

Chapitre II: Application API S7-200

II. 1) Introduction

La famille S7-200 est constituée de micro-automates programmables utilisables dans des applications d'automatisation variées. Sa forme compacte, son faible prix et son important jeu d'opérations en font une solution idéale pour la commande de petites applications. En outre, le large choix de modèles S7-200 ainsi que l'outil de programmation se basant sur Windows vous offrent la souplesse nécessaire pour résoudre vos problèmes d'automatisation.

La famille S7-200 est constituée de micro-automates programmables pouvant commander une large gamme d'appareils afin de répondre à vos besoins en matière d'automatisation.

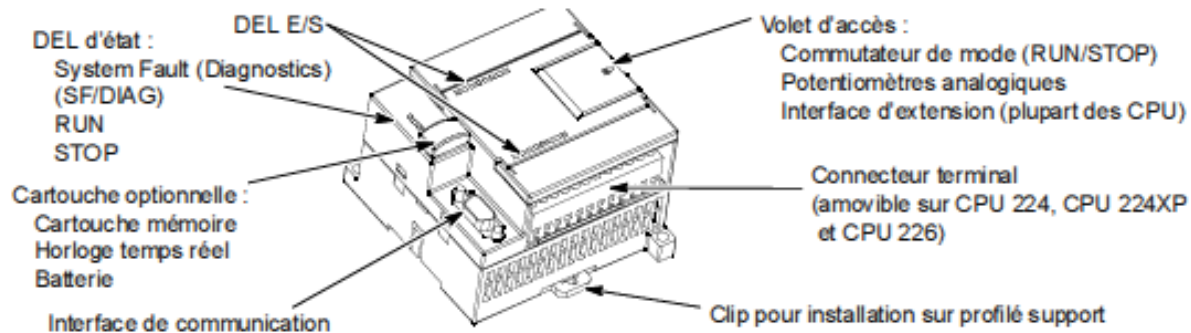
Le S7-200 surveille les entrées et modifie les sorties conformément au programme utilisateur, qui peut contenir des opérations booléennes, des opérations de comptage, des opérations de temporisation, des opérations arithmétiques complexes et des opérations de communication avec d'autres unités intelligentes. Sa forme compacte, sa configuration souple et son important jeu d'opérations en font une solution idéale pour la commande d'applications très variées.

La série S7-200 comprend les CPU suivantes :

CPU 221 CC/CC/CC 6 entrées/4 sorties; CPU 221 CA/CC/Relais 6 entrées/4 relais; CPU 222 CC/CC/CC 8 entrées/6 sorties; CPU 222 CA/CC/Relais 8 entrées/6 relais; CPU 224 CC/CC/CC 14 entrées/10 sorties; CPU 224 CA/CC/Relais 14 entrées/10 relais; CPU 224XP CC/CC/CC 14 entrées/10 sorties; CPU 224XP CA/CC/Relais 14 entrées/10 relais; CPU 226 CC/CC/CC 24 entrées/16 sorties; CPU 226 CA/CC/Relais 24 entrées/16 relais.

II. 2) Architecture de la CPU S7-200

La CPU S7-200 combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et des circuits de sortie dans un boîtier compact afin de créer un puissant micro-automate. Une fois que vous avez chargé votre programme, le S7-200 contient la logique nécessaire au contrôle et à la commande d'unités d'entrée et de sortie dans votre application.



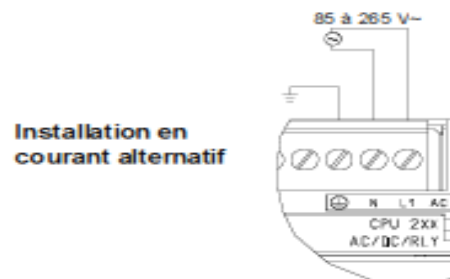
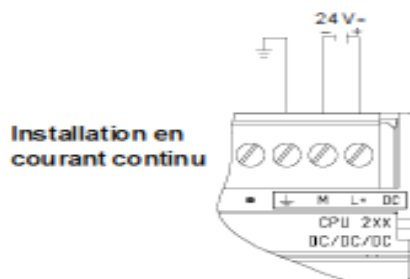
Siemens fournit des modèles de CPU S7-200 différents, disposant de divers éléments et fonctions afin de vous aider à créer des solutions efficaces pour vos applications variées. Le tableau ci-dessous compare brièvement certaines caractéristiques des différentes CPU.

Caractéristique	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Dimensions (mm)	90 x 80 x 62	90 x 80 x 62	120,5 x 80 x 62	140 x 80 x 62	190 x 80 x 62
Mém. de programme avec édition à l'état Marche sans édition à l'état Marche	4096 octets 4096 octets	4096 octets 4096 octets	8192 octets 12288 octets	12288 octets 16384 octets	16384 octets 24576 octets
Mémoire de données	2048 octets	2048 octets	8192 octets	10240 octets	10240 octets
Sauvegarde de la mémoire	50 heures, typique	50 heures, typique	100 heures, typique	100 heures, typique	100 heures, typique
E/S internes locales TOR Analogiques	6 E / 4 S -	8 E / 6 S -	14 E / 10 S -	14 E / 10 S 2 E / 1 S	24 E / 16 S -
Modules d'extension	0 module	2 modules ¹	7 modules ¹	7 modules ¹	7 modules ¹
Compteurs rapides Une phase Deux phases	4 à 30 kHz 2 à 20 kHz	4 à 30 kHz 2 à 20 kHz	6 à 30 kHz 4 à 20kHz	4 à 30 kHz 2 à 200 kHz 3 à 20 kHz 1 à 100 kHz	6 à 30 kHz 4 à 20 kHz
Sorties d'impuls.(CC)	2 à 20 kHz	2 à 20 kHz	2 à 20 kHz	2 à 100 kHz	2 à 20 kHz
Potentiomètres analogiques	1	1	2	2	2
Horloge temps réel	Cartouche	Cartouche	Intégrée	Intégrée	Intégrée
Interfaces de communication	1 RS-485	1 RS-485	1 RS-485	2 RS-485	2 RS-485
Virgule flottante	Oui				
Taille de la mémoire image d'E/S TOR	256 (128 entrées, 128 sorties)				
Vitesse d'exécution booléenne	0,22 microseconde/opération				

II. 2.1) Raccordement de l'alimentation à la CPU S7-200

La première étape consiste à raccorder le S7-200 à une source de courant. La figure ci-dessous montre les connexions de câblage pour un modèle en courant continu et un modèle en courant alternatif de la CPU S7-200.

Prenez toujours toutes les mesures de sécurité nécessaires et assurez-vous que l'alimentation de l'automate S7-200 est coupée avant de tenter d'installer ou de démonter le S7-200.



II. 2.2) Organisation de la mémoire de la CPU S7-200

II. 2.2.1) Sauvegarde et restauration des données par le S7-200

L'automate S7-200 fournit des fonctions variées assurant que votre programme utilisateur et vos données sont conservées correctement dans le S7-200.

A) Mémoire de données rémanente : Zones de mémoire de données que l'utilisateur sélectionne afin qu'elles restent inchangées en cas de mise hors tension puis sous tension, et ce tant que le super-condensateur et la cartouche pile optionnelle ne sont pas déchargés. V, M, valeurs en cours des temporisations et valeurs en cours des compteurs sont les seules zones de mémoire de données que vous pouvez configurer comme rémanentes.

B) Mémoire permanente : Mémoire non volatile utilisée pour stocker le bloc de code, le bloc de données, le bloc de données système, les valeurs forcées, les mémentos configurés pour être sauvegardés en cas de coupure de courant, ainsi que des valeurs sélectionnées écrites sous la commande du programme utilisateur.

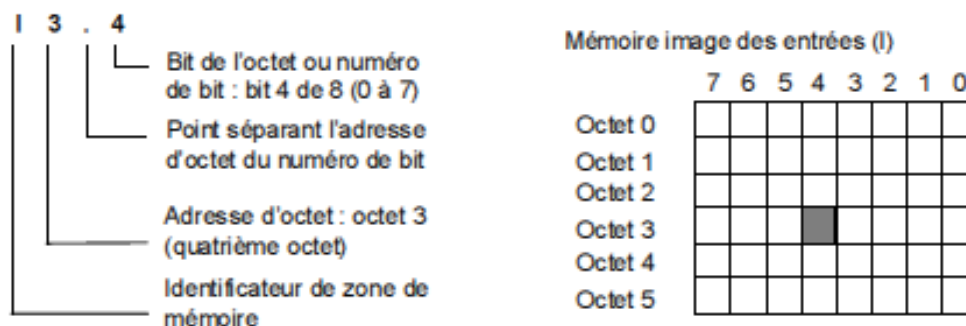
C) Cartouche mémoire : Mémoire non volatile débrochable servant à stocker le bloc de code, le bloc de données, le bloc de données système, des recettes, des journaux de données et des valeurs forcées.

