque vous affectez des paramètres effectifs à des paramètres formels, vous pouvez indiquer soit une adresse absolue, soit un mnémonique, soit une constante. Ces différents types d'affectation ne sont pas autorisés pour tous les paramètres dans STEP 7 Lite. Il est interdit, par exemple, d'affecter des valeurs constantes à des paramètres de sortie ou d'entrée/sortie puisque de tels paramètres doivent, par définition, changer de valeur. Ces restrictions concernent surtout les paramètres de type de données complexe auxquels on ne peut affecter ni adresse absolue, ni constante.

Les tableaux ci-après récapitulent les restrictions (—) s'appliquant aux types de données de paramètres effectifs affectés à des paramètres formels.

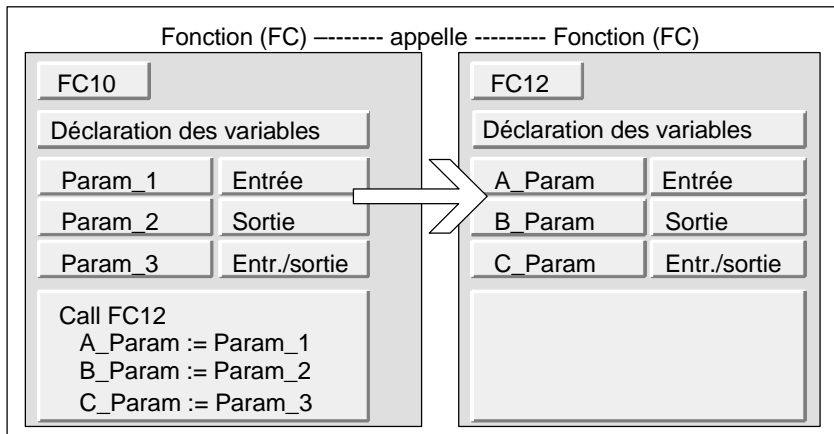
Les affectations valides sont indiquées par le symbole ●.

Types de données simples				
Type de déclaration	Adresse absolue	Mnémonique (dans table mnémos)	Mnémonique local bloc	Constante
Entrée	●	●	●	●
Sortie	●	●	●	—
Entrée/sortie	●	●	●	—

Types de données complexes				
Type de déclaration	Adresse absolue	Mnémonique de l'élément du DB (dans table mnémo.)	Mnémonique local bloc	Constante
Entrée	—	●	●	—
Sortie	—	●	●	—
Entrée/sortie	—	●	●	—

Types de données autorisés pour l'appel d'une FC par une autre FC

Vous pouvez affecter les paramètres formels d'une FC appelante aux paramètres formels de la FC appelée. La figure ci-après montre les paramètres formels de la FC10 qui sont affectés en tant que paramètres effectifs aux paramètres formels de la FC12.



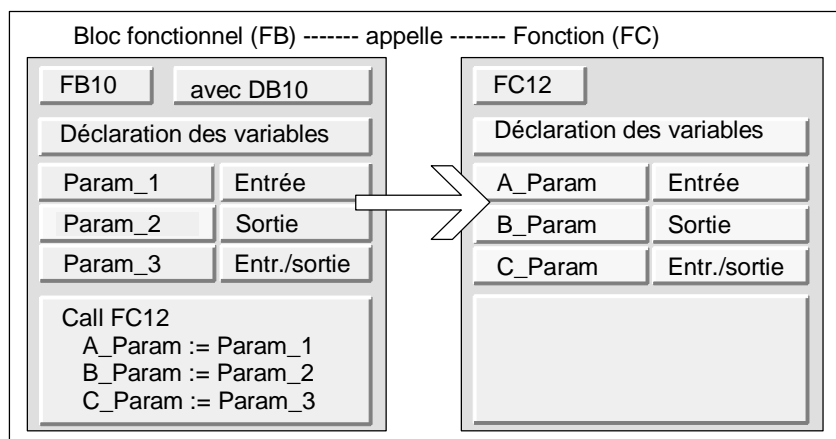
STEP 7 Lite impose toutefois des restrictions dans ce domaine. Ainsi, vous ne pouvez pas affecter en tant que paramètres effectifs des paramètres de type de données complexe ou de type de paramètre.

Le tableau ci-après montre les types de données autorisés (●) lorsqu'une fonction appelle une autre fonction.

Type de déclaration	Types de données simples	Types de données complexes	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrée → entrée	●	—	—	—	—	—	—
Entrée → sortie	—	—	—	—	—	—	—
Entrée → entrée/sortie	—	—	—	—	—	—	—
Sortie → entrée	—	—	—	—	—	—	—
Sortie → sortie	●	—	—	—	—	—	—
Sortie → entrée/sortie	—	—	—	—	—	—	—
Entrée/sortie → entrée	●	—	—	—	—	—	—
Entrée/sortie → sortie	●	—	—	—	—	—	—
Entrée/sortie → entrée/sortie	●	—	—	—	—	—	—

Types de données autorisés pour l'appel d'une FC par un FB

Vous pouvez affecter les paramètres formels d'un FB appelant aux paramètres formels de la FC appelée. La figure ci-après montre les paramètres formels du FB10 qui sont affectés en tant que paramètres effectifs aux paramètres formels de la FC12.

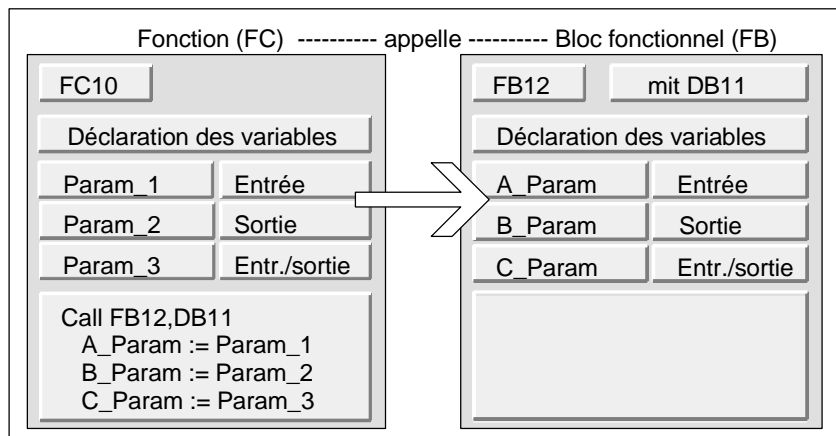


STEP 7 Lite impose toutefois des restrictions dans ce domaine. Ainsi, vous ne pouvez pas affecter en tant que paramètres effectifs des paramètres de type de paramètre. Le tableau ci-après montre les types de données autorisés (●) lorsque qu'un bloc fonctionnel appelle une fonction.

Type de déclaration	Types de données simples	Types de données complexes	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrée → entrée	●	●	—	—	—	—	—
Entrée → sortie	—	—	—	—	—	—	—
Entrée → entrée/sortie	—	—	—	—	—	—	—
Sortie → entrée	—	—	—	—	—	—	—
Sortie → sortie	●	●	—	—	—	—	—
Sortie → entrée/sortie	—	—	—	—	—	—	—
Entrée/sortie → entrée	●	—	—	—	—	—	—
Entrée/sortie → sortie	●	—	—	—	—	—	—
Entrée/sortie → entrée/sortie	●	—	—	—	—	—	—

Types de données autorisés pour l'appel d'un FB par une FC

Vous pouvez affecter les paramètres formels d'une FC appelante aux paramètres formels du FB appelé. La figure ci-après montre les paramètres formels de la FC10 qui sont affectés en tant que paramètres effectifs aux paramètres formels du FB12.



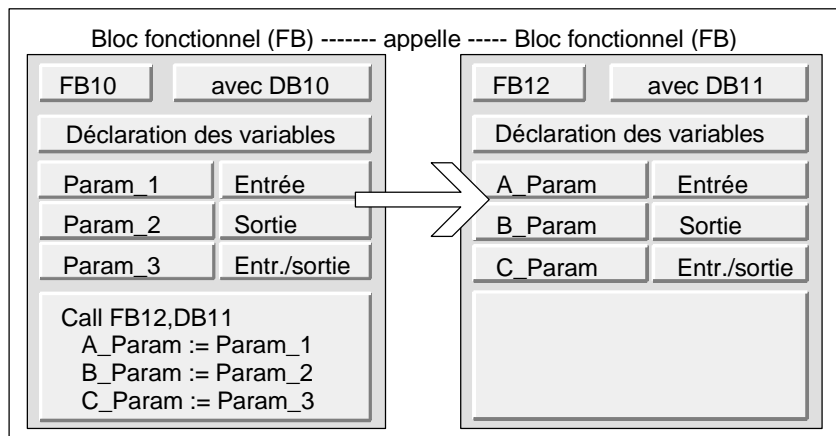
STEP 7 Lite impose toutefois des restrictions dans ce domaine. Ainsi, vous ne pouvez pas affecter en tant que paramètres effectifs des paramètres de type de données complexe. Mais vous pouvez affecter des paramètres d'entrée de type de paramètre TIMER, COUNTER et BLOCK aux paramètres d'entrée du FB appelé.

Le tableau ci-après montre les types de données autorisés (●) lorsqu'une fonction appelle un bloc fonctionnel.

Type de déclaration	Types de données simples	Types de données complexes	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrée → entrée	●	—	●	●	●	—	—
Entrée → sortie	—	—	—	—	—	—	—
Entrée → entrée/sortie	—	—	—	—	—	—	—
Sortie → entrée	—	—	—	—	—	—	—
Sortie → sortie	●	—	—	—	—	—	—
Sortie → entrée/sortie	—	—	—	—	—	—	—
Entrée/sortie → entrée	●	—	—	—	—	—	—
Entrée/sortie → sortie	●	—	—	—	—	—	—
Entrée/sortie → entrée/sortie	●	—	—	—	—	—	—

Types de données autorisés pour l'appel d'un FB par un autre FB

Vous pouvez affecter les paramètres formels d'un FB appelant aux paramètres formels du FB appelé. La figure ci-après montre les paramètres formels du FB10 qui sont affectés en tant que paramètres effectifs aux paramètres formels du FB12.



STEP 7 Lite impose toutefois des restrictions dans ce domaine. Ainsi, vous ne pouvez pas affecter, en tant que paramètres effectifs, des paramètres d'entrée et de sortie de type de données complexe aux paramètres d'entrée et de sortie du FB appelé. Mais vous pouvez affecter des paramètres d'entrée de type de paramètre TIMER, COUNTER et BLOCK aux paramètres d'entrée du FB appelé.

Le tableau ci-après montre les types de données autorisés (●) lorsqu'un bloc fonctionnel appelle un autre bloc fonctionnel.

Type de déclaration	Types de données simples	Types de données complexes	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre	Type de paramètre
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrée → entrée	●	●	●	●	●	—	—
Entrée → sortie	—	—	—	—	—	—	—
Entrée → entrée/sortie	—	—	—	—	—	—	—
Sortie → entrée	—	—	—	—	—	—	—
Sortie → sortie	●	●	—	—	—	—	—
Sortie → entrée/sortie	—	—	—	—	—	—	—
Entrée/sortie → entrée	●	—	—	—	—	—	—
Entrée/sortie → sortie	●	—	—	—	—	—	—
Entrée/sortie → entrée/sortie	●	—	—	—	—	—	—

A.3.4.9 Transmission au paramètre IN_OUT d'un FB

Pour la transmission de types de données complexes au paramètre IN_OUT d'un bloc fonctionnel (FB), c'est l'adresse d'opérande de la variable qui est transmise (call by reference).

Pour la transmission de types de données simples au paramètre IN_OUT d'un FB, les valeurs sont copiées dans le bloc de données d'instance avant l'exécution du FB et extraites du bloc de données d'instance lorsque l'exécution du FB est terminée.

Ainsi, les variables IN_OUT de type de données simple peuvent être initialisées avec une valeur.

Dans un appel, il n'est toutefois pas possible d'indiquer une constante comme paramètre effectif à la place d'une variable IN_OUT, car une constante ne peut être écrasée.

Les variables de type de données STRUCT ou ARRAY ne peuvent pas être initialisées, car dans ce cas le bloc de données d'instance ne contient qu'une seule adresse.

A.4 Exemples de programme

A.4.1 Exemples de projet et de programme

Le CD d'installation contient de nombreux exemples de projet. Les descriptions de projet qui ne sont pas données dans le présent chapitre figurent dans l'OB1 correspondant.

Exemples et exemples de projet	Contenus sur le CD	Décrits dans le présent chapitre
Projets "Getting Started CONT", „Getting Started LOG" et "Getting Started LIST"	•	Manuel distinct "Getting Started avec STEP 7 Lite"
Exemple de processus de mélange industriel		•
Exemple d'utilisation d'alarmes horaires		•
Exemple d'utilisation d'alarmes temporisées		•

Ces exemples ne visent pas à transmettre un style de programmation ou une compétence technique dans la commande d'un processus particulier, mais bien plutôt à montrer quelles étapes doivent être réalisées pour concevoir un programme.

Suppression et installation de projets-exemples fournis

Vous pouvez supprimer, puis à nouveau réinstaller les projets-exemples fournis. Pour l'installation, vous démarrez le programme Setup de STEP 7 Lite. Vous pouvez sélectionner les exemples de projet à réinstaller.

Les projets-exemples installés se trouvent sur le lecteur où vous avez installé STEP 7 Lite sous \Siemens\S7lite\Exemples.

Nota

Les exemples de projet fournis sont copiés, à moins qu'ils soient désélectionnés, lors de l'installation de STEP 7 Lite. Si vous en avez modifié certains, ils seront remplacés par les originaux lors d'une nouvelle installation de STEP 7 Lite.

Pour cette raison, il est recommandé de copier les exemples de projet fournis avant de les modifier et d'éditer uniquement la copie.

A.4.2 Exemple de masquage et de démasquage d'événements d'erreurs synchrones

Dans l'exemple suivant d'un programme utilisateur, nous allons vous montrer le masquage et le démasquage d'événements d'erreurs synchrones. La SFC36 "MSK_FLT" masque les erreurs suivantes dans le masque d'erreurs de programmation :

- Erreur de longueur de zone lors de la lecture
- Erreur de longueur de zone lors de l'écriture

Un second appel de la SFC36 "MSK_FLT" masque en plus une erreur d'accès :

- Erreur d'accès à la périphérie lors de l'écriture

La SFC38 "READ_ERR" interroge les événements d'erreurs synchrones masqués. L'erreur d'accès à la périphérie lors de l'écriture est à nouveau démasquée par la SFC37 "DMSK_FLT".

Instruction

La suite représente l'OB1, dans lequel l'exemple pour le programme utilisateur a été programmé en LIST.

LIST (réseau 1)	Signification
UN M 255.0	//Mémento non rémanent M 255.0
	//(uniquement lors du premier cycle=0)
SPBNB m001	
CALL SFC 36	//SFC36 MSK_FLT (masquage d'événements
	//d'erreurs synchrones)
PRGFLT_SET_MASK :=DW#16#C	//Bit2=Bit3=1 (BLFL et BLFS sont
	//masqués)
ACCFLT_SET_MASK :=DW#16#0	//Tous les bits=0 (aucune erreur d'accès
	//n'est masquée)
VAL_RET :=MW 100	//Valeur en retour
PRGFLT_MASKED :=MD 10	//Affichage du masque d'erreurs de
	//programmation actuel dans MD 10
ACCFLT_MASKED :=MD 14	//Affichage du masque d'erreurs d'accès
	//actuel dans MD 14
m001: U RB	
S M 255.0	//Mise à 1 de M255.0 si masquage réussi

LIST (réseau 2)	Signification
CALL SFC 36	//SFC36 MSK_FLT (masquage d'événements
	//d'erreurs synchrones)
PRGFLT_SET_MASK :=DW#16#0	//Tous les bits=0 (aucune autre erreur
	//de programmation n'est masquée)
ACCFLT_SET_MASK :=DW#16#8	//Bit3=1 (les erreurs d'accès en
	//écriture sont masquées)
VAL_RET :=MW 102	//Valeur en retour
PRGFLT_MASKED :=MD 20	//Affichage du masque d'erreurs de
	//programmation actuel dans MD 20
ACCFLT_MASKED :=MD 24	//Affichage du masque d'erreurs d'accès
	//actuel dans MD 24

LIST (réseau 3)	Signification
UN M 27.3	//Fin du bloc, si erreur d'accès en
BEB	//écriture (bit3 dans ACCFLT_MASKED) non
	//masquée

LIST (réseau 4)	Signification
L B#16#0	
T PAB 16	//Accès en écriture (avec valeur 0)
	//à PAB 16

LIST (réseau 5)		Signification
CALL	SFC 38	//SFC38 READ_ERR (interrogation
		//d'événements d'erreurs synchrones)
PRGFLT_QUERY	:=DW#16#0	//Tous les bits=0 (aucune interrogation
		//d'erreurs de programmation)
ACCFLT_QUERY	:=DW#16#8	//Bit3=1 (interrogation d'erreur d'accès
		//en écriture)
VAL_RET	:=MW 104	//Valeur en retour
PRGFLT_CLR	:=MD 30	//Affichage du masque d'erreurs de
		//programmation actuel dans MD 30
ACCFLT_CLR	:=MD 34	//Affichage du masque d'erreurs d'accès
		//actuel dans MD 34
U	RB	//Aucune erreur survenue ni erreur
		//d'accès en écriture détectée
U	M 37.3	
NOT		//Inversion du RLG
=	M 0.0	//M 0.0=1, si PAB 16 existant
LIST (réseau 6)		Signification
L	B#16#0	
T	PAB 17	//Erreur d'accès en écriture (avec
		//valeur 0) à PAB 17
LIST (réseau 7)		Signification
CALL	SFC 38	//SFC38 READ_ERR (interrogation
		//d'événements d'erreurs synchrones)
PRGFLT_QUERY	:=DW#16#0	//Tous les bits=0 (aucune interrogation
		//d'erreurs de programmation)
ACCFLT_QUERY	:=DW#16#8	//Bit3=1 (interrogation d'erreur d'accès
		//en écriture)
VAL_RET	:=MW 104	//Valeur en retour
PRGFLT_CLR	:=MD 30	//Affichage du masque d'erreurs de
		//programmation actuel dans MD 30
ACCFLT_CLR	:=MD 34	//Affichage du masque d'erreurs d'accès
		//actuel dans MD 34
U	RB	//Aucune erreur survenue ni erreur
		//d'accès en écriture détectée
U	M 37.3	
NOT		//Inversion du RLG
=	M 0.1	//M 0.1=1, si PAB 17 existant
LIST (réseau 8)		Signification
L	B#16#0	
T	PAB 18	//Accès en écriture (avec valeur 0)
		//à PAB 18
LIST (réseau 9)		Signification
CALL	SFC 38	//SFC38 READ_ERR (interrogation
		//d'événements d'erreurs synchrones)
PRGFLT_QUERY	:=DW#16#0	//Tous les bits=0 (aucune interrogation
		//d'erreurs de programmation)
ACCFLT_QUERY	:=DW#16#8	//Bit3=1 (interrogation d'erreur d'accès
		//en écriture)
VAL_RET	:=MW 104	//Valeur en retour
PRGFLT_CLR	:=MD 30	//Affichage du masque d'erreurs de
		//programmation actuel dans MD 30
ACCFLT_CLR	:=MD 34	//Affichage du masque d'erreurs d'accès
		//actuel dans MD 34
U	RB	//Aucune erreur survenue ni erreur
		//d'accès en écriture détectée
U	M 37.3	
NOT		//Inversion du RLG
=	M 0.2	//M 0.2=1, si PAB 18 existant
LIST (réseau 10)		Signification
L	B#16#0	
T	PAB 19	//Accès en écriture (avec valeur 0)
		//à PAB 19
LIST (réseau 11)		Signification

CALL	SFC 38	//SFC38 READ_ERR (interrogation //d'événements d'erreurs synchrones)
PRGFLT_QUERY	:=DW#16#0	//Tous les bits=0 (aucune interrogation //d'erreur de programmation)
ACCFLT_QUERY	:=DW#16#8	//Bit3=1 (interrogation d'erreur d'accès //en écriture)
VAL_RET	:=MW 104	//Valeur en retour
PRGFLT_CLR	:=MD 30	//Affichage du masque d'erreurs de //programmation actuel dans MD 30
ACCFLT_CLR	:=MD 34	//Affichage du masque d'erreurs d'accès //actuel dans MD 34
U	RB	//Aucune erreur survenue ni erreur //d'accès en écriture détectée
U	M 37.3	
NOT		//Inversion du RLG
=	M 0.3	//M 0.3=1, si PAB 19 existant

LIST (réseau 12)	Signification
CALL SFC 37	//SFC37 DMSK FLT (démasquage //d'événements d'erreurs synchrones)
PRGFLT_RESET_MASK :=DW#16#0	//Tous les bits=0 (aucun démasquage //d'erreur de programmation)
ACCFLT_RESET_MASK :=DW#16#8	//Bit3=1 (démasquage d'erreur d'accès en //écriture)
VAL_RET :=MW 102	//Valeur en retour
PRGFLT_MASKED :=MD 20	//Affichage du masque d'erreurs de //programmation actuel dans MD 20
ACCFLT_MASKED :=MD 24	//Affichage du masque d'erreurs d'accès //actuel dans MD 24

LIST (réseau 13)	Signification
U M 27.3	//Fin de bloc, si erreur d'accès en
BEB	//écriture (bit3 dans ACCFLT_MASKED) non //démasquée

LIST (réseau 14)	Signification
U M 0.0	
SPBNB m002	
L EB 0	//Transférer EB 0 dans PAB 16, si
T PAB 16	//existant
m002: NOP 0	

LIST (réseau 15)	Signification
U M 0.1	
SPBNB m003	
L EB 1	//Transférer EB 1 dans PAB 17, si
T PAB 17	//existant
m003: NOP 0	

LIST (réseau 16)	Signification
U M 0.2	
SPBNB m004	
L EB 2	//Transférer EB 2 dans PAB 18, si
T PAB 18	//existant
m004: NOP 0	

LIST (réseau 17)	Signification
U M 0.3	
SPBNB m005	
L EB 3	//Transférer EB 3 dans PAB 19, si
T PAB 19	//existant
m005: NOP 0	

Blocs fonctionnels système (SFB) et fonctions système (SFC)

A.4.3 Exemple d'inhibition et de validation d'événements d'alarme et d'événements asynchrones (SFC 39 et 40)

Dans cet exemple de programme utilisateur, on considère qu'une partie du programme ne doit pas être interrompue par des alarmes. Pour cette partie du programme, la SFC39 "DIS_IRT" inhibe les appels de l'OB35 (alarme horaire) et la SFC40 "EN_IRT" valide à nouveau les appels de l'OB35.

Les SFC39 et SFC40 sont appelées dans l'OB1 :

LIST (OB 1)	Signification
U M 0.0	//Partie du programme pouvant être
S M 90.1	//interrompue
U M 0.1	
S M 90.0	
:	
:	//Partie du programme ne devant pas être
	//interrompue par des alarmes :
CALL SFC 39	//Inhiber et rejeter les alarmes
MODE :=B#16#2	//Mode 2 : inhiber OB alarme individuels
NR_OB :=35	//Inhibition de l'OB35
VAL_RET :=MW 100	
:	
:	
L PEW 100	
T MW 200	
L MW 90	
T MW 92	
:	
:	
CALL SFC 40	//Validation des alarmes
MODE :=B#16#2	//Mode 2 : valider OB alarme individuels
NR_OB :=35	//Validation de l'OB35
VAL_RET :=MW 102	
	//Partie du programme pouvant être
	//interrompue
U M 10.0	
S M 190.1	
U M 10.1	
S M 190.0	
:	
:	

A.4.4 Exemple de traitement différé d'événements d'alarme et d'événements asynchrones (SFC 41 et 42)

Dans cet exemple de programme utilisateur, on considère qu'une partie du programme ne doit pas être interrompue par des alarmes. Dans cette partie du programme, la SFC41 "DIS_AIRT" retarde les alarmes qui sont ultérieurement validées par la SFC42 "EN_AIRT".

La SFC41 et la SFC42 sont appelées dans l'OB1 :

LIST (OB 1)		Signification
U	M 0.0	//Partie du programme pouvant être
S	M 90.1	//interrompue
U	M 0.1	
S	M 90.0	
:		
:		//Partie du programme ne devant pas être
		//interrompue par des alarmes
CALL	SFC 41	//Inhibition et retardement des alarmes
VAL_RET	:=MW 100	
L	PEW 100	
T	MW 200	
L	MW 90	
T	MW 92	
:		
:		
CALL	SFC 42	//Validation de l'alarme
VAL_RET	:=MW 102	
L	MW 100	//La valeur en retour contient le nombre
		//d'inhibitions d'alarmes utilisées
DEC	1	
L	MW 102	//La valeur en retour contient le nombre
		//d'inhibitions d'alarmes utilisées
<>I		//Après validation des alarmes, ce
		//nombre doit être le même qu'avant
SPB	err	//l'inhibition des alarmes (ici, "0")
U	M 10.0	//Partie du programme pouvant être
S	M 190.1	//interrompue
U	M 10.1	
S	M 190.0	
:		
:		
BEA		
erre:	L MW 102	//Le nombre d'inhibitions d'alarmes
	T AW 12	//utilisées est indiqué

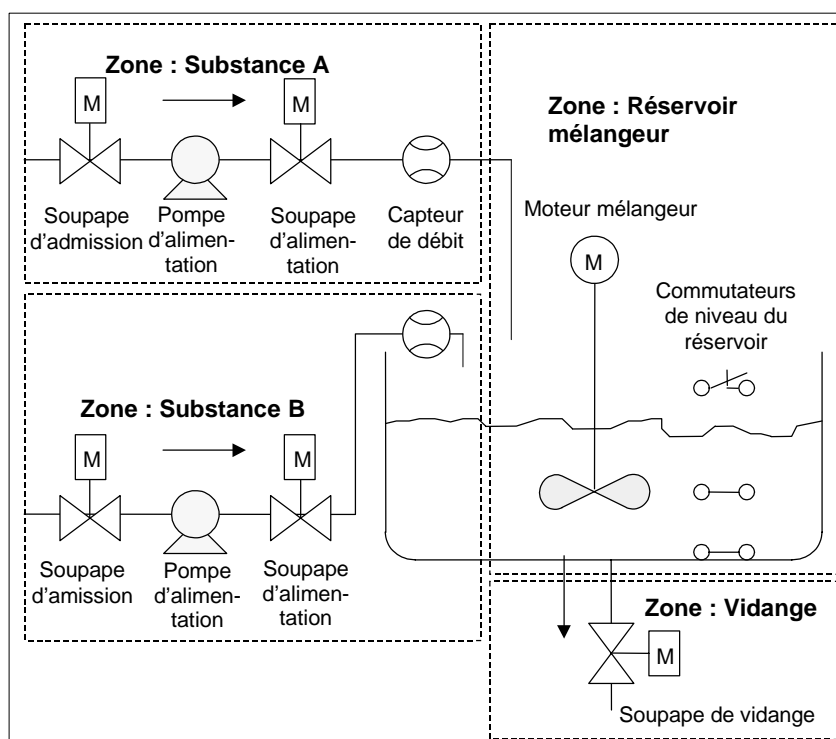
A.4.5 Exemple de programme pour un processus de mélange industriel

A.4.5.1 Exemple de programme pour un processus de mélange industriel

Notre exemple de programme se base sur les informations de la première partie du manuel relatives à la commande d'un processus de mélange industriel.

Problème posé

Deux substances (A et B) doivent être mélangées par un moteur mélangeur dans un réservoir. Cette masse doit ensuite s'écouler du réservoir par une soupape de vidange. La figure ci-après montre un diagramme de notre exemple de processus.



Description des processus partiels

Nous vous avons expliqué, dans la première partie du manuel, comment subdiviser l'exemple de processus en zones fonctionnelles et en différentes tâches. Voici la description des différentes zones.

Zones pour substances A et B

- Les conduites d'amenée des substances doivent comporter une soupape d'admission, une soupape d'alimentation ainsi qu'une pompe d'alimentation.
- Dans ces conduites se trouvent des capteurs de débit.
- La mise en marche des pompes d'alimentation doit être inhibée lorsque le capteur de niveau indique "Réservoir plein".
- La mise en marche des pompes d'alimentation doit être inhibée lorsque la soupape de vidange est ouverte.
- Les soupapes d'admission et d'alimentation doivent être ouvertes au plus tôt 1 seconde après le déclenchement de la pompe d'alimentation.
- Les soupapes doivent être fermées immédiatement après l'arrêt des pompes d'alimentation (signal du capteur de débit) afin d'éviter un écoulement de la substance en provenance de la pompe.
- Le déclenchement des pompes est surveillé par une temporisation : le capteur de débit doit signaler un débit 7 secondes au maximum après ce déclenchement.
- Les pompes d'alimentation doivent être arrêtées le plus rapidement possible lorsque les capteurs de débit ne signalent plus de débit pendant le fonctionnement des pompes.
- Le nombre de démarrages des pompes d'alimentation doit être comptabilisé (période de maintenance).

Zone Réservoir de mélange

- Le déclenchement du moteur mélangeur doit être verrouillé lorsque le capteur de niveau indique "Réservoir en dessous du minimum" ou lorsque la soupape de vidange est ouverte.
- Le moteur mélangeur émet un signal en retour une fois le régime nominal atteint. S'il n'émet pas ce signal 10 secondes au maximum après l'activation du moteur, il faut l'arrêter.
- Le nombre de démarrages du moteur mélangeur doit être comptabilisé (période de maintenance).
- Le réservoir de mélange doit comporter trois capteurs :
 - Réservoir plein : contact à ouverture. Lorsque le niveau maximal est atteint, le contact est ouvert.
 - Niveau dans le réservoir en dessous du minimum : contact à fermeture. Lorsque le niveau minimal est atteint, le contact est fermé.
 - Réservoir pas vide : contact à fermeture. Le contact est fermé si le réservoir n'est pas vide.

Zone Vidange

- La vidange doit être commandée par soupape magnétique.
- La soupape magnétique est commandée par l'opérateur, mais doit être refermée au plus tard lors du signal "Réservoir vide".
- L'ouverture de la soupape de vidange est verrouillée
 - lorsque le moteur mélangeur fonctionne ;
 - lorsque le réservoir est vide.

Poste d'opération

Il faut également installer un poste d'opération pour que l'opérateur puisse démarrer et arrêter ainsi que surveiller le processus. Ce poste d'opération comporte :

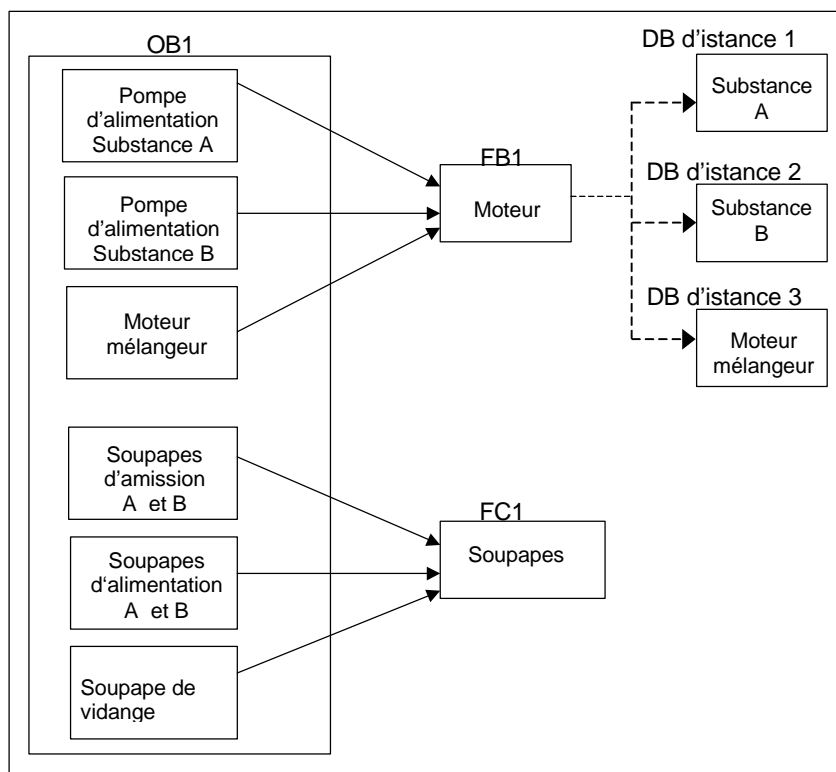
- des commutateurs pour commander les événements les plus importants Le bouton "Mettre à 0 indicateur de maintenance" permet d'éteindre les lampes de signalisation de maintenance pour les moteurs ayant besoin d'une maintenance et de mettre à zéro les valeurs correspondantes des compteurs pour l'intervalle entre les maintenances ;
- des lampes de signalisation indiquant l'état de fonctionnement,
- le commutateur d'arrêt d'urgence.

A.4.5.2 Définition de blocs de code

Vous définissez la structure de votre programme utilisateur en le répartissant dans différents blocs et en fixant la hiérarchie d'appel de ces blocs.

Hiérarchie d'appel des blocs

La figure ci-après présente la hiérarchie des blocs devant être appelés dans le programme structuré.



- OB1 : il s'agit de l'interface avec le système d'exploitation de la CPU ; il contient le programme principal. Le bloc fonctionnel FB1 et la fonction FC1 sont appelés et les paramètres spécifiques nécessaires pour la commande du processus sont transmis dans l'OB1.
- FB1 : la pompe d'alimentation pour la substance A, la pompe d'alimentation pour la substance B et le moteur mélangeur peuvent être commandés par un même bloc fonctionnel, puisque les tâches sont identiques (activation, désactivation, comptage des interventions, etc.).
- DB d'instance 1-3 : les paramètres effectifs et les données statiques pour la commande des pompes d'alimentation pour les substances A et B ainsi que pour celle du moteur mélangeur sont différents et sont donc inscrits dans trois DB d'instance affectés au FB1.
- FC1 : les soupapes d'admission et d'alimentation pour les substances A et B ainsi que la soupape de vidange utilisent également un bloc de code commun. Puisqu'il s'agit uniquement de programmer la fonction d'ouverture et de fermeture, une seule fonction suffit.

A.4.5.3 Affectation de mnémoniques

Définition de mnémoniques

Notre exemple de programme utilise des mnémoniques (ou noms symboliques) définis dans la table des mnémoniques avec STEP 7 Lite. Les tableaux ci-après présentent les mnémoniques et les adresses absolues correspondantes pour les éléments du programme utilisés.

Mnémoniques pour les pompes d'alimentation et le moteur mélangeur			
Mnémonique	Opérande	Type de données	Description
Feed_pump_A_start	E 0.0	BOOL	Commutateur bouton-poussoir de démarrage de la pompe d'alimentation pour substance A
Feed_pump_A_stop	E 0.1	BOOL	Commutateur bouton-poussoir d'arrêt de la pompe d'alimentation pour substance A
Flow_A	E 0.2	BOOL	La substance A coule.
Inlet_valve_A	A 4.0	BOOL	Commande de la soupape d'admission pour substance A
Feed_valve_A	A 4.1	BOOL	Commande de la soupape d'alimentation pour substance A
Feed_pump_A_on	A 4.2	BOOL	Lampe de signalisation "Pompe d'alimentation pour substance A en marche"
Feed_pump_A_off	A 4.3	BOOL	Lampe de signalisation "Pompe d'alimentation pour substance A arrêtée"
Feed_pump_A	A 4.4	BOOL	Commande de la pompe d'alimentation pour substance A
Feed_pump_A_fault	A 4.5	BOOL	Lampe de signalisation "Erreur de la pompe d'alimentation A"
Feed_pump_A_maint	A 4.6	BOOL	Lampe de signalisation "Maintenance de la pompe d'alimentation A"
Feed_pump_B_start	E 0.3	BOOL	Commutateur bouton-poussoir de démarrage de la pompe d'alimentation pour substance B
Feed_pump_B_stop	E 0.4	BOOL	Commutateur bouton-poussoir d'arrêt de la pompe d'alimentation pour substance B
Flow_B	E 0.5	BOOL	La substance B coule.
Inlet_valve_B	A 5.0	BOOL	Commande de la soupape d'admission pour substance B
Feed_valve_B	A 5.1	BOOL	Commande de la soupape d'alimentation pour substance B
Feed_pump_B_on	A 5.2	BOOL	Lampe de signalisation "Pompe d'alimentation pour substance B en marche"
Feed_pump_B_off	A 5.3	BOOL	Lampe de signalisation "Pompe d'alimentation pour substance B arrêtée"
Feed_pump_B	A 5.4	BOOL	Commande de la pompe d'alimentation pour substance B
Feed_pump_B_fault	A 5.5	BOOL	Lampe de signalisation "Erreur de la pompe d'alimentation B"
Feed_pump_B_maint	A 5.6	BOOL	Lampe de signalisation "Maintenance de la pompe d'alimentation B"
Agitator_running	E 1.0	BOOL	Signal en retour du moteur mélangeur

Mnémoniques pour les pompes d'alimentation et le moteur mélangeur			
Mnémonique	Opérande	Type de données	Description
Agitator_start	E 1.1	BOOL	Commutateur bouton-poussoir de démarrage du moteur mélangeur
Agitator_stop	E 1.2	BOOL	Commutateur bouton-poussoir d'arrêt du moteur mélangeur
Agitator	A 8.0	BOOL	Commande du moteur mélangeur
Agitator_on	A 8.1	BOOL	Lampe de signalisation "Moteur mélangeur en marche"
Agitator_off	A 8.2	BOOL	Lampe de signalisation "Moteur mélangeur arrêté"
Agitator_fault	A 8.3	BOOL	Lampe de signalisation "Erreur du moteur mélangeur"
Agitator_maint	A 8.4	BOOL	Lampe de signalisation "Maintenance du moteur mélangeur"

Mnémoniques pour les capteurs et les indicateurs de niveau du réservoir			
Mnémonique	Opérande	Type de données	Description
Tank_below_max	E 1.3	BOOL	Capteur "Réservoir de mélange pas plein"
Tank_above_min	E 1.4	BOOL	Capteur "Réservoir de mélange au-dessus du minimum"
Tank_not_empty	E 1.5	BOOL	Capteur "Réservoir de mélange pas vide"
Tank_max_disp	A 9.0	BOOL	Lampe de signalisation "Réservoir de mélange plein"
Tank_min_disp	A 9.1	BOOL	Lampe de signalisation "Réservoir en dessous du minimum"
Tank_empty_disp	A 9.2	BOOL	Lampe de signalisation "Réservoir de mélange vide"

Mnémoniques pour la soupape de vidange			
Mnémonique	Opérande	Type de données	Description
Drain_open	E 0.6	BOOL	Commutateur bouton-poussoir d'ouverture de la soupape de vidange
Drain_closed	E 0.7	BOOL	Commutateur bouton-poussoir de fermeture de la soupape de vidange
Drain	A 9.5	BOOL	Commande de la soupape de vidange
Drain_open_disp	A 9.6	BOOL	Lampe de signalisation "Soupape de vidange ouverte"
Drain_closed_disp	A 9.7	BOOL	Lampe de signalisation "Soupape de vidange fermée"

Mnémoniques pour les autres éléments du programme			
Mnémonique	Opérande	Type de données	Description
EMER_STOP_off	E 1.6	BOOL	Commutateur d'arrêt d'urgence
Reset_maint	E 1.7	BOOL	Bouton-poussoir de remise à zéro pour les lampes de signalisation de maintenance de tous les moteurs
Motor_block	FB1	FB1	FB pour commander pompes et moteur
Valve_block	FC1	FC1	FC pour commander les soupapes
DB_feed_pump_A	DB1	FB1	DB d'instance pour la commande de la pompe d'alimentation A
DB_feed_pump_B	DB2	FB1	DB d'instance pour la commande de la pompe d'alimentation B
DB_agitator	DB3	FB1	DB d'instance pour la commande du moteur mélangeur

A.4.5.4 Création du bloc fonctionnel pour le moteur

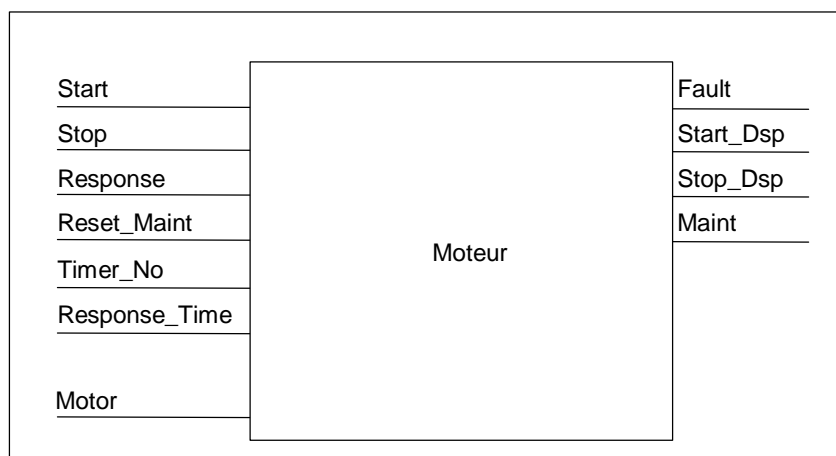
Tâches pour le FB

Le FB pour le moteur contient les fonctions logiques ci-après :

- Il existe une entrée de démarrage et une entrée d'arrêt.
- Une série de verrouillages permet le fonctionnement de l'équipement (pompes et moteur mélangeur). L'état des verrouillages est enregistré dans les données locales temporaires (pile L) de l'OB1 ("Enable_Motor") et combiné aux entrées de démarrage et d'arrêt lors de l'exécution du FB pour le moteur.
- Un signal en retour de l'équipement doit apparaître avant l'expiration d'un temps donné. Sinon, le programme considère qu'une erreur s'est produite et le moteur sera arrêté.
- Il faut définir la temporisation et la valeur de temps pour le cycle signal en retour/erreur.
- Si le bouton de démarrage est actionné et que la validation est donnée, l'appareil démarre et fonctionne jusqu'à ce que le bouton d'arrêt soit actionné.
- Une temporisation est déclenchée à la mise en marche de l'appareil. L'appareil s'arrête s'il n'émet pas de signal en retour avant que cette temporisation n'expire.