Vous pouvez utiliser l'Explorateur S7-200 pour stocker des fichiers de documentation (doc, texte, pdf, etc.) dans la cartouche. Vous pouvez également vous en servir pour assurer la maintenance générale des fichiers dans la cartouche mémoire (copie, effacement, répertoire et lancement). Pour installer une cartouche mémoire, retirez de la CPU S7-200 le volet en plastique couvrant l'emplacement correspondant et insérez la cartouche mémoire dans l'emplacement. Cette dernière a une forme telle que vous ne pouvez pas commettre d'erreur à l'enfichage.

II. 2.2.2) Accès aux données du S7-200

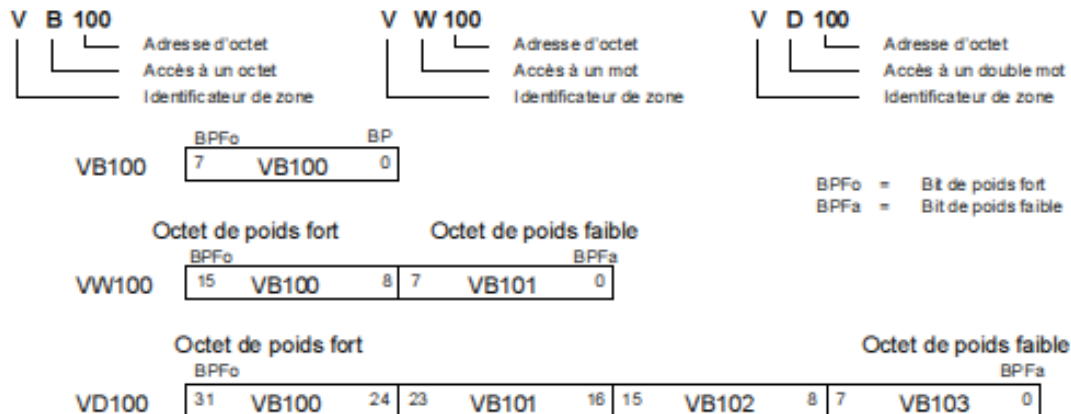
Le S7-200 range les informations à différents emplacements de la mémoire ayant chacun une adresse unique. Vous pouvez identifier explicitement l'adresse de mémoire à laquelle vous voulez accéder. Ainsi, votre programme dispose d'un accès direct aux informations.

Pour accéder à un bit dans une zone de mémoire, vous devez préciser son adresse composée d'un identificateur de zone de mémoire, de l'adresse d'octet et du numéro de bit. La figure ci-dessous montre comment accéder à un bit par la méthode appelée adressage "*octet.bit*". Dans cet exemple, la zone de mémoire et l'adresse d'octet (I = entrée et 3 = octet 3) sont suivies d'un point les séparant de l'adresse de bit (bit 4).



Le format d'adresse d'octet permet d'accéder à des données dans la plupart des zones de mémoire (**V**, **I**, **Q**, **M**, **S**, **L** et **SM**) sous forme d'octets, de mots ou de doubles mots. Pour accéder à un octet, un mot ou un double mot de données dans la mémoire, vous précisez l'adresse de manière comparable à l'adresse d'un bit. Cette adresse est composée d'un identificateur de zone, de la désignation de la taille des données et de l'adresse d'octet de départ de la valeur d'octet, de mot ou de double mot.

On accède aux données situées dans d'autres zones de mémoire (**T**, **C**, **HC** et **accumulateurs**, par exemple) via un format d'adresse comprenant l'identificateur de zone et le numéro de l'élément en question.



II. 2.2.3) Accès aux données dans les zones de mémoire

A) Mémoire image des entrées : I

Le S7-200 lit les entrées physiques au début de chaque cycle et écrit ces valeurs dans la mémoire image des entrées. Vous pouvez accéder à la mémoire image des entrées par bits, octets, mots ou doubles mots.

Bit : I[adresse d'octet].[adresse de bit] I0.1
Octet, mot ou double mot : I[taille][adresse d'octet de départ] IB4

B) Mémoire image des sorties : Q

A la fin du cycle, le S7-200 copie dans les sorties physiques les valeurs contenues dans la mémoire image des sorties. Vous pouvez accéder à la mémoire image des sorties par bits, octets, mots ou doubles mots.

Bit : Q[adresse d'octet].[adresse de bit] Q1.1
Octet, mot ou double mot : Q[taille][adresse d'octet de départ] QB5

C) Mémoire des variables : V

Vous pouvez vous servir de la mémoire des variables (mémoire V) pour sauvegarder des résultats intermédiaires d'opérations exécutées par la logique de commande dans votre programme. Vous pouvez également vous en servir pour la sauvegarde d'autres données en rapport avec votre processus ou votre tâche. Vous pouvez accéder à la mémoire V par bits, octets, mots ou doubles mots.

Bit : V[adresse d'octet].[adresse de bit] V10.2
Octet, mot ou double mot : V[taille][adresse d'octet de départ] VW100

D) Mémentos : M

Les mémentos internes (mémoire M) - ou relais de commande - fournissent de l'espace mémoire pour l'état intermédiaire d'une opération ou d'autres informations de commande. Vous pouvez accéder à la zone des mémentos par bits, octets, mots ou doubles mots.

Bit : M[adresse d'octet].[adresse de bit] M26.7
Octet, mot ou double mot : M[taille][adresse d'octet de départ] MD20

E) Temporisations : T

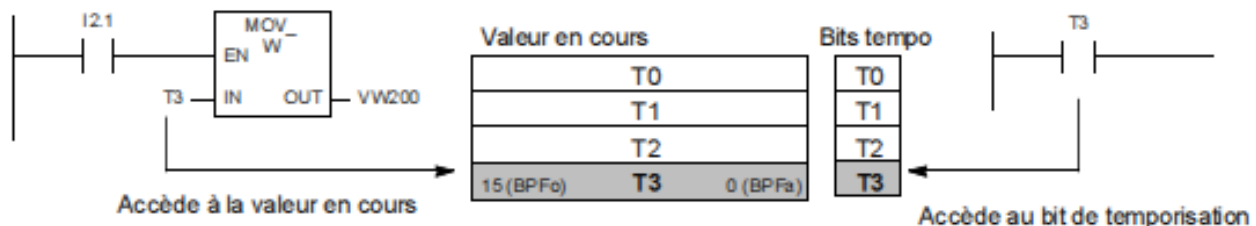
Le S7-200 fournit des temporisations qui comptent des incréments de temps selon des résolutions (incrément de base de temps) égales à 1 ms, 10 ms ou 100 ms. Deux variables sont associées à chaque temporisation :

- **la valeur en cours** : ce nombre entier signé de 16 bits contient la durée comptabilisée par la temporisation ;

- **le bit de temporisation** : ce bit est mis à 1 ou à 0 selon le résultat de la comparaison entre valeur en cours et valeur prédéfinie, cette dernière étant indiquée dans l'opération de temporisation.

Vous accédez à ces deux variables à l'aide de l'adresse de la temporisation (T + numéro de la temporisation). Selon l'opération utilisée, l'accès se fait au bit de temporisation ou à la valeur en cours : les opérations avec des bits comme opérandes accèdent au bit de temporisation alors que les opérations ayant des mots comme opérandes accèdent à la valeur en cours. Comme le montre la figure ci-dessous, le contact à fermeture accède au bit de temporisation et l'opération Transférer mot à la valeur en cours de la temporisation.

Format : T[numéro de la temporisation] T24



F) Compteurs : C

Le S7-200 fournit trois types de compteurs qui comptent chaque transition du niveau bas au niveau haut aux entrées de comptage : les compteurs qui incrémentent, ceux qui décrémentent et ceux pouvant à la fois incrémenter et décrémenter. Deux variables sont associées à chaque compteur :

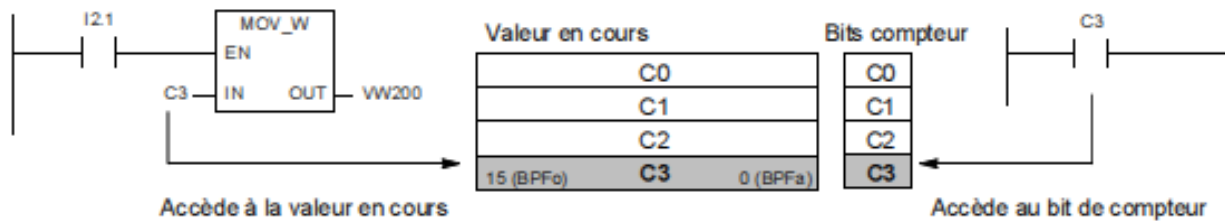
- **la valeur en cours** : ce nombre entier signé de 16 bits contient le total cumulé ;

- **le bit de compteur** : ce bit est mis à 1 ou à 0 selon le résultat de la comparaison entre valeur en cours et valeur prédéfinie, cette dernière étant indiquée dans l'opération de comptage.

Vous accédez à ces deux variables à l'aide de l'adresse du compteur (C + numéro du compteur).

Selon l'opération utilisée, l'accès se fait au bit de compteur ou à la valeur en cours : les opérations avec des bits comme opérandes accèdent au bit de compteur alors que les opérations ayant des mots comme opérandes accèdent à la valeur en cours. Comme le montre la figure ci-dessous, le contact à fermeture accède au bit de compteur et l'opération Transférer mot à la valeur en cours du compteur.

Format : C[numéro du compteur] C24



G) Compteurs rapides : HC

Les compteurs rapides comptent des événements très rapides indépendamment du cycle de la CPU. Ils ont une valeur de comptage (ou valeur en cours) entière signée de 32 bits. Pour accéder à la valeur de comptage d'un compteur rapide, vous indiquez l'adresse du compteur rapide, comprenant l'identificateur de zone HC et le numéro du compteur (HC0, par exemple). Vous ne pouvez accéder qu'en lecture à la valeur en cours des compteurs rapides et qu'en format de double mot (32 bits).

Format : HC[numéro du compteur rapide] HC1

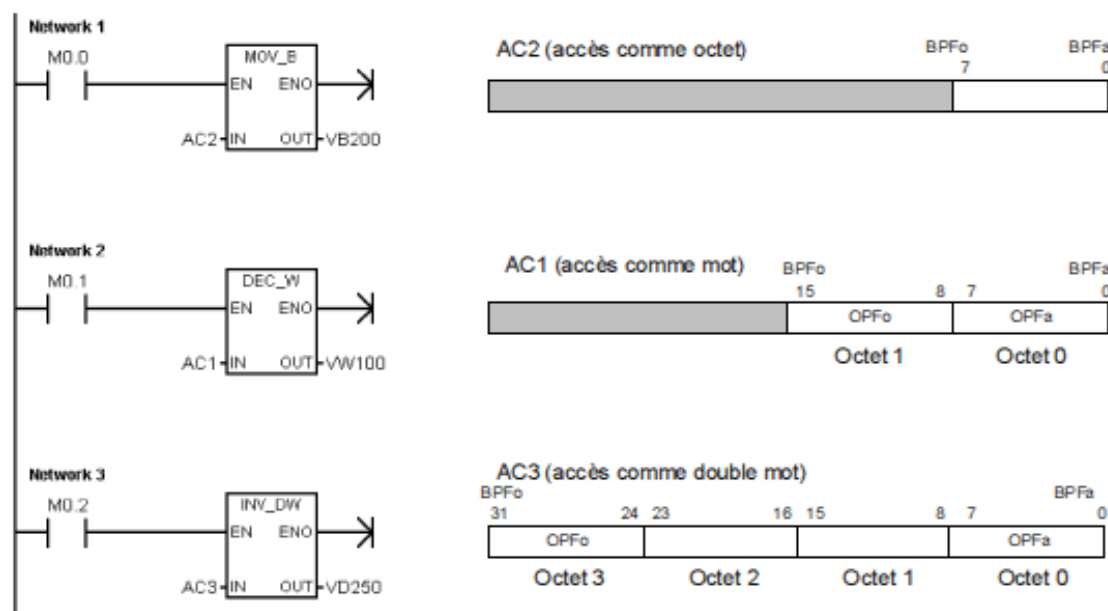
H) Accumulateurs : AC

Les accumulateurs sont des éléments en lecture/écriture pouvant être utilisés comme mémoire. Vous pouvez, par exemple, vous servir des accumulateurs pour transmettre des paramètres à ou depuis des sous-programmes et pour sauvegarder des valeurs intermédiaires utilisées dans un calcul. Le S7-200 dispose de quatre accumulateurs de 32 bits : AC0, AC1, AC2 et AC3. Vous pouvez accéder aux données contenues dans les accumulateurs par octets, mots ou doubles mots.

C'est l'opération utilisée pour accéder à l'accumulateur qui détermine la taille des données en accès. Comme le montre la figure ci-dessous, lorsque vous accédez aux accumulateurs en format d'octet

ou de mot, vous n'utilisez, respectivement, que les 8 et 16 bits de poids faible de cet accumulateur ; quand vous y accédez par doubles mots, vous faites appel à l'intégralité des 32 bits.

Format : AC[numéro de l'accumulateur] AC0



I) Mémentos spéciaux : SM

Les mémentos spéciaux permettent l'échange d'informations entre la CPU et votre programme. Vous pouvez faire appel à ces bits pour sélectionner et commander certaines fonctions spéciales de la CPU S7-200, telles que : bit mis à 1 lors du premier cycle, bit commutant à une fréquence fixe ou bit signalant l'état d'instructions arithmétiques ou de fonctionnement. Vous pouvez accéder aux mémentos spéciaux par bits, octets, mots ou doubles mots.

Bit :	SM[adresse d'octet].[adresse de bit]	SM0.1
Octet, mot ou double mot :	SM[taille][adresse d'octet de départ]	SMB86

J) Mémoire locale : L

L'automate programmable S7-200 fournit 64 octets de mémoire locale (L) dont 60 peuvent être utilisés comme mémoire intermédiaire ou pour la transmission de paramètres formels aux sous-programmes.

La mémoire L est similaire à la mémoire V à une exception majeure près. En effet, la mémoire V concerne les variables globales alors que la mémoire L concerne les données locales. Global signifie qu'il est possible d'accéder à la même adresse de mémoire à partir de n'importe quelle entité de programme (programme principal, sous-programmes ou programmes d'interruption). Local signifie que l'affectation de mémoire est associée à une entité spécifique du programme. Le S7-200 alloue 64 octets de mémoire L au programme principal, 64 octets à chaque niveau d'imbrication de sous-programme et 64 octets aux programmes d'interruption.

Il est impossible d'accéder à la mémoire L allouée au programme principal à partir des sous-programmes ou des programmes d'interruption. Un sous-programme ne peut pas accéder la mémoire L affectée au programme principal, à un programme d'interruption ou à un autre sous-programme. De même, un programme d'interruption ne peut pas accéder à la mémoire L affectée au programme principal ou à un sous-programme.

L'automate S7-200 affecte la mémoire L selon les besoins. Ainsi, il n'y a pas d'affectation de mémoire L aux sous-programmes et programmes d'interruption pendant que le programme principal est en cours d'exécution. L'affectation de mémoire locale se fait, selon les besoins, à l'apparition d'une interruption ou à l'appel d'un sous-programme. La nouvelle affectation de mémoire L peut réutiliser les mêmes emplacements de mémoire L qu'un sous-programme ou un programme d'interruption différent.

La mémoire L n'est pas initialisée par le S7-200 lors de son affectation et elle peut donc contenir n'importe quelle valeur. Lorsque vous transmettez des paramètres formels lors d'un appel de sous-programme, les valeurs des paramètres transmis sont placées par le S7-200 aux adresses appropriées de la mémoire L affectée à ce sous-programme. Les adresses de mémoire L ne recevant pas de valeur lors de l'étape de transmission des paramètres formels ne sont pas initialisées et peuvent donc contenir n'importe quelle valeur au moment de l'affectation.

Bit :	L[adresse d'octet].[adresse de bit]	L0.0
Octet, mot, double mot :	L[taille][adresse d'octet de départ]	LB33

K) Entrées analogiques : AI

Le S7-200 convertit une valeur analogique (telle que la température ou la tension) en valeur numérique de 16 bits (un mot). Vous accédez à ces valeurs par l'identificateur de zone (AI), la taille des données (W) et l'adresse d'octet de départ. Comme les entrées analogiques sont des mots et commencent toujours sur des octets pairs (tels que 0, 2 ou 4), vous y accédez par des

adresses d'octet paires (AIW0, AIW2, AIW4, par exemple). Seule la lecture des entrées analogiques est possible.

Format : AIW[adresse d'octet de départ] AIW4

L) Sorties analogiques : AQ

Le S7-200 convertit une valeur numérique de 16 bits (mot) en un courant ou une tension proportionnelle à la valeur numérique. Vous écrivez ces valeurs via l'identificateur de zone (AQ), la taille des données (W) et l'adresse d'octet de départ. Comme les sorties analogiques sont des mots et commencent toujours sur des octets pairs (tels que 0, 2 ou 4), vous y accédez par des adresses d'octet paires (AQW0, AQW2 ou AQW4, par exemple). Seule l'écriture des sorties analogiques est possible.

Format : AQW[adresse d'octet de départ] AQW4

M) Relais séquentiels (SCR) : S

On utilise les relais séquentiels (SCR) ou bits S pour organiser des étapes ou fonctionnements de l'installation en segments de programme équivalents. Ces relais permettent la segmentation logique du programme de commande. Vous pouvez accéder aux relais séquentiels par bits, octets, mots ou doubles mots.