Identification des entrées et sorties

La figure ci-après montre les entrées et les sorties du FB générique pour le moteur.



Définition des paramètres pour le FB

Vous devez définir des noms de paramètres génériques pour les entrées et les sorties afin de créer un FB "Moteur" réutilisable, permettant de commander les deux pompes et le moteur mélangeur.

Dans l'exemple du processus, le FB pour le moteur doit remplir les conditions ci-après :

- Des signaux provenant du poste d'opération sont nécessaires pour le démarrage ou l'arrêt du moteur ou des pompes.
- Un signal en retour provenant des pompes ou du moteur doit indiquer que le moteur est en marche.
- Il faut calculer le temps entre l'émission du signal de mise en marche du moteur et la réception du signal en retour. En l'absence de signal en retour à l'expiration de ce temps, le moteur doit être arrêté.
- Les lampes respectives sur le poste d'opération doivent s'allumer et s'éteindre.
- Le FB fournit un signal pour la commande du moteur.

Ces conditions peuvent être définies comme entrées et sorties du bloc fonctionnel. Le tableau ci-après présente les paramètres du FB pour le moteur.

Nom du paramètre	Entrée	Sortie	Entrée/sortie
START	n		
Stop	n		
Response	n		
Reset_Maint	n		
Timer_No	n		
Response_Time	n		
Fault		n	
Start_Dsp		n	
Stop_Dsp		n	
Maint		n	
Motor			n

Déclaration des variables du FB pour le moteur

Vous devez déclarer les paramètres d'entrée, de sortie et d'entrée/sortie du FB pour le moteur.

Opérande	Déclaration	Nom	Type	Valeur initiale
0.0	In	Demarrage	BOOL	FALSE
0.1	In	Stop	BOOL	FALSE
0.2	In	Response	BOOL	FALSE
0.3	In	Reset_Maint	BOOL	FALSE
2.0	In	Time_No	TIMER	
4.0	In	Response_Time	S5TIME	S5T#0MS
6.0	Out	Fault	BOOL	FALSE
6.1	Out	Start_Dsp	BOOL	FALSE
6.2	Out	Stop_Dsp	BOOL	FALSE
6.3	Out	Maint	BOOL	FALSE

Opérande	Déclaration	Nom	Type	Valeur initiale
8.0	in_out	Motor	BOOL	FALSE
10.0	stat	Time_bin	WORD	W#16#0
12.0	stat	Time_BCD	WORD	W#16#0
14.0	stat	Starts	INT	0
16.0	stat	Start_Edge	BOOL	FALSE

Pour les FB, les variables d'entrée, de sortie, d'entrée/sortie et statiques sont contenues dans le DB d'instance indiqué dans l'opération d'appel. Quant aux variables temporaires, elles se trouvent dans la pile L.

Programmation du FB pour le moteur

Dans STEP 7 Lite, il faut créer les blocs appelés par d'autres blocs avant les blocs contenant l'appel. Vous devez donc, dans notre exemple de programme, écrire le FB pour le moteur avant l'OB1.

La section des instructions du FB1 se présente comme suit en langage de programmation LIST.

Réseau 1 Démarrage/arrêt et maintien

```

U(
O #Start
O #Motor
)
UN#Stop
= #Motor

```

Réseau 2 Surveillance de la mise en route

```

U #Motor
L #Response_Time
SE #Timer_No
UN#Motor
R #Timer_No
L #Timer_No
T #Timer_bin
LC #Timer_No
T #Timer_BCD
U #Timer_No
UN#Response
S #Fault
R #Motor

```

Réseau 3 Lampe de démarrage et remise à zéro erreurs

```

U #Response
= #Start_Dsp
R #Fault

```

Réseau 4 Lampe d'arrêt

```

UN#Response
= #Stop_Dsp

```

Réseau 5 Comptage des démarrages

```

U  #Motor
FP #Start_Edge
SPBN    lab1
L  #Starts
+  1
T  #Starts

lab1: NOP 0

```

Réseau 6 Lampe de signalisation de maintenance

```

L  #Starts
L  50
>=I
=  #Maint

```

Réseau 7 Remise à zéro du compteur du nombre de démarrages

```

U  #Reset_Maint
U  #Maint
SPBN    END
L  0
T  #Starts

END: NOP 0

```

Création des blocs de données d'instance

Pour créer les trois blocs de données, choisissez respectivement la commande de menu **Insertion > Bloc**. Activez l'option "Bloc de données", entrez le numéro correspondant du bloc de données et sélectionnez le bloc fonctionnel FB 1 dans la liste déroulante. Vous venez ainsi de définir les blocs de données comme blocs de données d'instance avec affectation fixe au FB1.

A.4.5.5 Création de la fonction pour les soupapes

Tâches de la FC

La fonction pour les soupapes d'admission et d'alimentation ainsi que pour la soupape de vidange contient les fonctions logiques ci-après.

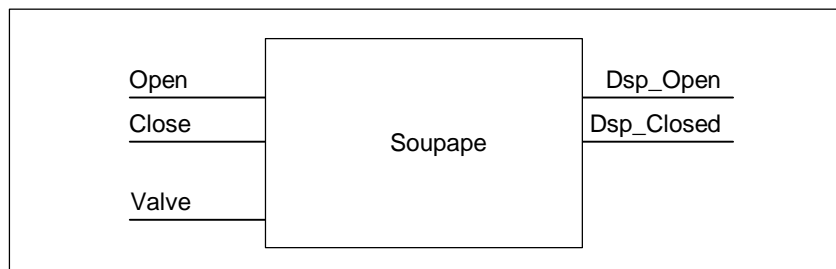
- Il existe une entrée d'ouverture et une entrée de fermeture des soupapes.
- Une série de verrouillages permet l'ouverture des soupapes. L'état des verrouillages est enregistré dans les données locales temporaires (pile L) de l'OB1 ("Enable_Valve") et combiné aux entrées d'ouverture et de fermeture lors de l'exécution de la FC pour les soupapes.

Le tableau ci-après présente les paramètres à transmettre à la fonction.

Paramètres pour les soupapes	Entrée	Sortie	Entrée/sortie
Open	✓		
Close	✓		
Dsp_Open		✓	
Dsp_Closed		✓	
Valve			✓

Identification des entrées et sorties

La figure ci-après montre les entrées et les sorties de la FC générique pour les soupapes. Les appareils appelant le FB pour le moteur transmettent les paramètres d'entrée et la FC pour les soupapes renvoie les paramètres de sortie.



Déclaration des variables de la FC pour les soupapes

Vous devez déclarer les paramètres d'entrée, de sortie et d'entrée/sortie pour la fonction commandant les soupapes comme vous l'avez fait pour le FB "Moteur" (voir la table de déclaration des variables ci-après).

Opérande	Déclaration	Nom	Type	Valeur initiale
0.0	in	Open	BOOL	FALSE
0.1	in	Close	BOOL	FALSE
2.0	out	Dsp_Open	BOOL	FALSE
2.1	out	Dsp_Closed	BOOL	FALSE
4.0	in_out	Valve	BOOL	FALSE

Pour les FC, les variables temporaires sont enregistrées dans la pile L. Les variables d'entrée, de sortie et d'entrée/sortie prennent la forme de pointeurs désignant le bloc de code ayant appelé la FC. Un espace mémoire supplémentaire est utilisé pour ces variables dans la pile L (après les variables temporaires).

Programmation de la FC pour les soupapes

Vous devez également écrire la fonction FC1 pour les soupapes avant l'OB1, car il faut créer le bloc appelé avant le bloc appelant.

La section des instructions de la FC1 se présente comme suit en langage de programmation LIST.

Réseau 1 Ouverture/fermeture et maintien

```

U(
O #Open
O #Valve
)
UN#Close
= #Valve

```

Réseau 2 , si soupape ouverte

```

U #Valve
= #Dsp_Open

```

Réseau 3 Signalisation, si soupape fermée

```

UN#Valve
= #Dsp_Closed

```

A.4.5.6 Création de l'OB1

L'OB1 détermine la structure de l'exemple de programme. Il contient, en outre, les paramètres transmis aux différents blocs fonctionnels et fonctions. Ainsi :

- Les réseaux LIST pour les pompes d'alimentation et le moteur mélangeur fournissent au FB pour le moteur les paramètres d'entrée pour le démarrage ("Start"), l'arrêt ("Stop"), pour le signal en retour ("Response") et pour la remise à zéro de l'indicateur de maintenance ("Reset_Maint"). Le FB pour le moteur s'exécute à chaque cycle de l'automate.
- Lorsque le FB pour le moteur s'exécute, les entrées "Timer_No" et "Response_Time" déterminent la temporisation à utiliser et la durée pendant laquelle un signal en retour doit être émis.
- La fonction pour les soupapes et le FB "Moteur" s'exécutent à chaque cycle de programme de l'automate, car ils sont appelés dans l'OB1.

Le programme utilise le FB "Moteur" avec différents DB d'instance afin d'accomplir les tâches requises pour la commande des pompes d'alimentation et du moteur mélangeur.

Déclaration de variables pour l'OB1

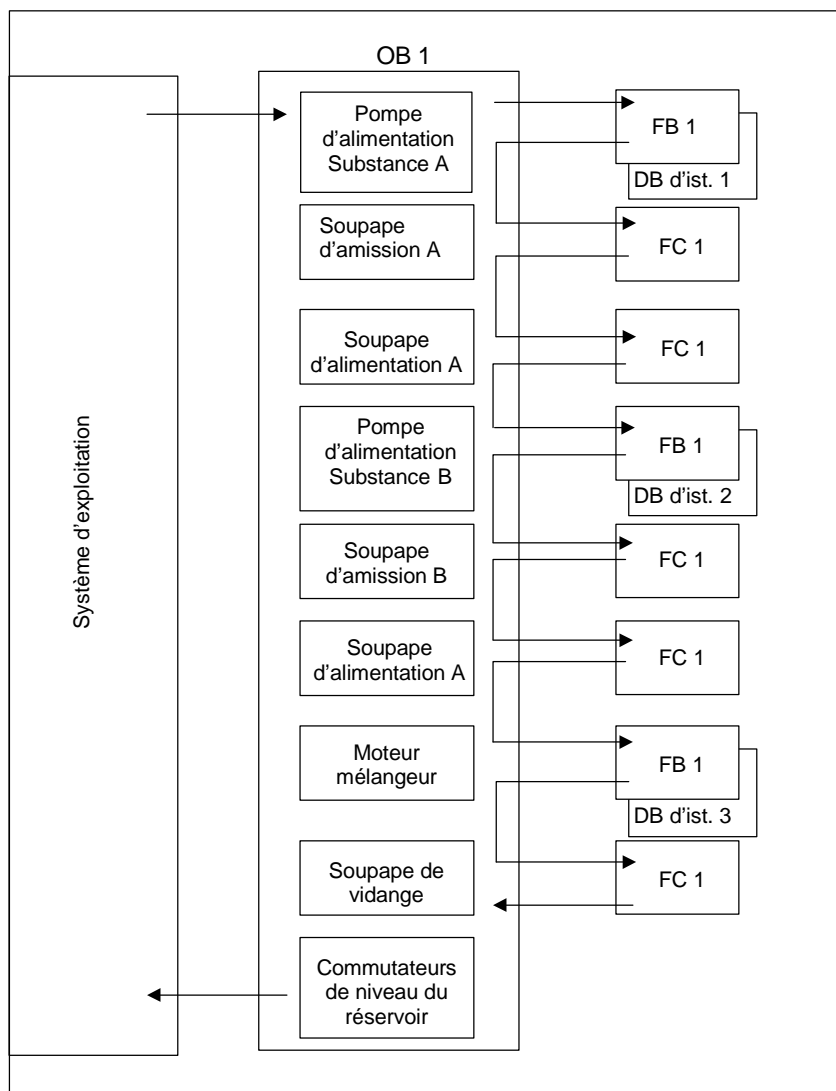
La table de déclaration des variables pour l'OB1 est représentée ci-après. Il ne faut pas modifier les vingt premiers octets qui contiennent les informations de déclenchement de l'OB1.

Opérande	Déclaration	Nom	Type
0.0	temp	OB1_EV_CLASS	BYTE
1.0	temp	OB1_SCAN1	BYTE
2.0	temp	OB1_PRIORITY	BYTE
3.0	temp	OB1_OB_NUMBR	BYTE
4.0	temp	OB1_RESERVED_1	BYTE
5.0	temp	OB1_RESERVED_2	BYTE
6.0	temp	OB1_PREV_CYCLE	INT
8.0	temp	OB1_MIN_CYCLE	INT
10.0	temp	OB1_MAX_CYCLE	INT
12.0	temp	OB1_DATE_TIME	DATE_AND_TIME
20.0	temp	Enable_Motor	BOOL
20.1	temp	Enable_Valve	BOOL
20.2	temp	Start_Fulfilled	BOOL
20.3	temp	Stop_Fulfilled	BOOL
20.4	temp	Inlet_Valve_A_Open	BOOL
20.5	temp	Inlet_Valve_A_Closed	BOOL
20.6	temp	Feed_Valve_A_Open	BOOL
20.7	temp	Feed_Valve_A_Closed	BOOL
21.0	temp	Inlet_Valve_B_Open	BOOL
21.1	temp	Inlet_Valve_B_Closed	BOOL
21.2	temp	Feed_Valve_B_Open	BOOL
21.3	temp	Feed_Valve_B_Closed	BOOL
21.4	temp	Open_Drain	BOOL
21.5	temp	Close_Drain	BOOL
21.6	temp	Close_Valve_Fulfilled	BOOL

Création du programme pour l'OB1

Dans STEP 7 Lite, il faut créer les blocs appelés par d'autres blocs avant les blocs contenant l'appel. Vous devez donc, dans notre exemple de programme, écrire le FB pour le moteur et la FC pour les soupapes avant le programme de l'OB1.

Les blocs FB1 et FC1 sont appelés à plusieurs reprises dans l'OB1, le FB1 avec différents DB d'instance :



La section des instructions de l'OB1 se présente comme suit en langage de programmation LIST.

Réseau 1 Verrouillages pour la pompe d'alimentation A

```
U "EMER_STOP_off"
U "Tank_below_max"
UN"Drain"
= #Enable_Motor
```

Réseau 2 Appel du FB Moteur pour substance A

```
U      "Feed_pump_A_start"
U      #Enable_Motor
=      #Start_Fulfilled
U(
O      "Feed_pump_A_stop"
ON#Enable_Motor
)
=      #Stop_Fulfilled
CALL   "Motor_block", "DB_feed_pump_A"
      Start :=#Start_Fulfilled
      Stop  :=#Stop_Fulfilled
      Response :="Flow_A"
      Reset_Maint :="Reset_maint"
      Timer_No :=T12
      Reponse_Time:=S5T#7S
      Fault :="Feed_pump_A_fault"
      Start_Dsp :="Feed_pump_A_on"
      Stop_Dsp :="Feed_pump_A_off"
      Maint :="Feed_pump_A_maint"
      Motor :="Feed_pump_A"
```

Réseau 3 Ajournement de la validation de soupape pour substance A

```
U "Feed_pump_A"
L S5T#1S
SE T 13
UN"Feed_pump_A"
R T 13
U T 13
= #Enable_Valve
```

Réseau 4 Commande de la soupape d'admission pour substance A

```
UN"Flow_A"
UN"Feed_pump_A"
=      #Close_Valve_Fulfilled
CALL   "Valve_block"
      Open :=#Enable_Valve
      Close :=#Close_Valve_Fulfilled
      Dsp_Open :=#Inlet_Valve_A_Open
      Dsp_Closed:=#Inlet_Valve_A_Closed
      Valve :="Inlet_Valve_A"
```


Réseau 5 Commande de la soupape d'alimentation pour substance A

```

UN"Flow_A"
UN"Feed_pump_A"
=          #Close_Valve_Fulfilled
CALL      "Valve_block"
    Open   :=#Enable_Valve
    Close  :=#Close_Valve_Fulfilled
    Dsp_Open   :=#Feed_Valve_A_Open
    Dsp_Closed:=#Feed_Valve_A_Closed
    Valve    :=#Feed_Valve_A

```

Réseau 6 Verrouillages pour la pompe d'alimentation B

```

U  "EMER_STOP_off"
U  "Tank_below_max"
UN"Drain"
=  #Enable_Motor

```

Réseau 7 Appel du FB Moteur pour substance B

```

U          "Feed_pump_B_start"
U          #Enable_Motor
=          #Start_Fulfilled
U(
O          "Feed_pump_B_stop"
ON#Enable_Motor
)
=          #Stop_Fulfilled
CALL      "Motor_block", "DB_feed_pump_B"
    Start   :=#Start_Fulfilled
    Stop    :=#Stop_Fulfilled
    Response :=#Flow_B
    Reset_Maint :=#Reset_maint
    Timer_No :=T14
    Reponse_Time:=S5T#7S
    Fault    :=#Feed_pump_B_fault
    Start_Dsp :=#Feed_pump_B_on
    Stop_Dsp  :=#Feed_pump_B_off
    Maint    :=#Feed_pump_B_maint
    Motor    :=#Feed_pump_B

```

Réseau 8 Ajournement de la validation de soupape pour substance B

```

U  "Feed_pump_B"
L  S5T#1S
SE T    15
UN"Feed_pump_B"
R  T    15
U  T    15
=  #Enable_Valve

```

Réseau 9 Commande de la soupape d'admission pour substance B

```

UN"Flow_B"
UN"Feed_pump_B"
=      #Close_Valve_Fulfilled
CALL   "Valve_block"
      Open  :=#Enable_Valve
      Close :=#Close_Valve_Fulfilled
      Dsp_Open  :=#Inlet_Valve_B_Open
      Dsp_Closed:=#Inlet_Valve_B_Closed
      Valve   :="Inlet_Valve_B"

```

Réseau 10 Commande de la soupape d'alimentation pour substance B

```

UN"Flow_B"
UN"Feed_pump_B"
=      #Close_Valve_Fulfilled
CALL   "Valve_block"
      Open  :=#Enable_Valve
      Close :=#Close_Valve_Fulfilled
      Dsp_Open  :=#Feed_Valve_B_Open
      Dsp_Closed:=#Feed_Valve_B_Closed
      Valve   :="Feed_Valve_B"

```

Réseau 11 Verrouillages pour le moteur mélangeur

```

U  "EMER_STOP_off"
U  "Tank_above_min"
UN"Drain"
=  #Enable_Motor

```

Réseau 12 Appel du FB Moteur pour moteur mélangeur

```

U      "Agitator_start"
U      #Enable_Motor
=      #Start_Fulfilled
U(
O      "Agitator_stop"
ON#Enable_Motor
)
=      #Stop_Fulfilled
CALL   "Motor_block", "DB_agitator"
      Start  :=#Start_Fulfilled
      Stop   :=#Stop_Fulfilled
      Response :="Agitator_running"
      Reset_Maint :="Reset_maint"
      Timer_No  :=T16
      Reponse_Time:=S5T#10S
      Fault    :="Agitator_fault"
      Start_Dsp :="Agitator_on"
      Stop_Dsp  :="Agitator_off"
      Maint     :="Agitator_maint"
      Motor     :="Agitator"

```

Réseau 13 Verrouillages pour la soupape de vidange

```

U  "EMER_STOP_off"
U  "Tank_not_empty"
UN"Agitator"
=  "Enable_Valve"

```

Réseau 14 Commande de la soupape de vidange

```
U      "Drain_open"
U      #Enable_Valve
=      #Open_Drain
U(
O      "Drain_closed"
ON#Enable_Valve
)
=      #Close_Drain
CALL   "Valve_block"
  Open  :=#Open_Drain
  Close :=#Close_Drain
  Dsp_Open    :="Drain_open_disp"
  Dsp_Closed  :="Drain_closed_disp"
  Valve  :="Drain"
```

Réseau 15 Indication du niveau du réservoir

```
UN"Tank_below_max"
=  "Tank_max_disp"
UN"Tank_above_min"
=  "Tank_min_disp"
UN"Tank_not_empty"
=  "Tank_empty_disp"
```

A.4.6 Exemple d'utilisation d'alarmes horaires

A.4.6.1 Structure du programme utilisateur 'Alarme horaire'

Problème posé

La sortie A 4.0 doit être mise à 1 du lundi 5.00 heures au vendredi 20.00 heures. Du vendredi 20.00 heures au lundi 5.00 heures, la sortie A 4.0 doit être remise à 0.

Transcription dans le programme utilisateur

Le tableau suivant montre les tâches partielles des blocs utilisés.

Bloc	Tâche partielle
OB1	Appel de la fonction FC12
FC12	Selon l'état de la sortie A 4.0, de l'état de l'alarme horaire et des entrées E 0.0 et E 0.1 <ul style="list-style-type: none">• Prédéfinir instant de déclenchement• Mettre alarme horaire à 1• Activer alarme horaire• CAN_TINT
OB10	Selon le jour de la semaine en cours <ul style="list-style-type: none">• Prédéfinir instant de déclenchement• Mettre à 1 ou remettre à 0 la sortie A 4.0• Mettre à 1 l'alarme horaire suivante• Activer l'alarme horaire suivante
OB80	Mises à 1 de la sortie A 4.1 Enregistrer informations de l'événement déclencheur de l'OB80 dans la zone des mémentos

Opérandes utilisés

Le tableau suivant montre les opérandes globaux utilisés. Les variables temporaires des blocs sont déclarées dans la section de déclaration du bloc respectif.

Opérande	Signification
E 0.0	Entrée de validation de "Mettre alarme horaire à 1" et "Activer alarme horaire"
E 0.1	Entrée d'annulation d'une alarme horaire
A 4.0	Sortie mise à 1/remise à 0 par l'OB d'alarme horaire (OB10)
A 4.1	Sortie mise à 1 en cas d'erreur de temps (OB80)
MW 16	ETAT de l'alarme horaire (SFC31 "QRY_TINT")
MB 100 à MB 107	Mémoire pour informations de l'événement déclencheur de l'OB10 (uniquement horodatage)
MB 110 à MB 129	Mémoire pour informations de l'événement déclencheur de l'OB80 (erreur de temps)
MW 200	RET_VAL de la SFC28 "SET_TINT"
MB 202	Mémoire intermédiaire des résultats binaires (bit d'état RB) pour les SFC
MW 204	RET_VAL de la SFC30 "ACT_TINT"
MW 208	RET_VAL de la SFC31 "QRY_TINT"

SFC et FC utilisés

Les fonctions système suivantes sont utilisées dans l'exemple de programme :

- SFC28 "SET_TINT" Réglage de l'alarme horaire
- SFC29 "CAN_TINT" Annulation de l'alarme horaire
- SFC30 "ACT_TINT" Activation de l'alarme horaire
- SFC31 "QRY_TINT" Interrogation de l'alarme horaire
- FC3 "D_TOD_DT" Regroupement de DATE et TIME_OF_DAY en DT

A.4.6.2 FC12

Section de déclaration

Les variables temporaires de blocs suivantes sont déclarées dans la section de déclaration de la FC12 :

Nom de la variable	Type de données	Déclaration	Commentaire
IN_HEURE	TIME_OF_DAY	temp	Prédéfinition de l'heure de déclenchement
IN_DATE	DATE	temp	Prédéfinition de la date de déclenchement
OUT_HEURE_DATE	DATE_AND_TIME	temp	Date/heure de déclenchement converties
MEMENTO_OK	BOOL	temp	Validation pour le réglage de l'alarme horaire

Section des instructions en LIST

Dans la section des instructions de la FC12, vous entrez le programme utilisateur LIST suivant :

LIST (FC 12)	Signification
Réseau 1 : CALL SFC 31 NR_OB := 10 VAL_RET:= MW 208 ETAT := MW 16	//SFC QRY_TINT //Interrogation de l'ETAT de l'alarme //horaire.
Réseau 2 : UN A 4.0 SPB mont L D#1995-1-27 T #IN_DATE L TOD#20:0:0.0 T #IN_HEURE SPA conv mont: L D#1995-1-23 T #IN_DATE L TOD#5:0:0.0 T #IN_HEURE conv: NOP 0	//Prédéfinir instant de déclenchement en //fonction de A 4.0 (dans la variable //#IN_DATE et #IN_HEURE) //La date de déclenchement est un //vendredi. //La date de déclenchement est un lundi.

LIST (FC 12)	Signification
Réseau 3 : CALL FC 3 IN1 := #IN_DATE IN2 := #IN_HEURE VAL_RET := #OUT_HEURE_DATE	//Convertir format de DATE et //TIME_OF_DAY en DATE_AND_TIME (pour //régler l'alarme horaire)
Réseau 4 : U E 0.0 UN M 17.2 U M 17.4 = #MEMENTO_OK	//Toutes conditions pour régler alarme //horaire remplies ? (entrée validation //mise à 1, alarme horaire non active et //OB d'alarme horaire chargé)
Réseau 5 : U #MEMENTO_OK SPBNB m001 CALL SFC 28 NR_OB := 10 SDT := #OUT_HEURE_DATE PERIODE := W#16#1201 VAL_RET := MW 200	//Si oui, régler l'alarme horaire...
m001 U RB = M 202.3	//...et activer l'alarme horaire.
Réseau 6 : U #MEMENTO_OK SPBNB m002 CALL SFC 30 NR_OB := 10 VAL_RET := MW 204	
m002 U RB = M 202.4	//Si l'entrée pour annuler l'alarme //horaire est mise à 1, alors annuler //l'alarme horaire.
Réseau 7 : U E 0.1 SPBNB m003 CALL SFC 29 NR_OB := 10 RET_VAL := MW 210	
m003 U RB = M 202.5	

Blocs fonctionnels système (SFB) et fonctions système (SFC)

A.4.6.3 OB10

Section de déclaration

Selon la section de déclaration prédéfinie pour l'OB10, les variables temporaires de bloc suivantes sont déclarées :

- structure pour l'ensemble des informations de l'événement déclencheur (STARTINFO),
- dans la structure STARTINFO, une structure pour l'heure (T_STMP),
- autres variables temporaire de bloc JOURSEM, IN_DATE, IN_HEURE et OUT_HEURE_DATE.

Nom de la variable	Type de données	Déclaration	Commentaire
STARTINFO	STRUCT	temp	Ensemble des informations de l'événement déclencheur de l'OB10 déclaré comme structure
ID_E	WORD	temp	ID d'événement
CLASSE_PR	BYTE	temp	Classe de priorité
NR_OB	BYTE	temp	Numéro de l'OB
RESERVED_1	BYTE	temp	Réservé
RESERVED_2	BYTE	temp	Réservé
PERIODE	WORD	temp	Périodicité de l'alarme horaire
RESERVED_3	DWORD	temp	Réservé
T_STMP	STRUCT	temp	Structure pour les indications d'horodatage
ANNEE	BYTE	temp	
MOIS	BYTE	temp	
JOUR	BYTE	temp	
HEURES	BYTE	temp	
MINUTES	BYTE	temp	
SECONDES	BYTE	temp	
MSEC_JOURSEM	WORD	temp	
	END_STRUCT	temp	
	END_STRUCT	temp	
JOURSEM	INT	temp	Jour de la semaine
IN_DATE	DATE	temp	Variable d'entrée pour FC3 (conversion du format horaire)
IN_HEURE	TIME_OF_DAY	temp	Variable d'entrée pour FC3 (conversion du format horaire)
OUT_HEURE_DATE	DATE_AND_TIME	temp	Variable de sortie pour FC3 et variable d'entrée pour SFC28

Section des instructions en LIST

Dans la section des instructions de l'OB10, vous entrez le programme utilisateur LIST suivant :

LIST (OB 10)	Signification
Réseau 1 :	
L #STARTINFO.T_STMP.MSEC_JOURSEM	//Sélectionner jour de la semaine
L W#16#F	
UW	
T #JOURSEM	//et mémoriser.
Réseau 2 :	
L #JOURSEM	//Si jour de la semaine n'est pas lundi,
L 2	//prédéfinir lundi 5.00 h comme prochain
<>I	//instant de déclenchement et remettre
SPB mont	//la sortie A 4.0 à zéro.
Réseau 3 :	
L D#1995-1-27	
T #IN_DATE	//Sinon, c.-à-d. si jour de la semaine =
L TOD#20:0:0.0	//lundi, prédéfinir vendredi 20.00 h
T #IN_HEURE	//comme prochain instant de déclench.
SET	//et mettre la sortie A 4.0 à 1.
= A 4.0	
SPA conv	
mont: L D#1995-1-23	
T #IN_DATE	
L TOD#5:0:0.0	
T #IN_HEURE	
CLR	
= A 4.0	
conv: NOP 0	//Prédéfinition de l'instant de
Réseau 4 :	//déclenchement terminée.
CALL FC 3	//Convertir l'instant de déclenchement
IN1 := #IN_DATE	//prédéfini dans le format DATE_AND_TIME
IN2 := #IN_HEURE	//(pour SFC28).
VAL_RET := #OUT_HEURE_DATE	
Réseau 5 :	
CALL SFC 28	//Régler l'alarme horaire.
NR_OB := 10	
SDT := #OUT_HEURE_DATE	
PERIODE := W#16#1201	
VAL_RET := MW 200	
U RB	
= M 202.1	
Réseau 6 :	
CALL SFC 30	//Activer alarme horaire
NR_OB := 10	
VAL_RET := MW 204	
U RB	
= M 202.2	
Réseau 7 :	
CALL SFC 20	//Transfert de bloc : enregistrer
SRCBLK := #STARTINFO.T_STMP	//indication horaire des informations de
VAL_RET := MW 206	//l'événement déclencheur de l'OB10 //dans
DSTBLK := P#M 100.0 OCTET 8	zone de mémentos MB 100 à MB 107.

A.4.6.4 OB1 et OB80

Puisque les informations de l'événement déclencheur de l'OB1 (OB pour le programme cyclique) ne sont pas exploitées dans cet exemple, seules celles de l'événement déclencheur de l'OB80 sont représentées.

Section des instructions de l'OB1

Dans la section des instructions de l'OB1, vous entrez le programme utilisateur LIST suivant :

LIST (OB 1)	Signification
CALL FC 12	//Appel de la fonction FC12

Section de déclaration de l'OB80

Selon la section de déclaration prédéfinie de l'OB80, les variables temporaires de bloc suivantes sont déclarées :

- structure pour l'ensemble des informations de l'événement déclencheur (STARTINFO),
- dans la structure STARTINFO, une autre structure pour l'heure (T_STMP).

Nom de la variable	Type de données	Déclaration	Commentaire
STARTINFO	STRUCT	temp	Ensemble des informations de l'événement déclencheur de l'OB80 déclaré comme structure
ID_E	WORD	temp	ID d'événement
CLASSE_PR	BYTE	temp	Classe de priorité
NR_OB	BYTE	temp	Numéro de l'OB
RESERVED_1	BYTE	temp	Réservé
RESERVED_2	BYTE	temp	Réservé
INFO_S1	WORD	temp	Information supplémentaire sur l'événement ayant occasionné l'erreur.
INFO_S2	DWORD	temp	Information supplémentaire sur l'ID d'événement, la classe de priorité et le numéro d'OB de l'événement d'erreur
T_STMP	STRUCT	temp	Structure pour les indications d'horodatage
ANNEE	BYTE	temp	
MOIS	BYTE	temp	
JOUR	BYTE	temp	
HEURES	BYTE	temp	
MINUTES	BYTE	temp	
SECONDES	BYTE	temp	
MSEC_JOURSEM	WORD	temp	
	END_STRUCT	temp	
	END_STRUCT	temp	

Section des instructions de l'OB80

Dans la section des instructions de l'OB80, appelé par le système d'exploitation en cas d'erreur de temps, vous entrez le programme utilisateur LIST suivant :

LIST (OB 80)	Signification
Réseau 1 :	
UN A 4.1	//Mettre la sortie A 4.1 à 1 lorsque
S A 4.1	//l'erreur de temps est survenue.
CALL SFC 20	//Transfert de bloc : enregistrer
SRCBLK := #STARTINFO	//l'ensemble de l'événement de
VAL_RET := MW 210	//déclenchement dans la zone de mémentos
DSTBLK := P#M 110.0 octet 20	//MB 110 à MB 129.

A.4.7 Exemple d'utilisation d'alarmes temporisées

A.4.7.1 Structure du programme utilisateur 'Alarme temporisée'

Problème posé

Lorsque l'entrée E 0.0 est mise à 1, la sortie A 4.0 doit être mise à 1, et ceci 10 secondes plus tard. Chaque mise à 1 de l'entrée E 0.0 doit déclencher une nouvelle fois le temps de retard.

Comme identificateur spécifique à l'utilisateur, l'instant (secondes et millisecondes) de déclenchement de l'alarme temporisée doit apparaître dans les informations de l'événement déclencheur de l'OB d'alarme temporisée (OB20).

Si E 0.1 est mise à 1 durant ces 10 secondes, le bloc d'organisation OB20 ne doit pas être appelé, c'est-à-dire la sortie A 4.0 ne doit pas être mise à 1.

Lorsque l'entrée E 0.2 est mise à 1, la sortie A 4.0 doit être remise à 0.

Transcription dans le programme utilisateur

Le tableau suivant montre les tâches partielles des blocs utilisés.

Bloc	Tâche partielle
OB1	Lecture et préparation de l'heure actuelle pour le déclenchement de l'alarme temporisée Déclenchement de l'alarme temporisée en fonction du changement de front à l'entrée E 0.0 Annulation de l'alarme temporisée en fonction de l'état de l'alarme temporisée et du changement de front à l'entrée E 0.1 Remise à 0 de la sortie A 4.0 en fonction de l'état de l'entrée E 0.2
OB20	Mise à 1 de la sortie A 4.0 Lecture et préparation de l'heure actuelle Enregistrement des informations de l'événement déclencheur dans la zone des mémentos

Opérands utilisés

Le tableau ci-après montre les tables de données globales utilisées. Les variables temporaires des blocs sont déclarées dans la section de déclaration du bloc respectif.

Opérande	Signification
E 0.0	Entrée de validation de "Déclencher l'alarme de temporisation"
E 0.1	Entrée d'annulation d'une alarme temporisée
E 0.2	Entrée de remise à 0 de la sortie A 4.0
A 4.0	Sortie mise à 1 par l'OB d'alarme temporisée (OB20)
MB 1	Utilisé pour les mementos de front et la mémoire intermédiaire des résultats binaires (bit d'état RB) pour les SFC
MW 4	ETAT de l'alarme temporisée (SFC34 "QRY_TINT")
MD 10	Secondes et millisecondes en format DCB reprises dans les informations de l'événement déclencheur de l'OB1
MW 100	VAL_RET de la SFC32 "SRT_DINT"
MW 102	VAL_RET de la SFC34 "QRY_DINT"
MW 104	VAL_RET de la SFC33 "CAN_DINT"
MW 106	VAL_RET de la SFC20 "BLKMOV"
MB 120 à MB 139	Mémoire pour informations de l'événement déclencheur de l'OB20
MD 140	Secondes et millisecondes en format DCB reprises dans informations de l'événement déclencheur de l'OB20
MW 144	Secondes et millisecondes en format DCB reprises dans informations de l'événement déclencheur de l'OB1 ; repris dans informations de l'événement déclencheur de l'OB20 (identificateur spécifique à l'utilisateur SIGN)

SFC utilisées

Les fonctions système suivantes sont utilisées dans le programme utilisateur "Alarmes temporisées" :

- SFC32 "SRT_DINT" Déclenchement de l'alarme temporisée
- SFC33 "CAN_DINT" Annulation de l'alarme temporisée
- SFC34 "QRY_DINT" Interrogation de l'état d'une alarme temporisée

A.4.7.2 OB20

Section de déclaration

En fonction de la section de déclaration prédéfinie de l'OB20, les variables temporaires de bloc suivantes sont déclarées :

- structure pour l'ensemble des informations de l'événement déclencheur (STARTINFO),
- dans la structure STARTINFO, une structure pour l'horodatage (T_STMP).

Nom de la variable	Type de données	Déclaration	Commentaire
STARTINFO	STRUCT	temp	Informations de déclenchement de l'OB20
ID_E	WORD	temp	ID d'événement
NR_NIVEX	BYTE	temp	Niveau d'exécution
NR_OB	BYTE	temp	Numéro d'OB
IDD1	BYTE	temp	Identification de données 1
IDD2	BYTE	temp	Identification de données 2
SIGN	WORD	temp	Identification spécifique à l'utilisateur
DTIME	TIME	temp	Heure de déclenchement de l'alarme temporisée
T_STMP	STRUCT	temp	Structure pour les indications d'horodatage (horodateur)
ANNEE	BYTE	temp	
MOIS	BYTE	temp	
JOUR	BYTE	temp	
HEURES	BYTE	temp	
MINUTES	BYTE	temp	
SECONDES	BYTE	temp	
MSEC_JOURSEM	WORD	temp	
	END_STRUCT	temp	
	END_STRUCT	temp	

Section des instructions

Dans la section des instructions de l'OB20, vous entrez le programme utilisateur LIST suivant :

LIST (OB 20)	Signification
Réseau 1 :	
SET	//Mise à 1 impérative de la sortie A 4.0
= A 4.0	
Réseau 2 :	
L AW 4	//Actualisation immédiate du mot de sortie
T PAW 4	
Réseau 3 :	
L #STARTINFO.T_STMP.SECONDES	//Lecture des secondes dans informations
T MW 140	//de l'événement déclencheur
L #STARTINFO.T_STMP.MSEC_JOURSEM	//Lecture millisecondes et jour semaine
T MW 142	//dans infos événement déclencheur
L MD 140	
SRD 4	//Elimination du jour de la semaine et
T MD 140	//réinscription des millisecondes (sont
	//à présent en format DCB dans MW 142).
Réseau 4 :	
L #STARTINFO.SIGN	//Lecture instant de déclenchement
T MW 144	//alarme temporisée (=appel de la SFC32)
	//dans infos événement déclencheur
Réseau 5 :	
CALL SFC 20	//Copie des informations de l'événement
SRCBLK := STARTINFO	//déclencheur dans la zone de mémentos
VAL_RET := MW 106	//(MB 120 à MB 139)
DSTBLK := P#M 120.0 OCTET 20	

A.4.7.3 OB1

Section de déclaration

En fonction de la section de déclaration prédéfinie de l'OB1, les variables temporaires de bloc suivantes sont déclarées :

- structure pour l'ensemble des informations de l'événement déclencheur (STARTINFO),
- dans la structure STARTINFO, une structure pour l'horodatage (T_STMP).

Nom de la variable	Type de données	Déclaration	Commentaire
STARTINFO	STRUCT	temp	Informations de déclenchement de l'OB1
ID_E	WORD	temp	ID d'événement
NR_NIVEX	BYTE	temp	Niveau d'exécution
NR_OB	BYTE	temp	Numéro d'OB
IDD 1	BYTE	temp	Identification de données 1
IDD 2	BYTE	temp	Identification de données 2
CYC_ACT	INT	temp	Temps de cycle en cours
CYC_MIN	INT	temp	Temps de cycle minimum
CYC_MAX	INT	temp	Temps de cycle maximal
T_STMP	STRUCT	temp	Structure pour les indications d'horodatage (horodatage)
ANNEE	BYTE	temp	
MOIS	BYTE	temp	
JOUR	BYTE	temp	
HEURE	BYTE	temp	
MINUTES	BYTE	temp	
SECONDES	BYTE	temp	
MSEC_JOURSEM	WORD	temp	
	END_STRUCT	temp	
	END_STRUCT	tempv	

Section des instructions

Dans la section des instructions de l'OB1, vous entrez le programme utilisateur LIST suivant :

LIST (OB 1)	Signification
Réseau 1 :	
L #STARTINFO.T_STMP.SECONDES	//Lecture secondes dans informations //de
T MW 10	1'événement déclencheur
L #STARTINFO.T_STMP.MSEC_JOURSEM	//Lecture millisecondes et jour semaine
T MW 12	//dans infos événement déclencheur
L MD 10	//Elimination du jour de la semaine et
SRD 4	//réinscription des millisecondes (sont
T MD 10	//à présent en format DCB dans le MW 12)
Réseau 2 :	//Front positif à l'entrée E 0.0 ?
U E 0.0	
FP M 1.0	
= M 1.1	
Réseau 3 :	//Si oui, déclenchement de l'alarme
U M 1.1	//temporisée (instant de déclenchement
SPBNB m001	//de l'alarme temporisée affecté au
CALL SFC 32	//paramètre SIGN)
NR_OB := 20	
DTME := T#10S	
SIGN := MW 12	
VAL_RET:= MW 100	
m001: NOP 0	
Réseau 4 :	//Interrogation de l'état de l'alarme
CALL SFC 34	//temporisée (SFC QRY_DINT)
NR_OB := 20	
VAL_RET:= MW 102	
ETAT := MW 4	
Réseau 5 :	//Front positif à l'entrée E 0.1 ?
U E 0.1	
FP M 1.3	
= M 1.4	
Réseau 6 :	//... et alarme temporisée activée ?
U M 1.4	//(bit 2 de l'ETAT de l'alarme
U M 5.2	//temporisée)
SPBNB m002	//Alors annulation de l'alarme
CALL SFC 33	//temporisée
NR_OB := 20	
VAL_RET:= MW 104	
m002: NOP 0	//Remise à 0 de la sortie A 4.0 avec
U E 0.2	//l'entrée E 0.2
R A 4.0	

A.5 Accès aux zones de données du processus et de périphérie

Ce chapitre décrit l'adressage des zones de données de périphérie (données utiles, données de diagnostic et de paramètres).

De plus amples informations sur les fonctions système citées dans le présent paragraphe sont données dans le manuel de référence *Logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système*.

A.5.1 Accès à la zone de données du processus

La CPU peut accéder aux entrées et sorties des modules d'entrées/sorties TOR centralisés ou décentralisés soit indirectement via la mémoire image du processus, soit directement via le bus interne/de fond de panier ou P.