Bit : S[adresse d'octet].[adresse de bit] S3.1

Octet, mot ou double mot : S[taille][adresse d'octet de départ] SB4

II. 2.2.4) Plages de mémoire et fonctions du S7-200

Le tableau ci-dessous donne les plages de mémoire et fonctions des CPU S7-200 :

Description	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Taille du programme utilisateur avec édition à l'état Marche sans édition à l'état Marche	4096 octets 4096 octets	4096 octets 4096 octets	8192 octets 12288 octets	12288 octets 16384 octets	16384 octets 24576 octets
Taille des données utilisateur	2048 octets	2048 octets	8192 octets	10240 octets	10240 octets
Mémoire image des entrées	I0.0 à I15.7	I0.0 à I15.7	I0.0 à I15.7	I0.0 à I15.7	I0.0 à I15.7
Mémoire image des sorties	Q0.0 à Q15.7	Q0.0 à Q15.7	Q0.0 à Q15.7	Q0.0 à Q15.7	Q0.0 à Q15.7
Entrées analogiques (lecture seule)	AIW0 à AIW30	AIW0 à AIW30	AIW0 à AIW62	AIW0 à AIW62	AIW0 à AIW62
Sorties analogiques (écriture seule)	AQW0 à AQW30	AQW0 à AQW30	AQW0 à AQW62	AQW0 à AQW62	AQW0 à AQW62
Mémoire des variables (V)	VB0 à VB2047	VB0 à VB2047	VB0 à VB8191	VB0 à VB10239	VB0 à VB10239
Mémoire locale (L) ¹	LB0 à LB63	LB0 à LB63	LB0 à LB63	LB0 à LB63	LB0 à LB63
Mémoires (M)	M0.0 à M31.7	M0.0 à M31.7	M0.0 à M31.7	M0.0 à M31.7	M0.0 à M31.7
Mémoires spéciales (SM)	SM0.0 à SM179.7	SM0.0 à SM299.7	SM0.0 à SM549.7	SM0.0 à SM549.7	SM0.0 à SM549.7
Lecture seule	SM0.0 à SM29.7	SM0.0 à SM29.7	SM0.0 à SM29.7	SM0.0 à SM29.7	SM0.0 à SM29.7
Temporisations	256 (T0 à T255)	256 (T0 à T255)	256 (T0 à T255)	256 (T0 à T255)	256 (T0 à T255)
Retard à la montée mémorisé : 1 ms	T0, T64 T1 à T4 et T65 à T68	T0, T64 T1 à T4 et T65 à T68	T0, T64 T1 à T4 et T65 à T68	T0, T64 T1 à T4 et T65 à T68	T0, T64 T1 à T4 et T65 à T68
100 ms	T5 à T31 et T69 à T95	T5 à T31 et T69 à T95	T5 à T31 et T69 à T95	T5 à T31 et T69 à T95	T5 à T31 et T69 à T95

N.B : La suite du tableau est donnée à la page suivante.

Description	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Retard à la montée/ Retard à la retombée	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
1 ms	T33 à T36 et T97 à T100	T33 à T36 et T97 à T100	T33 à T36 et T97 à T100	T33 à T36 et T97 à T100	T33 à T36 et T97 à T100
10 ms	T37 à T63 et T101 à T255	T37 à T63 et T101 à T255	T37 à T63 et T101 à T255	T37 à T63 et T101 à T255	T37 à T63 et T101 à T255
100 ms	T37 à T63 et T101 à T255	T37 à T63 et T101 à T255	T37 à T63 et T101 à T255	T37 à T63 et T101 à T255	T37 à T63 et T101 à T255
Compteurs	C0 à C255	C0 à C255	C0 à C255	C0 à C255	C0 à C255
Compteurs rapides	HC0 à HC5	HC0 à HC5	HC0 à HC5	HC0 à HC5	HC0 à HC5
Relais séquentiels (S)	S0.0 à S31.7	S0.0 à S31.7	S0.0 à S31.7	S0.0 à S31.7	S0.0 à S31.7
Accumulateurs	AC0 à AC3	AC0 à AC3	AC0 à AC3	AC0 à AC3	AC0 à AC3
Sauts/repères	0 à 255	0 à 255	0 à 255	0 à 255	0 à 255
Appels/sous-programmes	0 à 63	0 à 63	0 à 63	0 à 63	0 à 127
Programmes d'interruption	0 à 127	0 à 127	0 à 127	0 à 127	0 à 127
Fronts montants/descendants	256	256	256	256	256
Boucles PID	0 à 7	0 à 7	0 à 7	0 à 7	0 à 7
Interfaces	Interface 0	Interface 0	Interface 0	Interface 0, Interface 1	Interface 0, Interface 1

¹ LB60 à LB63 sont réservés par STEP 7-Micro/WIN, version 3.0 ou ultérieure.

Le tableau ci-dessous donne plages d'opérandes pour les CPU S7-200 :

Accès par		CPU 221	CPU 222	CPU 224	CPU 224 XP	CPU 226
bit (octet bit)	I	0.0 à 15.7	0.0 à 15.7	0.0 à 15.7	0.0 à 15.7	0.0 à 15.7
	Q	0.0 à 15.7	0.0 à 15.7	0.0 à 15.7	0.0 à 15.7	0.0 à 15.7
	V	0.0 à 2047.7	0.0 à 2047.7	0.0 à 8191.7	0.0 à 10239.7	0.0 à 10239.7
	M	0.0 à 31.7	0.0 à 31.7	0.0 à 31.7	0.0 à 31.7	0.0 à 31.7
	SM	0.0 à 165.7	0.0 à 299.7	0.0 à 549.7	0.0 à 549.7	0.0 à 549.7
	S	0.0 à 31.7	0.0 à 31.7	0.0 à 31.7	0.0 à 31.7	0.0 à 31.7
	T	0 à 255	0 à 255	0 à 255	0 à 255	0 à 255
	C	0 à 255	0 à 255	0 à 255	0 à 255	0 à 255
	L	0.0 à 63.7	0.0 à 63.7	0.0 à 63.7	0.0 à 63.7	0.0 à 63.7
octet	IB	0 à 15	0 à 15	0 à 15	0 à 15	0 à 15
	QB	0 à 15	0 à 15	0 à 15	0 à 15	0 à 15
	VB	0 à 2047	0 à 2047	0 à 8191	0 à 10239	0 à 10239
	MB	0 à 31	0 à 31	0 à 31	0 à 31	0 à 31
	SMB	0 à 165	0 à 299	0 à 549	0 à 549	0 à 549
	SB	0 à 31	0 à 31	0 à 31	0 à 31	0 à 31
	LB	0 à 63	0 à 63	0 à 63	0 à 63	0 à 63
	AC	0 à 3	0 à 3	0 à 3	0 à 255	0 à 255
	KB (constante)	KB (constante)	KB (constante)	KB (constante)	KB (constante)	KB (constante)

N.B : La suite du tableau est donnée à la page suivante.

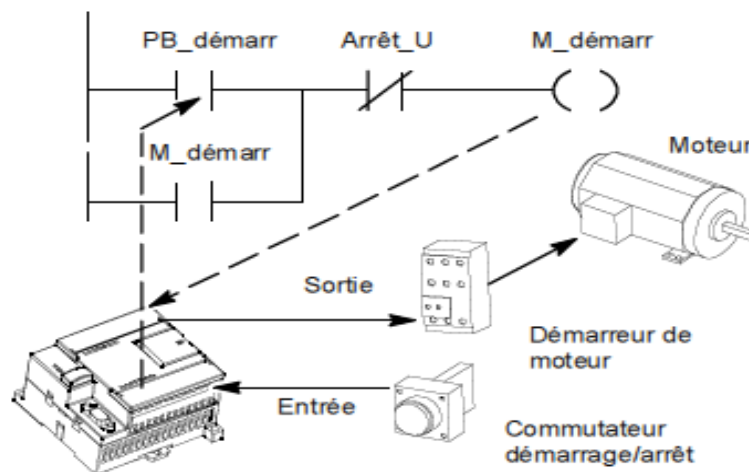
Accès par		CPU 221	CPU 222	CPU 224	CPU 224 XP	CPU 226
mot	IW	0 à 14	0 à 14	0 à 14	0 à 14	0 à 14
	QW	0 à 14	0 à 14	0 à 14	0 à 14	0 à 14
	VW	0 à 2046	0 à 2046	0 à 8190	0 à 10238	0 à 10238
	MW	0 à 30	0 à 30	0 à 30	0 à 30	0 à 30
	SMW	0 à 164	0 à 298	0 à 548	0 à 548	0 à 548
	SW	0 à 30	0 à 30	0 à 30	0 à 30	0 à 30
	T	0 à 255	0 à 255	0 à 255	0 à 255	0 à 255
	C	0 à 255	0 à 255	0 à 255	0 à 255	0 à 255
	LW	0 à 62	0 à 62	0 à 62	0 à 62	0 à 62
	AC	0 à 3	0 à 3	0 à 3	0 à 3	0 à 3
	AIW	0 à 30	0 à 30	0 à 62	0 à 62	0 à 62
	AQW	0 à 30	0 à 30	0 à 62	0 à 62	0 à 62
	KW (constante)	KW (constante)	KW (constante)	KW (constante)	KW (constante)	KW (constante)
double mot	ID	0 à 12	0 à 12	0 à 12	0 à 12	0 à 12
	QD	0 à 12	0 à 12	0 à 12	0 à 12	0 à 12
	VD	0 à 2044	0 à 2044	0 à 8188	0 à 10236	0 à 10236
	MD	0 à 28	0 à 28	0 à 28	0 à 28	0 à 28
	SMD	0 à 162	0 à 296	0 à 546	0 à 546	0 à 546
	SD	0 à 28	0 à 28	0 à 28	0 à 28	0 à 28
	LD	0 à 60	0 à 60	0 à 60	0 à 60	0 à 60
	AC	0 à 3	0 à 3	0 à 3	0 à 3	0 à 3
	HC	0 à 5	0 à 5	0 à 5	0 à 5	0 à 5
	KD (constante)	KD (constante)	KD (constante)	KD (constante)	KD (constante)	KD (constante)

II. 2.2.4) Le cycle d'un API

Le S7-200 exécute en cycles continus la logique de commande de votre programme, en lisant et en écrivant des données. Le S7-200 établit la relation entre votre programme et les entrées et sorties physiques

Le fonctionnement de base du S7-200 est très simple :

- Le S7-200 lit l'état des entrées.
- Le programme contenu dans le S7-200 utilise ces entrées pour évaluer la logique de commande.
- Le S7-200 actualise les données pendant l'exécution du programme.
- Le S7-200 écrit les données dans les sorties.