Quant aux entrées et sorties des modules d'entrées/sorties analogiques centralisés et décentralisés, la CPU y accède directement via le bus interne/de fond de panier ou P.

Adressage des modules

L'association entre les adresses utilisées dans le programme utilisateur et les modules se fait par configuration des modules avec STEP 7 Lite :

- pour la périphérie centralisée : disposition du profilé support ou châssis et affectation des modules aux emplacement dans la table de configuration ;
- pour la périphérie décentralisée (PROFIBUS DP) : disposition des esclaves de périphérie décentralisée (DP) dans la table de configuration "Réseau maître" avec indication de l'adresse PROFIBUS et affectation des modules aux emplacements.

La configuration des modules remplace le réglage d'adresses des différents modules par commutateurs. La CPU reçoit de la PG des données comme résultat de la configuration, données grâce auxquelles elle reconnaît les modules affectés.

Adressage de la périphérie

Il existe une zone d'adresses propre pour les entrées et pour les sorties. Aussi, l'adresse d'une zone de périphérie doit-elle contenir l'identification E (pour les entrées) et A (pour les sorties) en plus de l'indication d'octet ou de mot.

Le tableau suivant présente les zones d'adresses de périphérie disponibles.

Plage d'opérandes	Accès par des unités de taille suivante	Notation S7
Zone de périphérie des entrées	Octet d'entrée de périphérie Mot d'entrée de périphérie Double mot d'entrée de périphérie	PEB PEW PED
Zone de périphérie des sorties	Octet de sortie de périphérie Mot de sortie de périphérie Double mot de sortie de périphérie	PAB PAW PAD

Reportez-vous aux manuels suivants pour savoir quelles zones d'adresses sont possibles pour les différents modules :

- manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*,
- manuel de référence *Systèmes d'automatisation S7-300, M7-300 - Caractéristiques des modules*.

Adresse de début de module

L'adresse de début de module est l'adresse d'octet la plus basse d'un module. Elle représente l'adresse de début de la zone des données utiles du module et est souvent utilisée pour désigner le module entier.

Elle est, par exemple, inscrite dans les informations de déclenchement des blocs d'organisation associés à des alarmes de processus, de diagnostic, de débrogage/enfichage et à des erreurs d'alimentation, et identifie ainsi le module à l'origine de l'alarme.

A.5.2 Accès à la zone de données de périphérie

La zone de données de périphérie se décompose :

- en données utiles
- et en données de diagnostic et de paramètres.

Ces deux parties comportent une zone d'entrée (accès en lecture uniquement) et une zone de sortie (accès en écriture uniquement).

Données utiles

On accède aux données utiles via l'adresse d'octet (pour les modules TOR) ou via l'adresse de mot (pour les modules analogiques) de la zone d'entrée ou de sortie. Vous pouvez accéder à ces données utiles à l'aide de commandes de chargement et de transfert ou de fonctions de communication ou au moyen du transfert de mémoire image. Les données utiles peuvent être, par exemple, des signaux d'entrée et de sortie analogiques et TOR de modules de signaux,

Lors de la transmission de données utiles, il est possible d'atteindre une cohérence des données de quatre octets au maximum. Si vous utilisez l'instruction "Transférer double mot", 4 octets sont transmis en un bloc et sans modification. En revanche, si vous vous servez de quatre instructions "Transférer octet d'entrée", il se pourrait que soit déclenché à une limite d'instruction un OB d'alarme de processus qui transmette des données à la même adresse et modifie ainsi le contenu des quatre octets d'origine.

Données de diagnostic et de paramètres

Il n'est pas possible d'accéder individuellement aux données de diagnostic et de paramètres d'un module, mais uniquement sous la forme d'enregistrements entiers. En principe, les données de diagnostic et de paramètres sont transmises de manière cohérente.

On accède aux données de diagnostic et de paramètres via l'adresse de début du module concerné et le numéro d'enregistrement. Les enregistrements sont subdivisés en enregistrements d'entrée et de sortie, les enregistrements d'entrée pouvant uniquement être lus et les enregistrements de sortie uniquement être écrits. Vous pouvez accéder aux enregistrements à l'aide de fonctions système ou de fonctions de communication (contrôle-commande). Le tableau suivant montre l'affectation des enregistrements aux données de diagnostic et de paramètres.

Données	Description
Données de diagnostic	Pour les modules capables de diagnostic, vous recevez lors de la lecture des enregistrements 0 et 1 les données de diagnostic de ce module.
Données de paramètres	Pour les modules paramétrables, vous transférez lors de l'écriture des enregistrements 0 et 1 les paramètres de ce module.

Accès aux enregistrements

Vous pouvez utiliser les informations contenues dans les enregistrements d'un module pour modifier le paramétrage de modules paramétrables et pour lire les informations de diagnostic des modules aptes au diagnostic.

Le tableau suivant présente les fonctions système permettant d'accéder aux enregistrements.

SFC	Application
Paramétrage de modules	
SFC55 WR_PARM	Transfert des paramètres modifiables (enregistrement 1) au module de signaux adressé
SFC56 WR_DPARM	Transfert des paramètres des SDB 100 à 129 au module de signaux adressé
SFC57 PARM_MOD	Transfert de tous les paramètres des SDB 100 à 129 au module de signaux adressé
SFC58 WR_REC	Transfert d'un enregistrement quelconque au module de signaux adressé
Lecture d'informations de diagnostic	
SFC59 RD_REC	Lecture des données de diagnostic

A.6 Définition du comportement en fonctionnement

A.6.1 Définition du comportement en fonctionnement

Cette rubrique explique comment vous pouvez influencer, à l'aide des paramètres système ou de fonctions système, sur les propriétés des automates programmables S7 qui ne sont pas définitivement fixées.

Vous trouverez des informations détaillées sur les paramètres des modules dans les manuels relatifs aux différentes familles de systèmes d'automatisation, tels que :

- le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*,
- le manuel de référence *Systèmes d'automatisation S7-300, M7-300 - Caractéristiques des modules*.

Vous trouverez toutes les informations sur les fonctions système dans le manuel de référence *Logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système* ou dans l'aide en ligne sur la bibliothèque *System Function Blocks*.

A.6.2 Modification du comportement et des propriétés des modules

Paramètres par défaut

À la livraison, tous les modules paramétrables du système d'automates programmables S7 sont réglés à des valeurs par défaut qui conviennent à des applications standard. Ces valeurs par défaut vous permettent d'utiliser directement les modules sans devoir effectuer d'autres réglages. Elles sont présentées dans les descriptions de modules des manuels respectifs. Elles sont aussi prises en considération dans les dialogues de paramétrage de STEP 7 Lite.

Modules paramétrables

Vous pouvez toutefois bien sûr paramétrer le comportement et les propriétés des modules et, ainsi, les adapter aux exigences et aux caractéristiques de votre installation. Les CPU, FM, CP ainsi que certains modules d'entrées et de sorties analogiques et modules d'entrées TOR sont des modules paramétrables.

Il existe des modules paramétrables avec ou sans sauvegarde.

Après chaque coupure de courant, vous devez transmettre à nouveau les données de paramétrage aux modules sans sauvegarde. Les paramètres de ces modules sont enregistrés dans la zone de mémoire rémanente de la CPU (paramétrage indirect par la CPU).

Définition et chargement des paramètres

Vous définissez les paramètres des modules à l'aide de STEP 7 Lite. Lors de l'enregistrement de la configuration matérielle contenant également des paramètres, STEP 7 Lite crée des "Blocs de données système" qui sont chargés dans la CPU avec le programme utilisateur et qui, de là, sont transférés dans les modules concernés lors de la mise en route. Comme les blocs de données système représentent la configuration matérielle, ils sont visualisés dans la vue du projet comme "Matériel".

Paramétrages possibles

Exemples de propriétés de CPU paramétrables :

- comportement à la mise en route,
- cycle,
- MPI
- diagnostic,
- rémanence,
- mémentos de cadence,
- traitement d'alarmes,
- périphérie interne,
- niveau de protection,
- horloge temps réel,
- erreurs asynchrones.

Paramétrage à l'aide de SFC

A côté du paramétrage avec STEP 7 Lite, il y a aussi la possibilité de modifier les paramètres des modules depuis le programme à l'aide de fonctions système. Le tableau suivant indique les SFC qui peuvent servir à cet usage.

SFC	Application
SFC55 WR_PARM	Transfert des paramètres modifiables (enregistrement 1) au module de signaux adressé
SFC56 WR_DPARM	Transfert des paramètres des SDB associés au module de signaux adressé
SFC57 PARM_MOD	Transfert de tous les paramètres des SDB associés au module de signaux adressé
SFC58 WR_REC	Transfert d'un enregistrement quelconque au module de signaux adressé

Vous trouverez des informations détaillées sur les fonctions système dans le manuel de référence *Logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système*.

Pour savoir quels paramètres de module vous pouvez modifier dynamiquement, consultez les manuels des modules respectifs, tels que :

- le manuel *Système d'automatisation S7-300, Installation et configuration - Caractéristiques des CPU*,
- le manuel de référence *Systèmes d'automatisation S7-300, M7-300 - Caractéristiques des modules*.

A.6.3 Avantage des fonctions d'horodatage

Toutes les CPU possèdent une horloge (horloge temps réel ou horloge logicielle). Dans l'automate programmable, l'horloge peut aussi bien fonctionner comme horloge maître que comme horloge esclave à synchronisation externe. Elle permet l'utilisation d'alarmes horaires et de compteurs d'heures de fonctionnement.

Format horaire

L'horloge indique toujours l'heure (résolution minimale 1 s) et la date avec le jour de la semaine. Certaines CPU permettent aussi l'affichage de millisecondes (voir la documentation respective sur les caractéristiques techniques des CPU).

Réglage et lecture de l'heure de l'horloge CPU

Vous réglez l'heure et la date

- depuis le programme utilisateur en appelant la SFC0 SET_CLK
- ou depuis la PG/le PC au moyen du panneau de commande CPU élargi, ce qui démarre l'horloge.

Vous lisez l'heure et la date en cours

- depuis le programme utilisateur avec la SFC1 READ_CLK
- ou depuis la PG/le PC au moyen du panneau de commande CPU élargi.

Nota : pour éviter des indications différentes dans les systèmes HMI, il est conseillé de régler la CPU sur l'**heure d'hiver** !

Paramétrage de l'horloge

Lorsqu'un réseau comporte plus d'un module avec horloge, vous devez paramétrer dans STEP 7 Lite quelle CPU doit fonctionner comme maître et quelle CPU doit fonctionner comme esclave pour la synchronisation de l'heure. Le paramétrage vous permet également de définir si la synchronisation doit être réalisée via le bus de communication ou via l'interface MPI et à quels intervalles.

Synchronisation de l'heure

Afin de garantir que tous les modules du réseau sont réglés à la même heure, le programme système synchronise les horloges esclaves à des intervalles réguliers (paramétrables). La fonction système SFC48 SNC_RTCB vous permet de transférer la date et l'heure de l'horloge maître aux horloges esclaves.

Mise en œuvre d'un compteur d'heures de fonctionnement

Un compteur d'heures de fonctionnement permet de compter la durée d'activation d'un élément du système connecté ou la durée de fonctionnement de la CPU sous forme de total du nombre d'heures de fonctionnement.

Le compteur d'heures de fonctionnement est stoppé lorsque la CPU est à l'état d'arrêt. Sa valeur est conservée après un effacement général. Lors d'un démarrage à chaud, le compteur d'heures de fonctionnement doit être redémarré par le programme utilisateur ; lors d'un redémarrage, il continue automatiquement à compter s'il était déjà lancé.

Vous pouvez affecter une valeur initiale au compteur d'heures de fonctionnement à l'aide de la SFC2 SET_RTM. La SFC3 CTRL_RTM vous permet de le démarrer ou de l'arrêter. Avec la SFC4 READ_RTM, vous pouvez lire le nombre en cours d'heures de fonctionnement ainsi que l'état du compteur ("arrêté" ou "compte").

Une CPU peut posséder jusqu'à 8 compteurs d'heures de fonctionnement. La numérotation débute à 0.

A.6.4 Utilisation de mémentos de cadence et de temporisations

Mémento de cadence

Un mémento de cadence est un mémento dont l'état binaire change périodiquement dans un rapport impulsion-pause de 1:1. En paramétrant le mémento de cadence avec STEP 7 Lite, vous déterminez l'octet de mémentos de la CPU qui servira d'octet de mémentos de cadence.

Utilité

Vous pouvez vous servir de mémentos de cadence dans votre programme utilisateur pour commander, par exemple, des avertisseurs lumineux à feu clignotant ou pour déclencher des événements périodiques (comme l'enregistrement d'une valeur de mesure).

Fréquences possibles

Une fréquence est affectée à chaque bit de l'octet de mémentos de cadence. Le tableau suivant montre cette affectation.

Bits de l'octet de mémentos de cadence	7	6	5	4	3	2	1	0
Période (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Fréquence (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

Avertissement

Les mémentos de cadence sont asynchrones par rapport au cycle de la CPU. Dans les cycles longs, l'état du mémento de cadence peut donc changer plusieurs fois.

Temporisations

Les temporisations sont une zone de la mémoire système. La fonction d'une temporisation est définie par le programme utilisateur (par ex. retard à la montée). Le nombre de temporisations disponibles dépend de la CPU.

Avertissement

- Si vous employez dans votre programme utilisateur plus de temporisations que n'en autorise la CPU, une erreur synchrone sera signalée et l'OB121 déclenché.
 - Avec les S7-300, les temporisations ne peuvent être simultanément démarrées et actualisées que dans l'OB1 et dans l'OB100. Dans tous les autres OB, elles peuvent être seulement démarrées.
-

Index

*
*.k7e 8-2

A

Abréviations 3-9
Abréviations
sélection 6-64
Accès à la zone de données de périphérie.... A-110
Accès à la zone de données du processus ... A-109
Accès aux zones de données du processus
et de périphérie..... A-109
ACT_TINT 2-15, 2-16
Activation 6-11
Activation
affichage des mnémoniques dans le bloc.... 6-11
test avec état du programme 10-34
Actualisation
appels de bloc 6-67
Adressage
vérification 1-18
Adressage
absolu 6-1, 6-2
symbolique 6-1, 6-2
Adressage des modules A-109
Adressage symbolique 6-3
Adressage symbolique
dans l'exemple de programme A-81
Adresse de début A-110
Adresse d'entrée/sortie 5-14
Adresse d'entrée/sortie
attribuer 5-14
Adresses
remplacement dans les éléments CONT 6-52
saisie et édition dans des éléments CONT.. 6-51
Adresses
saisie dans les éléments LOG 6-60
Adresses PROFIBUS 5-14
Affectation
mnémoniques A-81, A-82, A-83
Affectation dans la mémoire image 5-19
Affectation de types de données
aux données locales de blocs de code A-63
Affectation des interruptions
vérification 1-18
Affichage 5-14
accès se recouvrant partiellement 6-89
agrandissement 6-32
blocs effacés..... 6-81
données locales requises au maximum..... 6-81
en arborescence 6-81
état de fonctionnement 7-3
état du module 11-2
longueurs de bloc 6-26
réduction..... 6-32

références croisées des opérandes
dont les plages d'adresses se recouvrent
en partie 6-89
zoom..... 6-33
Affichage
activation des mnémoniques dans le bloc.... 6-11
dans la visualisation d'état
de programme 10-28
définition pour l'état du programme 10-31
structure de blocs de données associés
à un UDT 6-73
structure de données de blocs de données
associés à un FB (DB d'instance)..... 6-71
vue synoptique 5-14
Affichage de la version du système d'exploitation
CPU dans la liste des modules..... 5-12
Affichage de la vue synoptique des adresses .. 6-77
Affichage des valeurs forcées de manière
permanente par la CPU..... 10-25
Affichage du catalogue du matériel 5-2
Afficher
commentaires 12-7
informations mnémoniques 12-7
mnémoniques 12-7
Afficher
barre d'état 6-17
barre d'outils 6-17
Agrandissement
affichage 6-32
Aide contextuelle 3-2
Aide de STEP 7 Lite 1-9
Aide en ligne
appel..... 3-2
rubriques 3-2
Alarme cyclique 2-14
déclenchement 2-19
règles..... 2-19
Alarme de diagnostic 11-9
Alarme de diagnostic (OB82) 11-30
Alarme de processus..... 2-14, 2-20
Alarme de processus
déclenchement 2-20
priorité 2-21
règles..... 2-20
Alarme horaire 2-14
changement de l'heure 2-16
déclenchement 2-15, 2-16, 2-17
désactivation 2-16
interrogation 2-16
priorité 2-16
règles..... 2-15
structure A-96
Alarme temporisée 2-14
déclenchement 2-18
priorité 2-18
règles..... 2-18
structure A-103

Alarmes	
paramétrer	5-6
Allocation de mémoire	
dans la pile L	A-19
Ambiguïté d'un mnémonique	6-8
Annulation	
tâche de forçage permanent.....	10-26
ANY	A-50, A-57, A-58, A-59
paramètre	
description et utilisation	A-60
Appel des fonctions d'aide.....	3-2
Appels de bloc	2-8, 2-9
actualisation.....	6-67
Appels imbriqués de blocs de code	
effets sur la pile B et la pile L.....	A-20
Application et enregistrement	4-7
Application et enregistrement des	
modifications.....	4-7
Appliquer	8-2
Arborescence	6-81
Architecture du système	
cycle	2-10
états de fonctionnement de la CPU	A-1
ARRAY	A-37
Arrêt	
état de fonctionnement de la CPU	A-1
Arrêt	
déterminer la cause	11-12
Arrêt (STOP).....	A-4
Attente	A-11
état de fonctionnement de la CPU	A-1
Automation License Manager.....	1-10
Avantages des fonctions d'horodatage	A-114
Avertissement.....	5-18
débordement de la pile L	A-18
Avertissement groupé.....	5-18

B

Barre de titre.....	3-3, 3-4
Barre des menus	3-3, 3-4
Barre d'état	3-3, 3-4
Barre d'état	
afficher/masquer	6-17
Barre d'outils.....	3-3, 3-4, 6-17
Barre d'outils	
afficher/masquer	6-17
Bas de page	12-1, 12-9
Bascule entre les différents types de fenêtres.	3-18
Base de temps pour S5 TIME.....	A-35
Bibliothèque standard.....	4-4
Bibliothèques	3-3, 3-4
Bibliothèques de blocs	
présentation.....	6-29
Bibliothèques d'opérations.....	6-31
Bits et octets utilisés	6-79
Blessures du personnel	
prévention.....	10-24
BLKMOV	A-14
Bloc	
ouverture dans la liste de la pile B.....	11-15
ouverture dans la liste de la pile L	11-15
ouverture pour une entrée en mémoire	
tampon de diagnostic	11-14

Bloc	
définition de l'environnement d'appel.....	10-32
pour modifier le pointeur	A-55
Bloc de code	
ouvrir	12-7
Bloc de données	
vue des déclarations.....	6-69
vue des données	6-70
Bloc de données (DB)	
blocs de données d'instance	2-29
global.....	2-31, 2-32
rémanent	A-23
structure	2-31
Bloc de données d'instance.....	2-29, 2-30
création de plusieurs instances	
pour un FB.....	2-26
Bloc de données d'instance	
rémanent	A-23
Bloc d'organisation (OB)	
OB d'arrière-plan (OB90).....	2-3, 2-23
BLOCK	
type de paramètre	A-50
BLOCK_DB	A-50
BLOCK_FB.....	A-50
BLOCK_FC	A-50
BLOCK_SDB.....	A-50
Blocs.....	1-6, 1-7, 1-8
charger dans CPU	9-2
charger dans la PG.....	9-10
comparaison.....	6-27
saisie en LIST.....	6-44
Blocs	
commentaires.....	6-45
dans le programme utilisateur	2-2
titres.....	6-45
Blocs chargés	
enregistrer dans l'EPROM intégrée	9-6
Blocs chargés dans PG/PC	
édition dans PG/PC	9-10
Blocs de code	
horodatage	6-92
Blocs de code	
dans l'éditeur de bloc.....	6-30
définition	
exemple.....	A-80
structure	6-30
Blocs de données	
modification de valeurs dans la vue	
des données.....	6-74
principes	6-68
réinitialisation de valeurs en leur	
substituant leur valeur initiale	6-75
saisie/affichage de la structure de données	
avec FB associé (DB d'instance).....	6-71
Blocs de données (DB).....	2-2
blocs de données d'instance	2-26
Blocs de données d'instance	2-29
horodatage	6-93
Blocs de données globaux	
horodatage	6-93
Blocs de données globaux	
saisie de la structure de données.....	6-71
Blocs de données globaux (DB)	2-31
Blocs déjà programmés.....	2-32

Blocs d'organisation	
détection d'erreur	
OB122	
valeurs de remplacement	11-26, 11-27
Blocs d'organisation	
classes de priorité	2-3, 2-4, 2-5
création d'un OB pour l'exemple	
d'un processus de mélange industriel.....	A-90
définition	2-3
informations de déclenchement.....	2-4
réaction aux erreurs.....	2-24
Blocs d'organisation (OB)	2-2
Blocs d'organisation et structure	
du programme	2-3
Blocs fonctionnels (FB).....	2-2
domaine d'application	2-27
paramètres effectifs	2-27, 2-28
Blocs fonctionnels (FB)	
création	
exemple de FB pour l'exemple	
de processus de mélange industriel	A-84
Blocs fonctionnels système	2-32
Blocs fonctionnels système	
types	2-32
Blocs fonctionnels système (SFB).....	2-2
Blocs fonctionnels système (SFB) et fonctions	
système (SFC).....	2-32
BM 147	5-9
Bobines	
placement	6-48
Boîte de dialogue 'Etat du module'	11-7
Boîtes	
placement.....	6-48, 6-58
Boîtes	
suppression	
modification	6-59
Boîtes de dialogue.....	3-6, 3-7
BOOL	
plage.....	A-27
type de données	A-26
Boutons	1-1, 1-5
Branche fermée	
création dans les réseaux CONT.....	6-55
Branche T	6-58
Branchements interdits en CONT	6-50
Branches	6-48, 6-49
Branches	
création dans les réseaux CONT.....	6-57
création dans les réseaux LOG	6-62
Branches (nouvelles)	
création dans les réseaux CONT.....	6-54
Branches parallèles	
création dans les réseaux CONT.....	6-54
Branches parallèles fermées	
ouverture dans les réseaux CONT	6-56
Bulle d'aide	3-2

C

CAN_TINT	2-16
Caractère (CHAR)	
plage.....	A-27
Caractère de commentaire	10-6
Caractère incomplet d'un mnémonique	6-8
Carte mémoire.....	A-14
paramétrage	1-17
Carte mémoire Micro (MMC)	8-2, 8-5, A-25
Carte mémoire S7	9-6
Carte MPI dans la PG ou le PC.....	1-18
Carte MPI-ISA (Auto)	1-18
Cartouches interface et interfaces	
(représentation dans HW Config)	5-11
Case à cocher	3-6
Case d'option.....	3-6
Certificate of License	1-10
Changement	
état de fonctionnement.....	7-3
heure pour l'alarme horaire.....	2-15
Changement d'état de fonctionnement.....	A-1
Charge du cycle due à la communication.....	2-10
Charge due à la communication.....	2-10, 2-13
Chargement	
conditions requises.....	9-1
depuis la CPU dans la PG.....	9-8
Chargement	
programme utilisateur.....	A-13
Chargement a posteriori de blocs	
dans la CPU	9-5
Chargement d'objets de la CPU	
dans la PG/le PC.....	9-10
Charger	
programme utilisateur dans la CPU.....	9-3
Charger dans CPU	9-2
Charger dans PG	9-9, 9-11
Charger de la CPU dans la PG	9-10
Charger des blocs dans la CPU	9-2
Charger la configuration matérielle	
dans la CPU	9-2
Charger quoi et quand dans la CPU ?	9-2
Charger une configuration dans la CPU.....	9-7
Charger une configuration	
dans un système cible	9-7
Choix	
mode de fonctionnement pour le test	10-33
Choix de la langue.....	8-13
Classement d'opérandes	6-4
Coller	
contenu du presse-papiers dans la table	
de variables	10-16
Coller le contenu du presse-papiers	
dans la table de variables.....	10-16
Coller le contenu du presse-papiers	
dans la table de variables ou	
dans la table de forçage	10-16
Combinaisons de touches	
déplacement du pointeur.....	3-16
Combinaisons de touches	
accès à l'aide en ligne	3-17
bascule entre les différents types	
de fenêtres	3-18
commandes de menu	3-14

- sélection de texte..... 3-17
- Commande d'une installation à distance
 - grâce au Téléservice 3-19
- Comment éviter des erreurs lors de l'appel
 - de bloc..... 6-95
- Commentaires
 - de blocs 6-46
 - de réseaux..... 6-45, 6-46
 - saisie dans les instructions LIST 6-66
- Commentaires de blocs
 - saisie 6-46
- Commentaires de réseaux
 - saisie 6-46
- Communication..... 2-10, 11-9
- Commuter sur "CPU en ligne" 1-5
- Comparaison de blocs 6-27
- Comparaison des configurations 'en ligne/hors
 - ligne/physique' 11-2
- Comparaison du matériel 1-5
- Comparaison HW 11-2
- Complexes
 - types de données A-37
- Comportement à la mise en route 5-6
- Comportement de rémanence
 - définir..... 5-6
- Comportement du système
 - paramétrer 5-6
- Comportement en fonctionnement
 - définition A-112
- Composition
 - liste des références croisées 6-77
- Compression
 - mémoire utilisateur 9-14
- Compression (ET 200S) 5-9
- Compression mémoire utilisateur (RAM) après
 - multiples effacements et chargements 9-14
- Compteur
 - zone de mémoire
 - rémanente A-23
- Compteur d'heures de fonctionnement A-114
- Compteurs
 - limites supérieures pour la saisie..... 10-11
- Compteurs utilisés 6-79
- Condition de déclenchement 10-17
- Condition de déclenchement pour l'affichage
 - de l'état du programme 10-32
- Conditions de déclenchement pour blocs 10-32
- Conditions préalables à l'installation 1-15
- Conditions requises pour le chargement 9-1
- Configuration
 - enregistrer 5-20
 - nécessaire quand ? 5-1
- Configuration
 - de systèmes intégrés compacts C7 5-12
 - matériel..... 4-10
- Configuration de station
 - charger dans un système cible 9-7
 - vérifier la cohérence 9-7
- Configuration et paramétrage d'une station..... 5-10
- Configuration matérielle..... 1-6, 1-8, 5-1, 5-4
 - charger dans CPU 9-2
 - charger dans la PG..... 9-10
- Configuration rapide 9-8
- Configurer..... 5-12
- Configurer ET 200S..... 5-9
- Conflit d'horodatage..... 6-90, 6-91, 6-92, 6-93, 6-94
- Connexion
 - insertion dans un réseau LOG..... 6-62
- Conseils
 - éditer une configuration de station 5-15
- Conseils de sécurité
 - débordement de la pile L..... A-18
- Conseils et astuces 13-1
- CONT 6-19, 6-20
 - branchements interdits 6-50
- CONT (schéma à contacts)..... 1-1
- Contenu des fenêtres 3-9
- Contenu des piles à l'état d'arrêt 11-12
- Contenu des registres 6-97
- Contrôle des modules
 - configuration prévue-configuration réelle..... 2-21
- Contrôle des temps de cycle pour éviter
 - les erreurs de temps..... 11-16
- Copie
 - variables dans une table de déclaration 6-40
- Copie
 - élément de projet..... 4-9
 - projet 4-9
 - zones sélectionnées dans le
 - presse-papiers..... 10-16
- Copie de lignes de mnémonique dans le
 - presse-papiers..... 6-16
- Copier..... 4-6
- Copier
 - projet 4-6
- Copier / Dupliquer une table de forçage..... 10-5
- Copier / Dupliquer une table de variables 10-4
- COUNTER
 - type de paramètre A-50
- Couper
 - zones sélectionnées et les copier
 - dans le presse-papiers 10-16
- Coupure secteur..... A-5
- Courant à l'état bloqué 5-16
- Court-circuit 6-50
- CPU
 - effacement général..... 9-12
- CPU (Central Processing Unit)
 - états de fonctionnement A-1, A-2, A-3
- CPU 31x A-23
- CPU 31x C 5-13
- CPU 31xC 8-2, 8-5, A-25
- CPU compacte A-23
- CPU compactes 5-13
- CPU en ligne 1-5
- CREAT_DB A-13
- Création
 - branches parallèles dans
 - les réseaux CONT 6-54
 - table de variables 10-3
- Création
 - branche fermée dans les réseaux CONT 6-55
 - branches dans les réseaux CONT..... 6-57
 - branches dans les réseaux LOG 6-62
 - de programmes utilisateur 6-30
 - FB pour le moteur..... A-84, A-85, A-86
 - FC pour les soupapes A-88, A-89
 - logiciel dans le projet (principe) 4-10
 - nouvelles branches dans les réseaux
 - CONT 6-54

OB1 pour l'exemple d'un processus de mélange industriel	A-90
structure.....	A-46, A-47
tableaux.....	A-43, A-45
tâche de forçage permanent.....	10-26
type de données utilisateur.....	A-48, A-49
Création d'un projet	4-4
Création d'une tâche de forçage permanent..	10-25
Créer et ouvrir une table de forçage	10-3
CTRL_RTM	A-115
Cycle	2-3, 2-4, 2-10, 2-11, 2-12
charge due à la communication.....	2-14
Cyclique	
traitement du programme	2-6
D	
Date	
réglage.....	7-3
DATE AND TIME (date et heure)	
format	A-38
page.....	A-38, A-39
DATE_AND_TIME	A-37
DB	2-31, 2-32
DCB	A-34
Débordement de la pile L	A-18
Décalage de phase.....	2-19
Déclaration de paramètres	
FC pour l'exemple d'un processus de mélange industriel	A-88
Déclaration de variables locales	
FB pour l'exemple d'un processus de mélange industriel	A-84
OB pour l'exemple d'un processus de mélange industriel	A-90
Déclenchement	
alarme cyclique.....	2-19
alarme horaire	2-15, 2-16, 2-17
alarme temporisée.....	2-18
Déclenchement	
alarme de processus	2-20
Déclenchement du traitement du programme par alarme	2-3
Décomposition	
jonction dans les réseaux CONT	6-56
Défaillance d'unité (OB86).....	11-33
Défauts	
localisation.....	11-2
Définir les propriétés de modules/interfaces....	5-13
Définition	
affichage de l'état du programme	10-31
blocs de code	A-80
de mnémoniques lors de la saisie du programme	6-11
environnement d'appel du bloc.....	10-32
format d'affichage	10-16
Définition de l'environnement d'appel de blocs	10-32
Définition de mnémoniques individuels	6-13
Définition du comportement en fonctionnement	A-112
Définition du niveau de protection définir.....	5-6
Définir les propriétés de composants	5-6
Démarrage	
installation de STEP 7 Lite	1-16
STEP 7 Lite	3-1
Démarrage à chaud.....	2-21, 2-22, 9-1,
.....	A-1, A-4, A-5, A-6, A-7
abandon	A-7
automatique.....	A-5
automatique sans sauvegarde	A-5
manuel.....	A-5
Démarrage à froid	2-21, 2-22, A-1, A-4, A-5,
.....	A-6, A-7
Démarrage de secours.....	5-18
Démasquage	
événements déclencheurs.....	2-24
Démasquage d'événements d'erreurs synchrones	
exemple.....	A-72
Dépannage.....	13-2
Départ-moteur	5-9
Déplacer	
objets.....	3-13
Déplacer les objets à imprimer	12-3
Déplacer un module	5-15
Désactivation	
alarme horaire	2-15
Désactivation	
test avec état du programme.....	10-34
Désactivation avec redémarrage (Autoreset) ...	5-18
Désactivation Fin de course Marche à droite / Désactivation Fin de course Marche à gauche.....	5-18
Désactivation sans redémarrage.....	5-18
Déséquilibre	5-17
Désinstallation	
de la licence d'utilisation	1-14
STEP 7 Lite	1-19
Détails	1-5
Détection d'erreur	
exemples de programme valeurs de remplacement	11-26
Détection d'erreur	
utilisation d'OB d'erreur en réaction aux erreurs	2-24
Détection d'erreurs	
types d'OB	
OB 81	11-25
Détection du courant résuduel.....	5-16
Diagnostic.....	11-5
du matériel.....	11-6
fonctions	11-1
mémoire tampon	11-1
Diagnostic du matériel	1-5, 11-2
Diagnostic module.....	11-7
Diagnostic système	
extension	11-20
Différence entre l'enregistrement et le chargement de blocs	9-3
Différence entre paramétrage nominal et effectif.....	A-4
Différences entre forçage de variables et forçage permanent de variables	10-26
DINT	
type de données.....	A-28
DIS_AIRT	2-24
DIS_IRT.....	2-24
Disposer	5-12

Disposer	
des systèmes intégrés compacts C7	5-12
Disposer des modules dans le profilé	
support/châssis	5-11
Dispositif antibloqueur	5-16
Disposition	
boîtes	6-58, 6-59
Disposition des fenêtres	3-9
enregistrer	3-10
modifier	3-9
restaurer	3-10
DM 370 Dummy	5-8
DMSK_FLT	2-24
Documentation	1-5, 1-7
imprimer	12-13
Documentation du logiciel STEP 7 Lite	1-9
Documentation du projet	1-7, 12-1
établir	12-3
Dommages matériels	
prévention	10-24
Données de diagnostic	11-1, 11-19
Données de diagnostic	
sur les modules	11-18
Données de périphérie	A-110
Données de référence	6-15, 6-16
application	6-76
Données de référence possibles	6-76
Données d'état du diagnostic	11-18
Données locales requises	6-81
Données système	11-19
Données utiles	A-110
Double mot (DWORD)	
plage	A-27
type de données	A-26
Droit d'accès	7-2
Durée d'interruption	A-4
DWORD	
type de données	A-34

E

Echange de données	
à différents états de fonctionnement	A-9
Echange de données de projet entre	
STEP 7 Lite et STEP 7	8-6
Editeur de bloc	1-8, 6-18
paramètres	6-31
Edition	
adresses ou paramètres dans	
des éléments CONT	6-51
blocs chargés dans PG/PC	9-10
Edition	
dans la table des mnémoniques	6-11
Edition d'une configuration matérielle	
chargée dans votre PG/PC	9-11
Effacement	
mémoire de chargement/travail	9-12
Effacement de la carte mémoire	
dans la CPU	9-14
Effacement général	A-4
CPU	9-12
Effacer et insérer une connexion	
dans un réseau LOG	6-63
Effacer et renommer un projet	4-10
Élément de projet	

copie	4-9
Éléments CONT	6-48
remplacement	6-52
remplacement d'adresses	6-52
remplacement de paramètres	6-52
représentation	6-47
saisie et édition d'adresses	6-51
saisie et édition de paramètres	6-51
Éléments CONT	
saisie	6-51
Éléments de données du type STRUCT	
saisie dans la table de déclaration	6-38
Éléments de langage	6-31
Éléments des boîtes de dialogue	3-6
Éléments du projet	
imprimer	12-13
Éléments LOG	
remplacement	6-61
représentation	6-57
Éléments LOG	
saisie	6-60
saisie d'adresses ou de paramètres	6-60
EM 300	5-9
Emplacements (S7-300)	5-7
EN / ENO	
connexion	6-59
EN 6.1131-3	1-4
En ligne (configuration)	11-2
EN_AIRT	2-24
EN_IRT	2-24
Enregistrement	
accès	A-111, A-112
écriture	A-110
lecture	A-110, A-111
table de variables	10-5
table des mnémoniques	6-16
Enregistrement d'une configuration	5-20
Enregistrement et application	4-7
Enregistrer	8-2
blocs chargés dans la carte mémoire	9-6
blocs chargés dans l'EPROM intégrée	9-6
disposition des fenêtres	3-10
Enregistrer des données de projet	
sur une carte mémoire Micro (MMC)	8-2
Enregistrer des projets sur un support	
de données	8-2
Enregistrer sous	8-1, 8-2
Enregistrer une table de forçage	10-5
En-tête	12-1, 12-7
Entrée en mémoire tampon de diagnostic	11-14
Entrées	
mémoire image	A-17
Entrées utilisées	6-79
Envoi	
de vos propres messages de diagnostic	11-20
Envoi de vos propres messages	
de diagnostic	11-20
Epingle	3-6
EPROM	9-6, A-24
Erreur	
durant l'installation	1-17
Erreur d'accès à la périphérie (OB122)	11-36
Erreur d'accès à la périphérie lors	
de la mise à jour de la mémoire image	A-18
Erreur d'alimentation (OB81)	11-29
Erreur de communication (OB87)	11-34

Erreur de programmation (OB121)	11-35
Erreur de temps (OB80)	11-29
Erreur d'exécution du programme (OB85)	11-32
Erreur matérielle CPU (OB84)	11-31
Erreur système	11-1
Erreurs asynchrones	
OB 81	11-23
Erreurs asynchrones	
utilisation d'OB en réaction aux erreurs	2-24
Erreurs détectables	11-23
Erreurs lors de l'appel de bloc	
éviter	6-95
Erreurs synchrones	
utilisation d'OB en réaction aux erreurs	2-24
ET 200S	5-9
ET 200X	5-9
Etablissement	
liaisons en ligne	7-1
Etapes d'une solution d'automatisation	1-1
Etat d'arrêt	
contenu des piles	11-12
Etat de fonctionnement	
"Arrêt"	A-1, A-3
"Arrêt" (STOP)	A-4
"Attente"	A-1, A-3, A-11
"Marche"	A-1, A-3
"Marche" (RUN)	A-9
"Mise en route"	A-1, A-3, A-4, A-8
affichage et changement	7-3
Etat de fonctionnement de la CPU	
modification lors du chargement	9-7
Etat du module	11-6, 11-7, 11-11
actualiser	11-10
affichage	11-2
fonctions d'information	11-9
Etat du programme	
forçage de variables	10-34
Etat du programme	
activation et désactivation du test	10-34
définition de l'affichage	10-31
Etat du programme de blocs de données	10-30
Etats de fonctionnement	
priorité	A-3
Etats de fonctionnement de la CPU	A-1
Événement de diagnostic	11-1, A-21
Événements	
asynchrones	2-10
Événements asynchrones	
inhibition et validation	A-75
traitement différé	A-76
Événements d'alarme	
inhibition et validation	A-75
traitement différé	A-76
Événements déclencheurs	
masquage	2-25
OB de mise en route	2-21
retardement	2-24
Événements d'erreurs synchrones	
masquage et démasquage	A-72
Exemple	
format de nombres à virgule flottante	A-29
inhibition et validation d'événements	
d'alarme et d'événements asynchrones	
(SFC 39 et 40)	A-75
masquage et démasquage d'événements	
d'erreurs synchrones	A-72

saisie d'une plage d'opérandes continue	10-13
traitement différé d'événements d'alarme	
et d'événements asynchrones	
(SFC 41 et 42)	A-76
type de données STRING	A-39
Exemple de programme	
processus de mélange industriel	A-77
Exemple de saisie d'opérandes dans une	
table de variables	10-12
Exemple d'utilisation d'occurrences	6-87
Exemples de programme	A-71
FB pour l'exemple d'un processus	
de mélange industriel	A-84
FC pour l'exemple d'un processus	
de mélange industriel	A-88
insertion de valeurs de remplacement	11-26
OB pour l'exemple d'un processus	
de mélange industriel	A-90
valeurs de remplacement	11-26
Exemples de projet	A-71
Exploitation	
paramètre de sortie RET_VAL	11-22
Exportation	
tables des mnémoniques	6-13
Exportation de textes à gestion multilingue	8-12
Exporter	8-1
Exporter des textes pour les traduire	8-8

F

F1	3-2
FB	2-26, 2-27, 2-28, A-37
FC	2-26
FC12	A-98
Fenêtre	
taille de l'affichage	6-17
Fenêtre "Catalogue du matériel"	
utilisation	5-15
Fenêtre de travail	
affichage	
réduction/agrandissement	6-32
zoom	6-33
fractionnement	6-33
Fenêtre du projet	4-1, 4-2, 4-3
Fenêtre du projet et vues dans STEP7Lite	1-5
Fenêtres	
basculer entre les différents types	3-18
FEPROM	A-23
Fichier *.awl	8-2, 8-4, 8-6, 8-7
Fichier *.k7p	8-2
Fichier *.sdf	8-2, 8-4, 8-6
Fichier awl	8-6
Fichier d'exportation	8-10, 8-11
Fichier d'exportation S7Lite	8-2
Fichier sdf	8-6
Filtrage	
mnémoniques	6-12
Filtrage	
table des mnémoniques	6-15
Fonction (FC)	2-26
Fonction (FC)	
création	
exemple de FC pour l'exemple	
d'un processus de mélange industriel	A-88
domaine d'application	2-26

Fonction de recherche d'erreurs
 dans la section des instructions..... 6-47

Fonctions (FC)..... 2-26

Fonctions de diagnostic..... 11-1

Fonctions d'horodatage..... A-114

Fonctions d'information..... 11-11

Fonctions d'information de l'état du module 11-9

Fonctions système..... 2-32, 2-33

Fonctions système
 types..... 2-32

Fonctions système (FC) 2-2

Forçage 10-26

définition du mode 10-20

initialiser la CPU à l'arrêt avec vos
 propres valeurs..... 10-23

marche à suivre 10-2

sorties de périphérie à l'état "Arrêt"
 de la CPU 10-23

Forçage de variables
 avec déclenchement défini 10-22

dans l'état du programme 10-34

immédiat 10-23

introduction 10-20

Forçage et visualisation 1-5

Forçage permanent 1-6, 10-26

Forçage permanent de variables
 introduction 10-24

mesures de sécurité 10-25

Format

BLOCK A-51

COUNTER..... A-51

TIMER A-51

type de données ARRAY..... A-40

type de données DATE_AND_TIME
 (date et heure)..... A-38

type de données DINT (entiers de 32 bits) .. A-28

type de données INT (entiers de 16 bits)..... A-28

type de données REAL
 (nombres à virgule flottante)..... A-29

type de données S5TIME (durée)..... A-35

type de données STRING..... A-39

type de données STRUCT..... A-41

type de paramètre ANY A-57

type de paramètre POINTER..... A-51, A-52

types de données WORD et DWORD
 pour les nombres décimaux
 codés binaire A-34

types de paramètre BLOCK
 COUNTER et TIMER..... A-51

Format d'affichage
 définition 10-16

Format DCB..... A-35

Format de données pour l'importation/exportation
 d'une table des mnémoniques..... 8-8