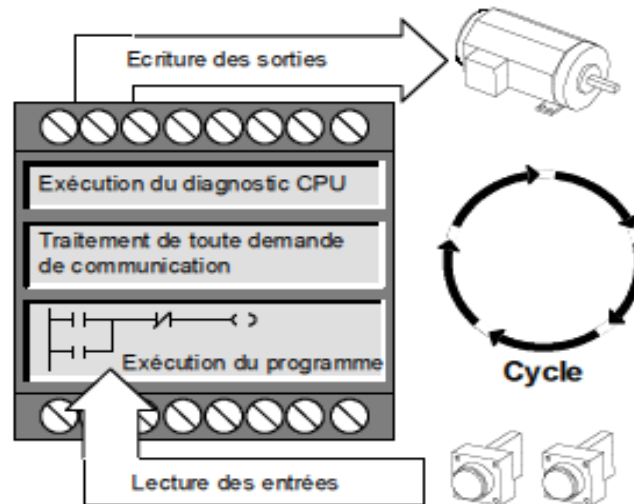
La figure ci-dessus montre un schéma simple illustrant comment un diagramme de relais électrique est en relation avec le S7-200. Dans cet exemple, l'état du commutateur pour le démarrage du moteur est combiné à l'état d'autres entrées. Le calcul de ces états détermine alors l'état de la sortie allant à l'actionneur qui démarre le moteur.

A) Le S7-200 exécute ses tâches au cours d'un cycle :

Le S7-200 exécute une série de tâches de manière répétitive. On appelle "cycle" cette exécution cyclique des différentes tâches. Comme illustré figure 4-2, le S7-200 exécute la plupart ou la totalité des tâches suivantes pendant un cycle :

- Lecture des entrées : Le S7-200 copie l'état des entrées physiques dans la mémoire image des entrées.
- Exécution de la logique de commande dans la programme : Le S7-200 exécute les instructions du programme et sauvegarde les valeurs dans différentes zones de mémoire.
- Traitement de toute demande de communication : Le S7-200 exécute toute tâche nécessaire pour la communication.
- Exécution du test d'auto-diagnostic de la CPU : Le S7-200 s'assure que le microprogramme, la mémoire de programme et les modules d'extension présents fonctionnent correctement.
- Ecriture des sorties : Les valeurs enregistrées dans la mémoire image des sorties sont écrites dans les sorties physiques.

L'exécution du programme utilisateur est fonction de l'état de fonctionnement, Arrêt (STOP) ou Marche (RUN), du S7-200. Votre programme s'exécute à l'état de fonctionnement "Marche" ; il ne s'exécute pas à l'état "Arrêt".



B) Lecture des entrées

Entrées TOR : Chaque cycle lit d'abord la valeur en cours des entrées TOR et l'écrit dans la mémoire image des entrées.

Entrées analogiques : La mise à jour des entrées analogiques de modules d'extension par le S7-200 ne fait pas partie du cycle normal à moins que vous n'ayez activé le filtrage des entrées analogiques. Il existe un filtre analogique permettant d'obtenir un signal plus stable. Vous pouvez activer ce filtre analogique pour chaque entrée analogique.

Lorsque vous activez le filtrage pour une entrée analogique, le S7-200 actualise cette entrée analogique une fois par cycle, exécute la fonction de filtrage et stocke la valeur filtrée en interne. La valeur filtrée est ensuite mise à disposition à chaque fois que votre programme accède à cette entrée.

Lorsque vous n'avez pas activé le filtrage, le S7-200 lit la valeur de l'entrée analogique dans les modules d'extension à chaque fois que votre programme accède à cette entrée.

Les entrées analogiques AIW0 et AIW2 présentes sur la CPU 224XP sont actualisées à chaque cycle avec le résultat le plus récent provenant du convertisseur analogique-numérique. Ce convertisseur est de type sigma-delta (moyennage) et ces valeurs ne nécessiteront normalement pas de filtrage logiciel.

C) Exécution du programme

Pendant la phase d'exécution du cycle, le S7-200 exécute votre programme, de la première instruction à l'instruction de fin. Les opérations d'E/S directes vous permettent d'accéder directement aux entrées et aux sorties, pendant l'exécution du programme ou d'un programme d'interruption.

Si vous utilisez des interruptions dans votre programme, les programmes d'interruption associés aux événements d'interruption sont sauvegardés comme partie du programme. Ils ne sont toutefois pas exécutés au cours du cycle normal, mais uniquement lorsque l'événement d'interruption correspondant se produit (ce qui est possible en tout point du cycle).

- Traitement de toute demande de communication

Pendant la phase de traitement des messages du cycle, le S7-200 traite tous les messages reçus de l'interface de communication ou des modules d'E/S intelligents.

- Exécution du test d'auto-diagnostic de la CPU

Pendant cette phase du cycle, le S7-200 vérifie le bon fonctionnement de la CPU et l'état des modules d'extension.

- Ecriture dans les sorties TOR

A la fin de chaque cycle, le S7-200 écrit dans les sorties TOR les valeurs figurant dans la mémoire image des sorties. Les sorties analogiques sont, elles, mises à jour immédiatement, indépendamment du cycle.

II. 2.2.5) Options de communication (S7 200 – PC/PG)

Siemens propose deux options de programmation pour la connexion de votre ordinateur à votre S7-200 : une connexion directe à l'aide d'un câble PPI multi-maître ou une carte CP (processeur de communication) avec un câble MPI.

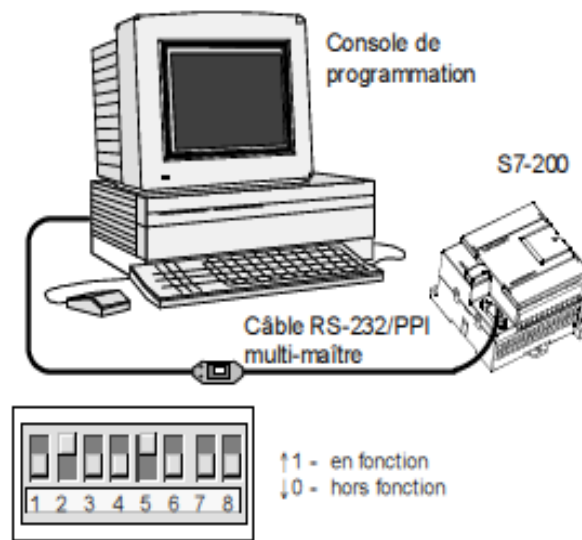
Le câble de programmation PPI multi-maître constitue la solution la plus commune et économique pour connecter votre ordinateur au S7-200. Ce câble relie l'interface de communication du S7-200 au port de communication série de votre ordinateur. Vous pouvez également vous en servir pour raccorder d'autres unités de communication au S7-200.

A) Connexion du câble RS-232/PPI multi-maître

La figure 2-2 montre un câble RS-232/PPI multi-maître reliant le S7-200 à la console de programmation.

Procédez comme suit pour connecter le câble :

1. Raccordez le connecteur RS-232 (identifié par "PC") du câble RS-232/PPI multi-maître à l'interface de communication de la console de programmation. Pour cet exemple, utilisez COM 1
2. Raccordez le connecteur RS-485 (identifié par "PPI") du câble RS-232/PPI multi-maître à l'interface de communication 0 ou 1 du S7-200.
3. Assurez-vous que les commutateurs câble multiples du RS-232/PPI multi-maître sont réglés comme illustré à la figure 2-2.