Format du type de données DWORD..... A-33

Format du type de données TIME A-36

Format du type de données WORD A-33

Format horaire A-114

Format pointeur A-50, A-52

Fractionnement
 fenêtre de travail 6-33

G

Gestion multilingue des textes.....8-8

H

Heure
 changement..... 2-16, 2-17

lecture..... A-114

réglage 7-3, A-114

Heure (TIME OF DAY)
 plage..... A-27

Hiérarchie d'appel dans le
 programme utilisateur.....2-8

Horloge
 paramétrage A-114

synchronisation A-114

Horodatage..... 6-91, 11-9

blocs de code6-92

blocs de données d'instance6-93

blocs de données globaux.....6-93

UDT et DB dérivés d'UDT.....6-94

Hors ligne (configuration)11-2

I

Icônes
 dans la vue d'ensemble du projet 1-1

Icônes
 dans la fenêtre de projet..... 1-5

Icônes (voir icônes) 1-5

Icônes (voir Icônes) 1-1

Icônes dans la table de comparaison HW 11-2

Icônes utilisées dans la fenêtre du projet 3-5

Identification
 de mnémoniques.....6-3

Importation
 source externe.....4-4

table des mnémoniques6-13

Importation de textes à gestion multilingue8-13

Importer 8-1, 8-6, 8-7

Importer des textes pour les traduire.....8-8

Imprimer
 blocs.....12-13

contenu de la mémoire tampon
 de diagnostic12-13

documentation du projet12-13

données de référence.....12-13

éléments du projet.....12-13

objets individuels12-13

table de configuration12-13

table de variables12-13

table des mnémoniques12-13

Indirect
 paramétrageA-112, A-113

Informations mnémonique.....6-2

Informations succinctes3-2

Inhibition d'événements d'alarme
 et d'événements asynchrones exemple A-75

Insérer
 connexion dans un réseau LOG6-62

Insertion
 lignes vides dans une table de déclaration
 de variables6-38

mnémoniques	6-10
programme	4-5
réseaux CONT supplémentaires	6-53
valeurs de remplacement en cas d'erreur..	11-26
Insertion	
d'une plage d'opérandes continue	
dans une table de variables.....	10-9
lignes de mnémonique	6-14
réseaux LIST supplémentaires	6-66
réseaux LOG supplémentaires	6-62
Insertion d'un plage d'opérandes continue	
dans une table de forçage	10-9
Installation	
STEP 7 Lite	1-15
Installation de Automation License Manager...	1-12
Installation de STEP 7 Lite	1-15
Instance	2-29
Instances du bloc.....	10-32
Instructions	
saisie	
marche à suivre	6-44
Instructions CONT	
règles de saisie.....	6-48
Instructions LIST	
règles de saisie.....	6-64
Instructions LIST	
saisie	6-65
saisie de commentaires.....	6-66
sélection de zones de texte	6-65
Instructions LOG	
règles de saisie.....	6-58
Instructions tirées des bibliothèques	
d'opérations	6-31
INT	
type de données	A-28
Interaction entre la table de déclaration des	
variables et la section des instructions	6-35
Interface DP de la CPU	5-6
Interface MPI	1-15
Interface MPI de la CPU	5-6
Interface PG/PC	1-18
paramétrage	1-18
Interface PROFIBUS DP de la CPU	5-6
Interface utilisateur	3-3
Interrogation de l'alarme horaire	2-16
Intervalle dans la mémoire utilisateur (RAM) .	9-14
Introduction à la configuration matérielle	5-1
Introduction au test avec des tables	
de variables et des tables de forçage	10-1
Introduction aux types de données	
et de paramètre	A-26

J

Jonction	
décomposition dans les réseaux CONT	6-56

K

k7e.....	8-2
k7p.....	8-2

L

Langage de programmation	
choix	6-19
Langage de programmation	
CONT (schéma à contacts)	6-20
LOG (logigramme).....	6-21
passage de l'un à l'autre	6-34
Langage de programmation LIST	
(liste d'instructions).....	6-21
Langages de programmation.....	1-3, 1-4
Largeur de zone d'opérande	6-47, 6-57
Largeur des colonnes d'une table	
de déclaration modification.....	6-40
Largeur des colonnes d'une table	
de déclaration régler.....	6-33
Liaisons en ligne	
établissement	7-1
Licence d'utilisation avec	
Automation License Manager.....	1-10
License Key.....	1-10, 1-14
License Manager.....	1-10, 1-11
Ligne de commentaire	10-6
Lignes de mnémonique	
copie dans le presse-papiers.....	6-16
insertion.....	6-14
sélection	6-16
suppression	6-14
Lignes vides	
insertion dans une table de déclaration de	
variables	6-38
Limites supérieures pour la saisie	
de compteurs.....	10-11
Limites supérieures pour la saisie	
de temporisations	10-10
LIST	6-19, 6-21
saisie de blocs.....	6-44
LIST	
paramètres	6-64
LIST (liste d'instructions)	1-1
Liste déroulante.....	3-6
Liste des références croisées.....	1-7, 6-76, 6-77
Liste d'états système	11-18
Liste d'états système	
contenu.....	11-18
lecture.....	11-18
Liste d'états système (SZL)	11-18
Liste d'impression.....	12-3, 12-7
Liste d'instructions.....	6-21
Lizense	1-10
Localisation des défauts	11-2
LOG.....	6-19, 6-21
LOG (logigramme).....	1-1
Logiciel optionnel.....	3-19
Logigramme	6-21
Longueurs de bloc	
affichage.....	6-26

M

- MAJ+F1 3-2
- Manipulations de base pour la configuration matérielle 5-2
- Marche
 - activités de la CPU A-8
 - état de fonctionnement de la CPU A-1
- Marche (RUN) A-9
- Marche à suivre
 - affichage de l'état de fonctionnement et changement 7-3
 - affichage et réglage de l'heure et de la date.. 7-3
 - ouverture du bloc pour une entrée
 - en mémoire tampon de diagnostic..... 11-14
 - pour saisir des instructions 6-44
 - pour visualiser et forcer 10-2
- Marche à suivre
 - pour créer des blocs de code 6-30
 - pour déterminer la cause d'un passage à l'état d'arrêt..... 11-12
- Marche à suivre de principe pour la visualisation et le forçage avec des tables de forçage 10-2
- Marche à suivre pour configurer une station 5-3
- Marche à suivre pour installer STEP 7 Lite 1-16
- Masquage
 - événements déclencheurs..... 2-24
- Masquage d'événements d'erreurs synchrones
 - exemple A-72
- Masquer
 - barre d'état 6-17
 - barre d'outils 6-17
- Matériel..... 1-6
 - comparaison 11-4
 - configuration 5-1, 5-4
 - diagnostic 11-2, 11-7
- Matériel
 - configuration 4-10
- Matériel pris en charge 1-1
- Mémento
 - zone de mémoire rémanente A-23
- Mémento de cadence
 - configurer..... 5-6
- Mémentos de cadence A-115
- Mémentos utilisés..... 6-79
- Mémoire 11-9
- Mémoire circulante (mémoire tampon de diagnostic) A-21
- Mémoire de chargement..... 9-3, 9-4, A-12, A-13
- Mémoire de chargement
 - enregistrement de DB non significatifs pour l'exécution..... A-13
- Mémoire de chargement et mémoire de travail A-13
- Mémoire de chargement et mémoire de travail dans la CPU..... 9-3
- Mémoire de chargement/travail
 - effacement..... 9-12
- Mémoire de session 3-9
- Mémoire de travail.....9-3, 9-4, A-12, A-13, A-14
- Mémoire image
 - entrées/sorties A-17
 - mise à jour 2-11
- Mémoire image
 - effacer2-21
- Mémoire image du processus 2-10, A-17
- Mémoire rémanente
 - des CPU S7-300 A-23
- Mémoire système A-12
- Mémoire tampon de diagnostic 11-9, A-21
 - contenu..... 11-1, A-21, A-22, A-23
 - définition A-21
 - exploitation A-21
 - lecture..... 11-17, A-23
- Mémoire utilisateur
 - compression9-14
- Mémoire vive A-12
- Mémoire vive non volatileA-23, A-24
- Mémoriser des informations provenant de la CPU9-8
- Menu d'insertion 12-9, 12-10
- Message de diagnostic
 - écriture de vos propres messages11-20
 - envoi aux correspondants11-20
- Mesures à prendre dans le programme
 - pour traiter les erreurs11-21
- Mesures de sécurité pour le forçage permanent de variables10-25
- Méthodes de conception
 - conception d'un programme structuré A-80
- Microsoft Windows1-15
- Mise à jour
 - mémoire image..... A-17
 - mémoire image du processus2-10
- Mise en page..... 12-1, 12-9, 12-10
- Mise en page CONT.....6-47
- Mise en page LOG6-57
- Mise en route..... A-4, A-5, A-6, A-7, A-9
 - abandon A-4
 - activités de la CPU A-4
 - état de fonctionnement de la CPU..... A-1
- MMC..... 8-2, 8-4, 8-5
- Mnémonique
 - affecter A-81
- Mnémoniques
 - dans la table des mnémoniques6-5
 - définition individuelle6-13
 - édition.....6-13
 - filtrage.....6-12
 - incomplets6-8
 - insertion.....6-45
 - non univoques6-8
 - saisie dans une table de variables10-6
- Mnémoniques
 - définition lors de la saisie du programme6-11
 - globaux..... 6-2, 6-3
 - locaux 6-2, 6-3
 - saisie6-12
 - tri6-12
- Mnémoniques de bloc6-9
- Mnémoniques globaux
 - saisie dans un programme6-45
- Mnémoniques globaux
 - saisie dans la table des mnémoniques.....6-12
 - saisie individuelle dans les boîtes de dialogue.....6-11
- Mnémoniques globaux et mnémoniques locaux.....6-2

Mnémoniques incomplets ou non univoques	
dans la table des mnémoniques	6-8
Mnémoniques individuels	
définition	6-13
Mnémoniques manquants	6-16
Mode de fonctionnement	
choix pour le test	10-33
Mode de forçage.....	10-20
Mode de substitution	6-47, 6-52, 6-61
Mode de visualisation	10-17
Mode terminal.....	5-18
Modèle de documentation	12-1, 12-10
Modèle de mise en page	12-1, 12-10
Modèle de moteur thermique.....	5-17
Modèle de police	12-1, 12-10
Modification	
largeur des colonnes d'une table	
de déclaration de variables.....	6-40
Modification	
valeurs dans la vue des données	
de blocs de données	6-74
Modification du comportement	
et des propriétés des modules.....	A-112
Modifier la disposition des fenêtres	3-9
Module	
déplacer.....	5-15
remplacer.....	5-15
Module	
adresse de début.....	A-109
Module fictif DM 370 Dummy.....	5-8
Module RTD	5-9
Module TC	5-9
Module TOR de simulation SIM 374 IN/OUT	5-8
Modules	5-6
remplacement.....	13-1
Modules	
paramétrage	A-112
spécification.....	9-11
Modules de signaux aptes aux alarmes	
de processus paramétrage.....	2-20, 2-21
Modules défectueux	11-5
Modules paramétrables	A-112
Mot (WORD)	
plage.....	A-27
type de données	A-26
Mot de passe	7-2
Mot de passe	
attribuer	5-6
Moteur-DROITE / Moteur GAUCHE	5-18
MSK_FLT	2-24
Multi-instance	2-26, 2-29
saisie dans la table de déclaration	
des variables	6-42
Multi-instances	
règles de formation.....	6-42
Multi-instances	
utilisation.....	6-41

N

Niveau de protection	
définir.....	5-6
Niveaux de priorité	5-6
Nombre à virgule flottante	
éléments de base	A-29
exemple.....	A-29, A-30, A-31
paramètres	A-29
zones de composants.....	A-29
Nombre entier (16 bits)	
format	A-28
Nombre entier (16 bits) (INT)	
plage.....	A-27
Nombre entier (32 bits)	
format	A-28
Nombre entier (32 bits) (DINT)	
plage.....	A-27
Nombre réel	
plage.....	A-27
type de données	A-27
Nombres à virgule flottante	
format	A-29, A-30, A-31
Nombres décimaux codés binaire (DCB)	A-34
Nouveau projet.....	1-5
Nouvelles branches	
création dans les réseaux CONT	6-54
NVRAM	A-23

O

OB	2-3, 2-4, 2-5, 2-6
OB d'alarme	
paramétrage	2-15, 2-16
utilisation	2-14
OB d'alarme	
désactivation	2-3
paramétrage	2-3
OB d'alarme cyclique (OB30 à OB38)	2-19
OB d'alarme de diagnostic	11-30, 11-31
OB d'alarme de processus (OB40 à OB47).....	2-20
OB d'alarme horaire (OB10 à OB17).....	2-15
OB d'alarme temporisée (OB20 à OB23)	2-18
OB d'arrière-plan	
priorité	2-23
programmation	2-23
OB de défaillance d'unité.....	11-33
OB de mise en route	A-4
configuration prévue-configuration réelle.....	2-21
événements déclencheurs.....	2-21
OB de mise en route	
contrôle des modules	2-22
OB d'erreur	11-23
OB d'erreur	
types d'OB	
OB121 et OB122	2-24
OB80 à OB87	2-24
utilisation d'OB d'erreur en réaction aux	
événements.....	2-24
OB d'erreur d'accès à la périphérie	11-36
OB d'erreur d'alimentation.....	11-29
OB d'erreur de communication	11-34
OB d'erreur de programmation.....	11-35
OB d'erreur de temps	11-29

OB derreur d'exécution du programme 11-32
 OB d'erreur en réaction à la détection
 d'une erreur 11-23
 OB d'erreur matérielle CPU 11-31
 OB pour exécution cyclique
 du programme (OB1) 2-10
 OB pour la mise en route (OB100/OB102) 2-21
 OB pour le traitement de programme
 déclenché par alarme 2-14
 OB pour le traitement d'erreurs (OB80 à
 OB87/OB121 à OB122) 2-24
 OB pour l'exécution du programme
 en arrière-plan (OB90) 2-23
 OB1 A-107
 OB1 et OB80 A-102
 OB10 A-100
 OB10 à OB17 2-15
 OB100 2-21, 2-22, A-4
 OB102 2-21, 2-22, A-4
 OB121 11-35
 OB121 et OB122 2-24
 OB122 11-36
 OB20 A-105
 OB20 à OB23 2-18
 OB30 à OB38 2-19
 OB40 à OB47 2-20
 OB80 11-29
 OB80 à OB87 2-24
 OB81 11-29
 OB82 11-30
 OB84 11-31
 OB85 11-32, A-17
 OB86 11-33
 OB87 11-34
 OB90 2-23
 Objet à imprimer 12-1, 12-8, 12-9
 Objets
 déplacer 3-13
 renommer 3-13
 supprimer 3-13
 Objets à imprimer 12-5, 12-6
 changer l'ordre 12-3
 Octet
 plage A-27
 Octet (BYTE)
 type de données A-26
 Onglet 3-6
 Opérandes 6-7
 saisie dans une table de variables 10-6
 Opérandes et types de données autorisés
 dans la table des mnémoniques 6-7
 Opérandes libres 6-15
 Opérandes utilisés 1-7, 6-76, 6-79
 Opérations
 insertion 6-31
 Optimisation de la traduction 8-15
 Optimisation du modèle à traduire 8-14
 Organisation des zones de mémoire A-12
 Ouverture
 bloc dans la liste de la pile B 11-15
 bloc dans la liste de la pile L 11-15
 du bloc pour une entrée en mémoire tampon
 de diagnostic 11-14
 table de variables 10-3
 Ouverture

branches parallèles fermées dans les
 réseaux CONT 6-56
 table des mnémoniques 6-12
 Ouvrir 4-6
 table des mnémoniques 6-13
 Ouvrir
 projet 4-6

P

Page de garde 12-3
 Pagination 12-3
 Panneau de commande CPU 3-3, 3-4
 Paquette optionel 10-1
 Paramètres formels 2-26
 Paramétrage 5-1, 6-4
 horloge A-114
 interface PG/PC 1-18
 Paramétrage
 avec SFC A-112
 avec STEP7Lite A-112
 du classement d'opérandes
 (absolu/symbolique) 6-4
 indirect A-112
 modules de signaux aptes aux alarmes
 de processus 2-20
 Paramétrage dans le programme utilisateur 5-6
 Paramétrage de l'interface PG/PC 1-18
 Paramétrage de soudures froides 5-9
 Paramétrage des interfaces de la CPU 5-6
 Paramètre de CPU "Charge du cycle due
 à la communication" 2-10
 Paramètre de sortie RET_VAL
 exploitation 11-22
 Paramètres
 éditeur de bloc 6-31
 pour le langage de programmation LOG 6-57
 remplacement dans les éléments CONT 6-52
 saisie et édition dans des éléments CONT 6-51
 Paramètres
 pour le langage de programmation LIST 6-64
 saisie dans les éléments LOG 6-60
 Paramètres de modules A-112
 Paramètres de modules
 transfert à l'aide de SFC A-112
 transfert avec STEP7Lite A-112
 Paramètres effectifs .. 2-26, 6-67, A-65, A-66, A-67,
 A-68, A-69
 Paramètres formels 6-67, A-65, A-66, A-67,
 A-68, A-69
 Paramètres pour le langage de programmation
 CONT 6-47
 PARM_MOD A-111, A-113
 Passage
 d'un langage de programmation à un autre .. 6-34
 Passage d'un état de fonctionnement
 à un autre A-1
 Paysage 12-1
 Performances 11-9
 Périphérie
 zones de données A-109
 Physique (configuration) 11-2
 Pictogrammes (voir Icônes) 1-1, 11-2

Pile B	
appels imbriqués	A-20
données enregistrées dans la pile B.....	A-20
Pile des blocs	A-12, A-20
Pile des données locales.....	A-12, A-18
Pile des interruptions	A-19
Pile I	
description	A-19
utilisation par la mémoire système	A-19
Pile L	
allocation de mémoire aux	
variables locales	A-18
écraser.....	A-18
enregistrement de variables temporaires.....	2-26
Pile L	
traitement de données dans un	
appel imbriqué	A-20
Piles.....	11-9
Piles des interruptions	A-12
Placement	
boîtes.....	6-58
Pneumatic-Interface-Modul.....	5-9
Point de déclenchement	10-17, 10-18,
.....	10-20, 10-21
POINTER.....	A-50
POINTER	
type de paramètre	A-50
Pointeur zéro	A-57, A-58
Police.....	12-1, 12-8
Police non proportionnelle	12-7
Police proportionnelle	12-8
Portrait.....	12-1
Positionnement rapide sur des occurrences	
dans le programme.....	6-86
Possibilités de chargement.....	A-14
Possibilités de chargement selon la mémoire	
de chargement.....	9-4
Possibilités de saisie de	
mnémoniques globaux	6-9
Pour STEP7.....	8-6
Powermodul.....	5-9
Présélections pour l'éditeur de bloc	
CONT/LOG/LIST	6-31
Présentation	
bibliothèques de blocs	6-29
modes de test.....	10-1
Présentation de STEP 7 Lite	1-1
Prévention	
blessures du personnel	10-24
dommages matériels	10-24
Principes	
blocs de données	6-68
Priorité	5-6
alarme horaire	2-15
alarme temporisée.....	2-18
Priorité	
alarme de processus	2-21
définir pour les alarmes	5-6
modification	2-4
OB d'arrière-plan	2-23
Processus	
subdivision.....	A-77
Profondeur d'imbrication.....	2-8
Programmation	
transmission de paramètres	2-26
utilisation de blocs de données.....	2-26
Programmation	
conception d'un programme structuré	A-80
d'un OB1	
exemple.....	A-91
d'une FC	
exemple.....	A-88
FB	
exemple.....	A-86
OB d'arrière-plan	2-23
Programmation linéaire	2-7
Programmation structurée.....	2-3
Programme.....	1-2, 1-3, 1-4, 1-6, 1-7, 1-8
charger dans la PG.....	9-10
insertion.....	4-4
Programme	
données de référence.....	1-5
Programme de mise en route	2-22
Programme structuré	
avantages.....	2-2
conception	A-80
Programme utilisateur	9-10
charger dans la CPU	9-3
Programme utilisateur	
chargement	A-13, A-14
dans la mémoire de la CPU.....	A-13
éléments.....	2-2
tâches.....	2-1
Programmes dans une CPU.....	2-1
Programmes exemple	
réaction à une défaillance de pile	11-23
Projet.....	1-5, 1-6, 1-7, 1-8
création.....	4-4
vue d'ensemble	4-1
Projet	
copie.....	4-9
copier.....	4-6
création du logiciel (principe).....	4-10
ouvrir	4-6
supprimer	4-6
Projets	
ordre de traitement.....	4-4
Propriétés de bloc	1-8, 6-30
affichage des longueurs	6-26
Propriétés de bloc autorisées	
pour chaque type de bloc	6-26
Protection de l'accès aux systèmes cibles	
par mot de passe.....	7-2
Protection KNOW HOW	6-26
Q	
QRY_TINT.....	2-16
Qu'est-ce que c'est ?	3-2
Qu'est-ce qu'un projet STEP 7 Lite ?.....	4-1

R

RAM	A-12, A-23
Rapport entre les états de fonctionnement de la CPU	A-1
RDSYSST	11-17, 11-18, A-23
READ_CLK	A-114
READ_RTM	A-115
Real	
type de données	A-29
Recharger quoi et quand dans la PG ?	9-9
Recherche d'erreurs	11-2
Recherche d'erreurs	
dans les blocs	6-47
Recherche ou remplacement de termes	3-11
Réduction	
affichage	6-32
Registre adresses	6-97
Registre des DB	6-97
Registres de bloc de données	A-20
Réglage	
heure et date	7-3
Règle	
pour l'importation de tables des mnémoniques	6-13
Régler	
la largeur des colonnes d'une table de déclaration	6-33
Règles	
alarme cyclique	2-19
alarme horaire	2-15
alarme temporisée	2-18
formation de multi-instances	6-42
saisie d'instructions CONT	6-48
saisie d'instructions LIST	6-64
saisie d'instructions LOG	6-58
Règles	
alarme de processus	2-20
pour l'exportation de tables des mnémoniques	6-13
utilisation des License Keys	1-14
Règles d'enfichage (S7-300)	5-7
Règles d'enfichage et autres règles (configuration matérielle)	5-6
Règles d'utilisation des License Keys	1-14
Règles particulières pour le module de simulation TOR SIM 374 IN/OUT 16	5-8
Règles particulières pour le module fictif DM 370 Dummy	5-8
Règles pour disposer des modules (ET 200S) ..	5-9
Règles pour disposer des modules (ET 200X) ..	5-9
Règles pour disposer des modules (SIMATIC 300)	5-7
Réinitialisation	
de valeurs en leur substituant leur valeur initiale	6-75
Rémanence	
après une coupure secteur	A-4
Remarques générales	
sur la saisie de mnémoniques	6-10
Remarques sur la modification du contenu des registres	6-97
Remplacement	

adresses ou paramètres dans les éléments	
CONT	6-52
éléments CONT	6-52
éléments LOG	6-61
Remplacement de modules	13-1
Remplacer un module	5-15
Renommer	
objets	3-13
Renseignements sur les modules	11-7
Représentation	
éléments CONT	6-47
éléments LOG	6-57
Représentation	
mnémoniques globaux et mnémoniques locaux	6-3
Représentation des nombres	
bit A-26	
date et heure (DATE AND TIME)	A-38
double mot	A-26
entier (16 bits)	A-28
entier (32 bits)	A-28
mot	A-26
nombres à virgule flottante	A-29, A-30, A-31
nombres décimaux codés binaire (DCB)	A-34
octet	A-26
S5 TIME	A-35
Représentation d'interfaces et de cartouches	
interface	5-11
Réseaux	6-21
insertion	6-53
terminaison en CONT	6-48
Réseaux	
commentaires	6-45, 6-46
titres	6-45
Réseaux CONT	
création de branches parallèles	6-54
Réseaux CONT	
création de branches	6-57
création de nouvelles branches	6-54
création d'une branche fermée	6-55
décomposition d'une jonction	6-56
ouverture de branches parallèles fermées ..	6-56
sélection	6-53
Réseaux LOG	
effacer et insérer une connexion	6-63
Réseaux LOG	
création de branches	6-62
insérer une connexion	6-62
sélection	6-61
Restaurer	
disposition des fenêtres	3-10
Retardement	
événements déclencheurs	2-25
RPL_VAL	11-26, 11-27
RUN ("Marche")	A-9

S

S5 TIME	
base de temps	A-35
format	A-35
plage	A-27
S5TIME	
type de données	A-35
S7-31xC	8-1

S7-PLCSIM.....	10-1	Session	
Saisie		enregistrer les paramètres.....	3-9
adresses ou paramètres dans des		SET_CLK	2-16, A-114
éléments CONT	6-51	SET_RTM.....	A-115
éléments de données du type STRUCT		SET_TINT	2-16
dans la table de déclaration.....	6-38	Setup	
mnémoniques globaux dans		paramétrage de la carte mémoire	1-16
un programme	6-45	saisie du numéro d'identification.....	1-16
multi-instance dans la table de déclaration		système de fichiers flash	1-16
des variables	6-42	Seuils de courant.....	5-16
opérandes ou mnémoniques		SFB	2-32, 2-33, A-37
dans une table de variables.....	10-6	SFB20 STOP.....	2-10
type de données ARRAY dans la table		SFC	2-33
de déclaration	6-39	utilisation	A-17
types de données simples dans la table		SFC 0 SET_CLK	2-15
de déclaration	6-38	SFC 51 RDSYSST	11-18
Saisie		SFC 55 WR_PARM.....	A-112
adresses ou paramètres	6-60	SFC 56 WR_DPARM	A-112
commentaires dans les instructions LIST	6-66	SFC 57 PARM_MOD	A-112
commentaires de blocs et de réseaux	6-46	SFC0 SET_CLK	A-114
de mnémoniques globaux individuels		SFC1 READ_CLK	A-114
dans les boîtes de dialogue	6-11	SFC2 SET_RTM	A-114
éléments CONT	6-51	SFC20 BLKMOV	A-13
éléments LOG	6-60	SFC22 CREAT_DB	A-13
instructions LIST	6-65	SFC26 UPDAT_PI.....	2-10, A-17
mnémoniques	6-12	SFC27 UPDAT_PO.....	2-10, A-17
structure de blocs de données associés		SFC28 SET_TINT	2-15
à un UDT	6-73	SFC29 CAN_TINT	2-15
structure de données de blocs de données		SFC3 CTRL_RTM.....	A-114
associés à un FB (DB d'instance).....	6-71	SFC30 ACT_TINT	2-15
structure de données de blocs de données		SFC31 QRY_TINT	2-15
globaux.....	6-71	SFC32 SRT_DINT.....	2-18
structure de types de données utilisateur		SFC36 MSK_FLT	2-25
(UDT).....	6-73	SFC36 MSK_FLT	
Saisie dans les boîtes de dialogue	3-6	exemple dans CONT	A-72
Saisie de plusieurs mnémoniques globaux		exemple dans LIST.....	A-72
dans la table des mnémoniques	6-12	SFC37 DMSK_FLT	2-25
Saisie d'opérandes ou de mnémoniques		SFC37 DMSK_FLT	
dans une table de forçage	10-8	exemple dans CONT	A-72
Saisie du numéro d'identification	1-16	exemple dans LIST.....	A-72
Saut		SFC38 READ_ERR	
de la liste des références croisées		exemple dans CONT	A-72
à la partie de programme.....	6-89	exemple dans LIST.....	A-72
de la structure du programme à la partie		SFC39 DIS_IRT	2-25
de programme	6-89	SFC39 DIS_IRT	
Schéma à contacts	6-20	exemple dans LIST.....	A-75
SDB (données système).....	A-110	SFC4 READ_RTM	A-114
sdf.....	8-6, 8-7	SFC40 EN_IRT	2-25
Section des instructions.....	6-30, 6-35	SFC40 EN_IRT	
en LIST	6-34	exemple dans LIST.....	A-75
Section des instructions		SFC41 DIS_AIRT	2-25
édition	6-43	SFC41 DIS_AIRT	
fonction de recherche d'erreurs	6-47	exemple dans LIST.....	A-76
structure.....	6-43	SFC42 EN_AIRT	2-25
Sélection		SFC42 EN_AIRT	
langage de programmation.....	6-19	exemple dans LIST.....	A-76
Sélection		SFC44 RPL_VAL	11-26
dans les réseaux CONT	6-53	SFC46 STP	2-10
dans les réseaux LOG	6-61	SFC48 SNC_RTCB.....	A-114
lignes de mnémonique	6-16	SFC51 RDSYSST	11-17, A-21
mode de tri dans la table des mnémoniques	6-16	SFC52 WR_USMSG	11-20
zones de texte dans les instructions LIST ...	6-65	SFC55 WR_PARM.....	A-110
Sélection d'un type de station.....	5-10	SFC56 WR_DPARM	A-110
Sélectionner des lignes dans la table de		SFC57 PARM_MOD	A-110
configuration	5-15	SIM 374 IN/OUT 16.....	5-8

Simples
 types de données A-27
 SNC_RTCB A-114
 Solution aux erreurs
 programme-exemple 11-23
 Sommaire 12-3
 Sorties
 mémoire image A-17, A-18
 Sorties de périphérie
 forçage à l'état "Arrêt" de la CPU 10-23
 Sorties utilisées 6-79
 Soudure froide 5-9
 Sources 8-6, 8-7
 externes 4-4
 Spécification de modules 9-11
 SRT_DINT 2-18
 Station
 paramétrage 5-10
 STEP 7 8-1
 STEP 7 Lite 1-1, 1-2, 1-3, 1-4
 désinstallation 1-19
 erreur durant l'installation 1-16
 installation 1-15
 lancement du logiciel 3-1
 STEP 7 Lite
 OB d'erreur
 réaction aux erreurs 2-24
 STEP 7 Lite (fenêtre du projet) 1-5
 STEP7 8-6
 STOP ("Arrêt") A-4
 STRING A-37
 STRUCT A-37
 Structure 8-10
 alarme horaire A-96, A-97
 alarme temporisée A-104
 Structure
 création A-46
 mémoire de chargement A-13, A-14
 section des instructions 6-43
 table de déclaration des variables 6-36, 6-37
 UDT 6-22
 Structure du fichier d'exportation 8-10
 Structure du programme 1-5, 6-76, 6-81, 6-82
 Structure et éléments de la table
 des mnémoniques 6-5
 Style des caractères 12-1, 12-8
 Support de données 8-5
 Suppression
 variables dans une table de déclaration 6-40
 Suppression
 lignes de mnémonique 6-14
 Suppression de certains blocs dans la CPU 9-13
 Suppression d'une langue 8-13
 Supprimer
 objets 3-13
 Surveillance du processus 10-2
 Symboles
 dans la structure du programme 6-81, 6-82,
 6-83, 6-84, 6-85
 différence 11-4
 égalité 11-4
 Symbolique 6-3
 Synchronisation
 état 11-4
 horloge A-114
 Système de fichiers flash 1-16

Système d'exploitation 1-15
 Système d'exploitation
 tâches 2-1
 Système d'exploitation de la CPU 2-10
 Systèmes intégrés compacts C7
 configurer 5-12
 SZL 11-18, 11-19

T

Table de configuration 5-5
 Table de déclaration
 copie de variables 6-40
 saisie de types de données simples 6-38
 saisie d'éléments de données
 du type STRUCT 6-38
 saisie du type de données ARRAY 6-39
 suppression de variables 6-40
 Table de déclaration
 régler la largeur des colonnes 6-33
 Table de déclaration des variables 6-30, 6-35
 pour l'OB 81 11-23
 saisie de multi-instance 6-42
 tâche 6-34
 Table de déclaration des variables
 FB pour l'exemple d'un processus
 de mélange industriel A-84
 FC pour l'exemple d'un processus
 de mélange industriel A-88
 OB pour l'exemple d'un processus
 de mélange industriel A-90
 structure 6-36
 Table de variables
 coller le contenu du presse-papiers 10-16
 copie 10-4
 création et ouverture 10-3
 édition 10-6
 exemple 10-6, 10-7
 saisie d'opérandes ou de mnémoniques 10-6
 taille maximale 10-7
 vérification de la syntaxe 10-7
 Table des mnémoniques 1-6, 1-7, 6-3, 6-7, 8-6, 8-7
 exportation 6-13
 ouvrir 6-13
 Table des mnémoniques
 filtrage 6-15
 format de données pour l'importation/
 exportation 8-8
 importation 6-13
 ouverture 6-12
 pour mnémoniques globaux 6-5
 structure et éléments 6-5
 tri 6-16
 Table des variables
 enregistrer 10-1
 utilisation 10-1
 Table des variables
 insertion d'une plage d'opérandes continue 10-9
 Tableau
 actions possibles du démarreur de moteur 5-18
 Tableau (type de données ARRAY)
 description A-43
 nombre de niveaux imbriqués A-42
 Tableaux
 création A-44

- utilisation pour l'accès aux données A-43
- Tables de déclaration de variables
 - insertion de lignes vides 6-38
 - modifier la largeur des colonnes 6-40
- Tâche de forçage permanent..... 10-24
- Tâche de forçage permanent
 - annulation..... 10-26
- Taille de l'affichage dans une fenêtre 6-17
- Taille des caractères 12-1, 12-8
- Téléservice 3-19
- Temporisation (T)
 - zone de mémoire
 - rémanente A-23
- Temporisations
 - limites supérieures pour la saisie..... 10-10
- Temporisations (T) A-115
- Temporisations utilisées 6-79
- Temps de blocage 5-16
- Temps de cycle 11-9
 - contrôle pour éviter les erreurs de temps .. 11-16
- Temps de cycle de l'OB1 2-10
- Temps de cycle maximal 2-10
- Temps de cycle minimal 2-11, 2-12
- Temps de récupération..... 5-17
- Temps de surveillance..... 2-22
- Temps de surveillance du cycle..... 2-10
- Test
 - avec la table des variables 10-1
 - avec la visualisation d'état du programme. 10-27
 - choix du mode de fonctionnement..... 10-33
 - présentation..... 10-1
- Test
 - activation et désactivation avec état du
 - programme 10-34
- Test à l'aide de la table des variables..... 13-1
- Texte-exemple..... 12-7
- Thermocouple..... 5-9
- TIMER A-50
- TIMER
 - type de paramètre A-50
- Titres de blocs 6-45
- Titres de réseaux..... 6-45
- Traduction de textes à gestion multilingue 8-12
- Traitement de programme
 - déclenché par alarme 2-14
- Traitement d'erreurs 11-21
- Traitement différé d'événements d'alarme et
 - d'événements asynchrones
 - exemple A-76
- Traitement du programme
 - cyclique..... 2-3, 2-5, 2-6
 - déclenché par alarme 2-3
- Traiter
 - projet 4-6
- Trajet du courant 6-50
- Transmission au paramètre IN_OUT d'un FB.. A-70
- Transmission de paramètres
 - enregistrement des valeurs transmises 2-26
- Transmission de paramètres
 - conception de paramètres pour un
 - programme structuré A-84
 - exemple de FB pour l'exemple d'un
 - processus de mélange industriel A-84
 - types de paramètre..... A-50
- Transmission d'informations de diagnostic 11-17
- Travailler sur la PG/le PC sans projet original . 13-2
- Tri
 - liste des références croisées 6-77, 6-78
- Tri
 - mnémoniques 6-12
 - table des mnémoniques 6-16
- Type de déclaration
 - modification 6-36
- Type de données
 - DATE AND TIME\
 - Date et heure..... A-38
- Type de données
 - ARRAY 6-39
 - STRUCT 6-38
- Type de données
 - ARRAY A-41
 - DWORD A-34
 - S5 TIME A-35
 - STRING A-39, A-40
 - STRUCT A-41
 - UDT 6-22
 - utilisateur 6-22
 - WORD A-34
- Type de données ARRAY
 - saisie dans la table de déclaration 6-39
- Type de données structure (STRUCT)
 - description A-46
 - nombre de niveaux imbriqués A-42
- Type de données utilisateur
 - création..... A-48, A-49
- Type de paramètre
 - ANY A-50
 - BLOCK_DB A-50
 - BLOCK_FB A-50
 - BLOCK_FC A-50
 - BLOCK_SDB A-50
 - COUNTER A-50
 - POINTER A-50
 - TIMER A-50
- Type de paramètre POINTER
 - utilisation A-53
- Type de station
 - sélection 5-10
- Types d'alarme 2-3
- Types de données
 - FB
 - SFB 2-26
- Types de données
 - affectation aux données locales de blocs
 - de code..... A-63
 - ARRAY A-37
 - BOOL A-26, A-27
 - caractère (CHAR) A-27
 - date A-27
 - DATE_AND_TIME A-37
 - définis par l'utilisateur A-37
 - description A-27
 - DINT
 - entier (32 bits) A-28
 - double mot..... A-26
 - double mot (DWORD) A-27
 - durée (TIME) A-27
 - FB
 - SFB A-37
 - heure (TIME OF DAY) A-27

INT	
entier (16 bits).....	A-28
introduction.....	A-26
mot	A-26
mot (WORD).....	A-27
nombre entier (16 bits) (INT)	A-27
nombre entier (32 bits) (DINT).....	A-27
nombre réel (REAL).....	A-27
octet.....	A-26, A-27
REAL\	
nombre à virgule flottante	A-29, A-30
S5 TIME.....	A-27
STRING	A-37
STRUCT	A-37
type de paramètre	
ANY	
paramètre	A-60, A-61, A-62
UDT	A-37
Types de données (simples)	
saisie dans la table de déclaration.....	6-38
Types de données autorisés pour la	
transmission de paramètres	A-65
Types de données complexes	A-37
Types de données simples	A-27
Types de données structurés	A-37, A-42
Types de données structurés	
structure.....	A-46
structure	
structures d'imbrication et tableaux	A-42
tableau.....	A-43, A-44, A-45
tableau	
structures d'imbrication et tableaux	A-42
Types de données utilisateur	
description	A-48
saisie de la structure.....	6-73
Types de données utilisateur (UDT)	6-22
Types de licences.....	1-10
Enterprise License	1-10
Floating License	1-12
Rental License.....	1-10
Trial License	1-12
Upgrade License	1-12
Types de paramètre	A-50
Types de paramètre	
ANY	A-57, A-60, A-61, A-62
introduction.....	A-26
Types de texte à gestion multilingue	8-10

U

UDT	6-22, A-37
UDT	
saisie de la structure.....	6-73
UPDAT_PI.....	2-10, A-17
UPDAT_PO	2-10, A-17
Utilisation	
mémentos de cadence et temporisations ..	A-115
Utilisation	
de structures pour l'accès aux données	A-46
de tableaux pour l'accès aux données.....	A-43
de types de données utilisateur	
pour l'accès aux données	A-48
type de paramètre ANY	A-60
types de données complexes	A-42
zones de mémoire système.....	A-15

Utilisation de la déclaration des variables	
dans les blocs de code	6-34
Utilisation de multi-instances	6-41
Utilisation de SFC.....	A-17
Utilisation du clavier	3-14
Utilisation du type de paramètre POINTER	A-53
Utiliser une carte mémoire Micro	
comme support de données	8-5

V

Valeur de remplacement	
utilisation de la SFC44 (RPL_VAL)	11-26
Valeur en cours	6-70
Valeur initiale.....	6-70, 6-75
Valeurs	
modification dans la vue des données	
de blocs de données	6-74
réinitialisation en leur substituant leur	
valeur initiale	6-75
Validation d'événements d'alarme	
et d'événements asynchrones	
exemple.....	A-75
Variables	10-20
copie dans la table de déclaration	6-40
forçage	10-20, 10-22, 10-23, 10-34
forçage permanent	10-25
suppression dans une table de déclaration ..	6-40
visualisation.....	10-17, 10-18
Vérification de cohérence	5-20
Vérification de la cohérence du programme	6-90
Vérifier la cohérence de la configuration	
d'une station	9-7
Visualisation	
définition du mode	10-17
marche à suivre.....	10-2
Visualisation de variables	
avec déclenchement défini	10-18
introduction.....	10-17
unique et immédiate	10-19
Visualisation d'état de programme	
affichage.....	10-28
Visualisation d'état du programme	10-27
Visualisation/Forçage	1-5
Volume d'informations selon le type	
de module dans l'état du module.....	11-11
Vue "Configuration HW"	5-4
Vue 'Comparaison HW'	11-4
Vue d'ensemble du projet.....	3-3, 4-1
Vue des déclarations.....	6-68
bloc de données	6-69
Vue des données	6-68, 6-74
bloc de données	6-70
Vue 'Diagnostic HW'	11-6
Vue en ligne sur la CPU	1-5
Vue synoptique des adresses	6-77
Vues	1-5, 1-6, 1-7, 1-8

W

Windows	1-15
WORD	
type de données	A-34
WR_DPARM.....	A-111, A-113
WR_PARM	A-111, A-113
WR_USMSG	11-20

Z

Zone de liste	3-6
Zone de texte.....	3-6
Zone de travail.....	3-3, 3-4
pour la configuration	5-2
Zone EPROM	A-13
Zone RAM	A-13

Zones de données locales.....	5-6
Zones de mémoire	
mémoire de chargement.....	A-12
mémoire de travail	A-12
mémoire système	A-12
Zones de mémoire	
mémoire rémanente	A-23
zones d'adresses.....	A-16
Zones de mémoire rémanentes	
des CPU S7-300	A-23
Zones d'opérandes	
description	A-15, A-16
Zones sélectionnées	
copie dans le presse-papiers.....	10-16
couper et copier dans le presse-papiers.....	10-16
Zoom	
affichage	6-33