B) Vérification des paramètres de communication pour STEP 7-Micro/WIN

Le projet-exemple utilise les paramètres par défaut pour STEP 7-Micro/WIN et le câble RS-232/PPI multi-maître. Procédez comme suit pour vérifier ces paramètres :

1. Vérifiez que l'adresse du câble PC/PPI dans la boîte de dialogue "Communication" est définie à 0.
2. Vérifiez que l'interface pour le paramètre de réseau est définie à "PC/PPI câble (COM1)".
3. Vérifiez que la vitesse de transmission est définie à 9,6 kbps.

II. 3) Programmation avec la CPU S7-200

II. 3.1) Progiciel de programmation STEP 7-Micro/WIN

Le progiciel de programmation STEP 7-Micro/WIN fournit un environnement convivial pour concevoir, éditer et surveiller la logique nécessaire à la commande de votre application. STEP 7-Micro/WIN comprend trois éditeurs de programme, ce qui s'avère très pratique et efficace pour la mise au point du programme de commande de votre application. Pour vous aider à trouver les informations dont vous avez besoin, STEP 7-Micro/WIN fournit un système d'aide

en ligne complet et un CD de documentation qui contient une version électronique de ce manuel, des conseils d'application et d'autres informations utiles.

II. 3.2) Environnement matériel requis

STEP 7-Micro/WIN s'exécute sur un ordinateur personnel (PC) ou sur une console de programmation (PG) Siemens, telle que la PG 760. Votre PC ou votre PG doit satisfaire aux exigences minimales suivantes :

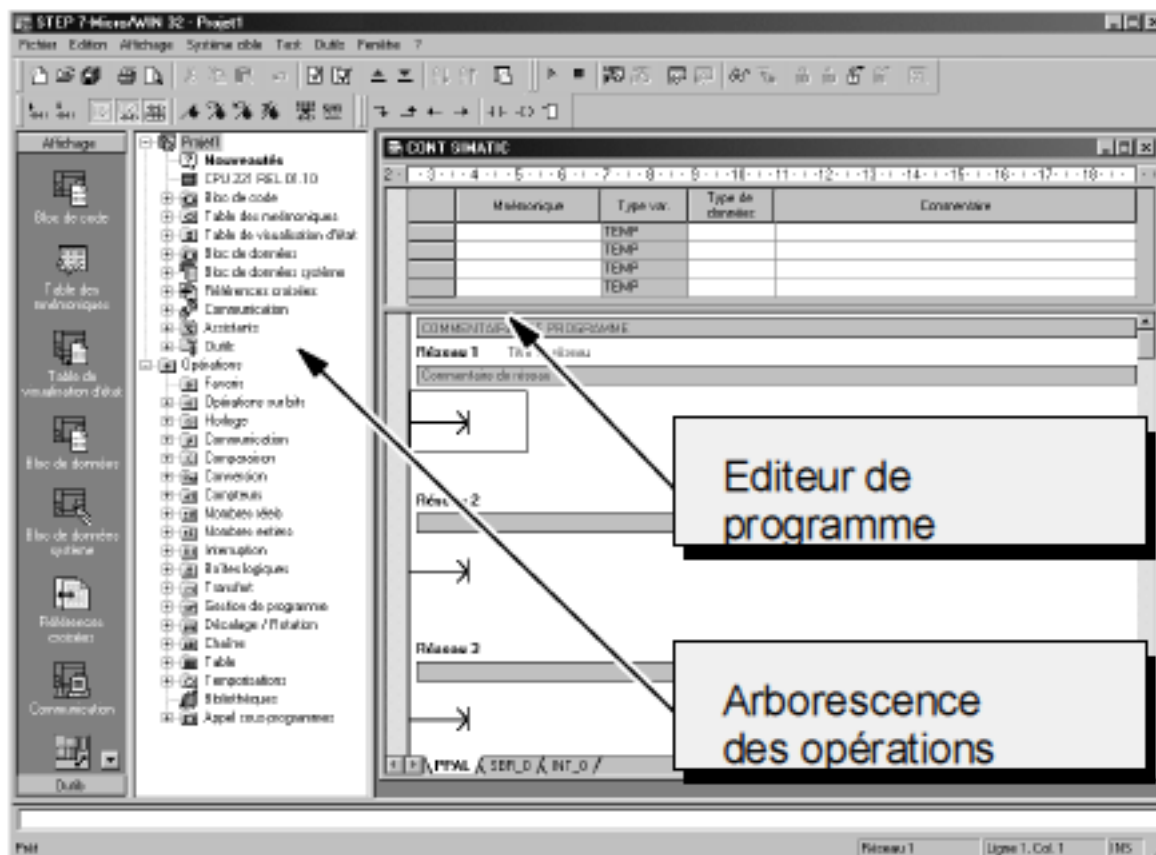
- Système d'exploitation : Windows 2000, Windows XP (Edition professionnelle ou familiale)
- Au moins 100 Mo d'espace libre sur le disque dur
- Souris (recommandée)

II. 3.3) Saisie du programme avec STEP 7-Micro/WIN

Pour ouvrir STEP 7-Micro/WIN, double-cliquez sur l'icône STEP 7-Micro/WIN ou sélectionnez la commande Démarrer > SIMATIC > STEP 7-Micro/WIN 32 V4.0. Comme illustré à la figure ci-dessous, la fenêtre de projet STEP 7-Micro/WIN vous fournit un espace de travail pratique pour créer votre programme de commande.

Examinez l'arborescence des opérations et l'éditeur de programme. L'arborescence des opérations sert à insérer des opérations CONT dans les réseaux de l'éditeur de programme par glisser-déplacer des opérations de l'arborescence aux réseaux.

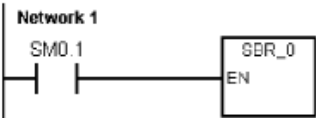
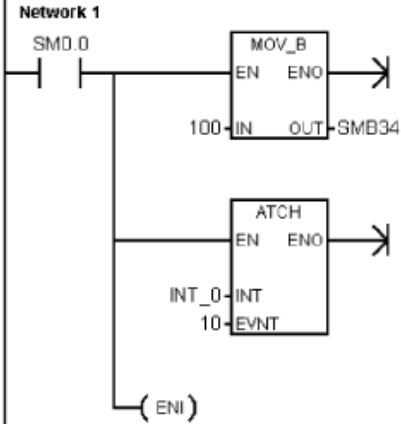

Les icônes de la barre d'outils constituent des raccourcis des commandes de menu. Une fois le programme saisi et enregistré , vous pouvez le charger dans le S7-200.



II. 3.4) Eléments fondamentaux d'un programme

Un bloc de code est composé de code exécutable et de commentaires. Le code exécutable comprend un programme principal et des sous-programmes et programmes d'interruption facultatifs. Le code est compilé et chargé dans le S7-200 ; ce n'est pas le cas des commentaires de programme. Vous pouvez utiliser les éléments d'organisation (programme principal, sous-programmes et programmes d'interruption) pour structurer votre programme de commande.

L'exemple suivant montre un programme comprenant un sous-programme et un programme d'interruption. Cet exemple de programme utilise une interruption cyclique pour lire la valeur d'une entrée analogique toutes les 100 ms.

Exemple : Eléments fondamentaux d'un programme		
P P A L		<p>NETWORK 1 //Appeler sous-programme 0 //au 1er cycle.</p> <pre>LD SM0.1 CALL SBR_0</pre>
		<p>NETWORK 1 //Définir la période de l'interruption //cyclique à 100 ms. //Valider l'interruption 0.</p> <pre>LD SM0.0 MOVB 100, SMB34 ATCH INT_0, 10 ENI</pre>
		<p>NETWORK 1 //Lire l'entrée analogique 4.</p> <pre>LD SM0.0 MOVW AIW4, VW100</pre>

II. 3.4.1) Programme principal

Le programme principal contient les opérations qui commanderont votre application. Le S7-200 exécute ces opérations séquentiellement, une fois par cycle. Le programme principal est également appelé OB1.

II. 3.4.2) Sous-programmes

Ces éléments facultatifs du programme ne sont exécutés que lorsqu'ils sont appelés : par le programme principal, par un programme d'interruption ou par un autre sous-programme. Les sous-programmes sont utiles pour exécuter une fonction de manière répétée. Plutôt que de récrire le code pour chaque endroit du programme principal où la fonction doit être exécutée, vous écrivez le code une fois dans un sous-programme et appelez ce sous-programme autant de fois

que nécessaire dans le programme principal. Les sous-programmes présentent plusieurs avantages :

- Leur utilisation permet de réduire la taille globale de votre programme.
- Leur utilisation diminue le temps de cycle, car vous avez extrait du code du programme principal. Le S7-200 évalue le code dans le programme principal à chaque cycle, que ce code soit exécuté ou non, mais il n'évalue le code dans un sous-programme qu'en cas d'appel de ce dernier; il ne l'évalue donc pas lorsque le sous-programme n'est pas appelé.
- Leur utilisation permet de créer du code qui est portable. Vous pouvez isoler le code pour une fonction dans un sous-programme, puis copier ce sous-programme dans d'autres programmes avec peu ou pas de modifications.

Remarque

L'utilisation d'adresses en mémoire V peut limiter la portabilité de votre sous-programme, car des affectations d'adresses de mémoire V dans un programme peuvent s'avérer incompatibles avec des affectations dans un autre programme. Les sous-programmes utilisant la table des variables locales (mémoire L) pour toutes les affectations d'adresses sont, en revanche, portables dans une large mesure, car il n'est alors pas nécessaire de se préoccuper de conflits d'adresses entre le sous-programme et une autre partie du programme lors de l'utilisation de variables locales.

II. 3.4.2) Programmes d'interruption

Ces éléments facultatifs du programme réagissent à des événements d'interruption spécifiques. Vous concevez un programme d'interruption pour traiter un événement d'interruption prédéfini. Le S7-200 exécute le programme d'interruption lorsque cet événement précis se produit.

Les programmes d'interruption ne sont pas appelés par le programme principal. En effet, vous associez un programme d'interruption à un événement d'interruption et le S7-200 exécute les opérations figurant dans le programme d'interruption uniquement lorsque cet événement se produit.

Remarque

Comme il est impossible de prédire quand le S7-200 va générer une interruption, il vaut mieux limiter le nombre de variables utilisées à la fois par le programme d'interruption et à un autre endroit du programme.

Utilisez la table des variables locales du programme d'interruption pour être sûr que ce dernier ne fait appel qu'à de la mémoire temporaire et n'écrase pas des données provenant d'autres endroits de votre programme.

Il existe un certain nombre de techniques de programmation permettant d'assurer un partage correct des données entre votre programme principal et les programmes d'interruption. Ces techniques sont décrites au chapitre 6 avec les opérations d'interruption.

II. 3.4.3) Autres éléments du programme

D'autres blocs contiennent des informations destinées au S7-200. Vous pouvez choisir de charger ces blocs lorsque vous chargez votre programme dans la CPU.

A) Bloc de données système

Le bloc de données système permet de configurer différentes options matérielles pour le S7-200.

B) Bloc de données

Le bloc de données contient les valeurs de différentes variables (mémoire V) utilisées par votre programme. Vous pouvez vous en servir pour entrer des valeurs initiales pour les données.

II. 3.5) Création de programmes à l'aide de STEP 7-Micro/WIN

L'éditeur de programme contient la logique du programme et une table de variables locales dans laquelle vous affectez des mnémoniques aux variables locales temporaires. Les sous-programmes et les programmes d'interruption apparaissent sous forme d'onglets au bas de la fenêtre de l'éditeur de programme. Cliquez sur ces onglets pour aller et venir entre sous-programmes, programmes d'interruption et programme principal.

STEP 7-Micro/WIN vous propose trois éditeurs pour la création de votre programme : schéma à contacts (CONT-SIMATIC ou LD-CEI), liste d'instructions (LIST) et logigramme (LOG-SIMATIC ou FBD-CEI). Avec quelques restrictions, les programmes écrits dans l'un quelconque de ces éditeurs de programme peuvent être affichés et édités dans les autres éditeurs.

II. 3.5.1) Fonctions de l'éditeur LIST

L'éditeur LIST affiche le programme sous forme textuelle. Avec l'éditeur LIST, vous entrez des abréviations d'opérations quand vous créez des programmes de commande. L'éditeur LIST permet également de créer des programmes qu'il n'est pas possible de représenter avec les éditeurs CONT/LD ou LOG/FBD. En effet, avec cet éditeur, vous programmez dans un langage natif du S7-200 et non dans un éditeur graphique où des restrictions sont nécessaires pour que les schémas soient tracés correctement. Comme vous le constatez dans l'exemple ci-dessous, la programmation avec ce langage littéral est très similaire à la programmation en langage assembleur.

Exemple de programme LIST:

```
LD I0.0    //Lire une entrée
A I0.1     //La combiner selon ET avec autre entrée
= Q1.0     //Ecrire valeur dans sortie 1
```

Le S7-200 exécute chaque opération dans l'ordre spécifié par le programme, de haut en bas, puis recommence en haut. LIST utilise une pile logique pour résoudre la logique de commande. Vous devez insérer les opérations pour LIST gérer le fonctionnement de la pile.

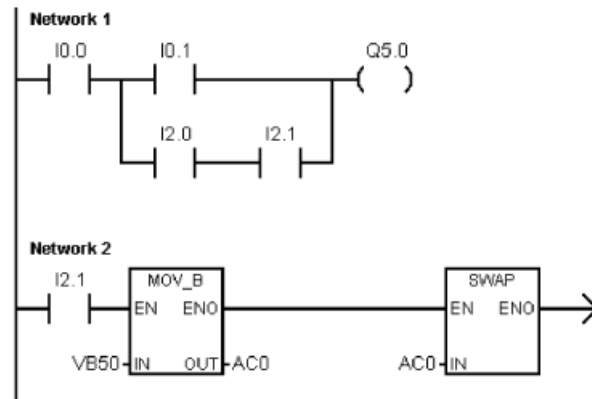
Considérez les aspects principaux suivants lorsque vous sélectionnez l'éditeur LIST :

- LIST convient le mieux aux programmeurs expérimentés.
- LIST vous permet parfois de résoudre des problèmes que vous ne pouvez pas résoudre facilement avec les éditeurs CONT/LD ou LOG/FBD.
- Seul le jeu d'opérations SIMATIC est disponible avec l'éditeur LIST.
- Il est toujours possible, avec l'éditeur LIST, de visualiser ou d'éditer des programmes créés avec les éditeurs CONT ou LOG. L'inverse n'est pas toujours vrai : vous ne pouvez pas toujours afficher un programme écrit sous l'éditeur LIST avec les éditeurs CONT ou LOG.

II. 3.5.2) Fonctions de l'éditeur CONT

L'éditeur CONT (ou LD) affiche le programme sous forme de représentation graphique similaire aux schémas de câblage électriques. Les programmes CONT permettent au programme d'émuler le flux de courant électrique partant d'une source de tension, à travers une série de conditions d'entrée logiques validant, à leur tour, des conditions de sortie logiques. Un programme CONT comprend une barre d'alimentation à gauche qui est alimentée en courant. Les contacts fermés permettent au courant de circuler à travers eux vers l'élément suivant alors que les contacts ouverts bloquent le trajet du courant. La logique est subdivisée en réseaux. Le programme est exécuté réseau par réseau, de la gauche vers la droite et de haut en bas comme indiqué par le programme. L'exemple ci-dessous montre un exemple de programme CONT. Les différentes opérations sont représentées par des symboles graphiques de trois types fondamentaux.

Exemple de programme en CONT/LD :



Les contacts représentent des conditions d'entrée logiques, telles que commutateurs, boutons-poussoirs ou conditions internes. Les bobines représentent généralement des résultats de sortie logiques, tels que lampes, de moteur relais démarreurs moteur, intermédiaires ou conditions de sortie internes.

Les boîtes représentent des opérations supplémentaires, telles que temporisations, compteurs ou opérations arithmétiques.

Considérez les aspects principaux suivants lorsque vous sélectionnez l'éditeur CONT :

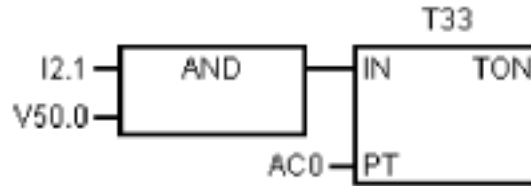
- Les programmeurs débutants apprennent facilement le schéma à contacts.
- La représentation graphique facilite la compréhension et est utilisée partout dans le monde.
- Vous pouvez utiliser le schéma à contacts avec les jeux d'opérations SIMATIC (CONT) et CEI 1131-3 (LD).
- Vous pouvez toujours afficher avec l'éditeur LIST un programme que vous avez créé avec l'éditeur CONT SIMATIC.

II. 3.5.2) Fonctions de l'éditeur LOG

L'éditeur LOG (ou FBD) affiche le programme sous forme de représentation graphique similaire aux schémas de portes logiques courants. Il n'y a pas de contacts ni de bobines comme dans l'éditeur CONT/LD, mais des opérations équivalentes existent sous forme de boîtes d'opérations.

L'exemple ci-dessous montre un programme LOG. LOG ne fait pas appel au concept de barres d'alimentation gauche et droite, ainsi le terme "flux de signal" sert à exprimer le concept de commande travers analogue flux à les blocs logiques LOG.

Exemple de programme en LOG/FBD :



On appelle le chemin 1 logique à travers des éléments LOG "flux de signal". On peut affecter directement à un opérande l'origine d'une entrée de flux de signal et la destination d'une sortie de flux de signal.

La logique du programme est dérivée des liaisons entre ces boîtes d'opérations : on peut utiliser la sortie d'une opération (d'une boîte ET, par exemple) pour valider une autre opération (une temporisation, par exemple) afin de créer la logique de commande nécessaire. Ce concept de liaison vous permet de résoudre un large éventail de problèmes de logique.

Considérez les aspects principaux suivants lorsque vous sélectionnez l'éditeur LOG :

- La représentation graphique de type porte logique permet de suivre aisément le déroulement du programme.
- Vous pouvez utiliser cet éditeur avec les jeux d'opérations SIMATIC - vous le connaissez alors sous le nom d'éditeur LOG - et CEI 1131-3 - il porte alors le nom d'éditeur Langage FBD.
- Vous pouvez toujours afficher avec l'éditeur LIST un programme que vous avez créé avec l'éditeur LOG SIMATIC.

II. 3.5.4) Choix entre jeux d'opérations SIMATIC et CEI 1131-3

La plupart des automates programmables proposent des types d'opérations fondamentales similaires, mais il y a généralement de petites différences dans leur représentation, fonctionnement, etc., d'un fabricant d'automates à l'autre. Ainsi, pendant les dernières années, la Commission Electrotechnique Internationale (CEI) a mis au point une norme globale naissante portant spécifiquement sur de nombreux aspects de la programmation d'automates. Cette norme encourage les fabricants d'automates programmables à proposer des opérations similaires à la fois dans leur représentation et leur fonctionnement.

Votre S7-200 propose deux jeux d'opérations vous permettant de résoudre une large gamme de tâches d'automatisation. Le jeu d'opérations CEI est conforme à la norme CEI 1131-3 pour la programmation d'automates alors que le jeu d'opérations SIMATIC a été spécialement conçu pour le S7-200.

Remarque

Lorsque le mode CEI est activé dans STEP 7-Micro/WIN, un losange rouge (♦) est affiché dans l'arborescence des opérations à côté des opérations qui ne sont pas définies par la norme CEI 1131-3.

Il existe quelques différences fondamentales entre le jeu d'opérations SIMATIC et le jeu d'opérations CEI :

- Le jeu d'opérations CEI est limité aux opérations standard parmi les fournisseurs d'automates programmables. Certaines opérations faisant partie du jeu SIMATIC standard ne sont pas des opérations standard dans la norme CEI 1131-3, mais restent disponibles comme opérations non standard (si vous les utilisez, votre programme n'est alors plus strictement compatible avec la norme CEI 1131-3).
- Certaines boîtes d'opérations CEI acceptent plusieurs formats de données. On parle alors d'opération surchargée. Par exemple, plutôt que d'avoir les boîtes d'opérations arithmétiques ADD_I (Additionner entiers) et ADD_R (Additionner réels), l'opération ADD de CEI examine le format des données ajoutées et choisit automatiquement l'opération correcte dans le S7-200. Cela permet de gagner un temps précieux lors de la conception du programme.
- Si vous utilisez les opérations CEI, les paramètres des opérations sont automatiquement vérifiés pour voir s'ils ont le format de données correct, par exemple entier signé ou entier non signé. Par exemple, une erreur se produit si vous essayez d'entrer une valeur entière pour une opération attendant une valeur binaire (0 ou 1). Cette fonction réduit les erreurs dans la syntaxe de programmation.

Vous devez considérer les aspects suivants lorsque vous sélectionnez le jeu d'opérations SIMATIC ou le jeu d'opérations CEI :

- Les opérations SIMATIC ont généralement un temps d'exécution plus court. Le temps d'exécution des opérations CEI peut être plus long.
- Certaines opérations CEI, telles que temporisations, compteurs, multiplication et division, fonctionnent différemment de leur équivalent SIMATIC.
- Vous pouvez utiliser les trois éditeurs de programme (CONT, LIST et LOG) avec le jeu d'opérations SIMATIC. Seuls les éditeurs de programme LD (équivalent de CONT) et FBD (équivalent de LOG) sont disponibles pour les opérations CEI.
- Le fonctionnement des opérations CEI est standard pour différentes marques d'AP et les connaissances dans la création de programmes conformes à la norme CEI peuvent être étendues à toutes les plateformes d'AP.
- La norme CEI définit moins d'opérations qu'il n'y en a dans le jeu d'opérations SIMATIC, mais vous pouvez toujours inclure des opérations SIMATIC dans votre programme CEI.
- La norme CEI 1131-3 indique qu'il faut déclarer un type pour les variables et prévoit le contrôle système du type de données.

II. 3.6) Opérations combinatoires sur bits

II. 3.6.1) Contacts

A) Contacts standard

Les opérations Contact à fermeture (LD, A, O) et Contact à ouverture (LDN, AN, ON) obtiennent la valeur référencée dans la mémoire ou dans la mémoire image du processus. Les opérations Contacts standard obtiennent la valeur référencée dans la mémoire ou dans la mémoire image si le type de données est I ou Q.

Le contact à fermeture est fermé (activé) lorsque le bit est égal à 1 et le contact à ouverture est fermé (activé) lorsque le bit est égal à 0. En LOG, vous pouvez rajouter des entrées aux boîtes AND et OR à condition de ne pas dépasser 32 entrées au maximum. En LIST, l'opération LD charge la valeur du bit en haut de la pile et les opérations A et O combinent la valeur du bit à la valeur supérieure de la pile selon ET et OU respectivement. L'opération LDN charge la négation de la valeur du bit en haut de la pile et les opérations AN et ON combinent cette valeur inversée à

la valeur supérieure de la pile selon ET et OU respectivement.

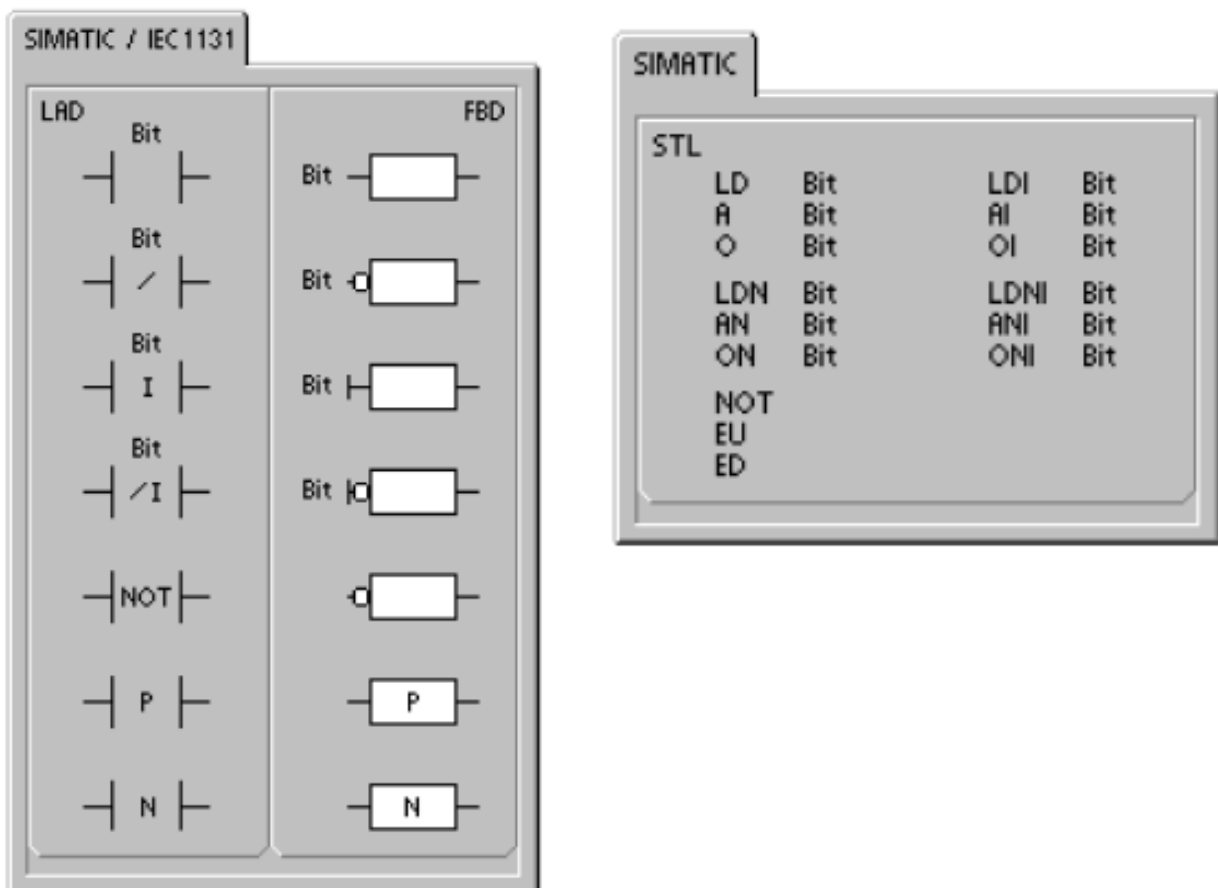
B) Contacts directs

Un contact direct n'est pas actualisé en fonction du cycle du S7-200, mais immédiatement. Les opérations Contact direct à fermeture (LDI, AI, OI) et Contact direct à ouverture (LDNI, ANI, ONI) lisent la valeur de l'entrée physique lors de l'exécution de l'opération, mais la mémoire image n'est pas mise à jour.

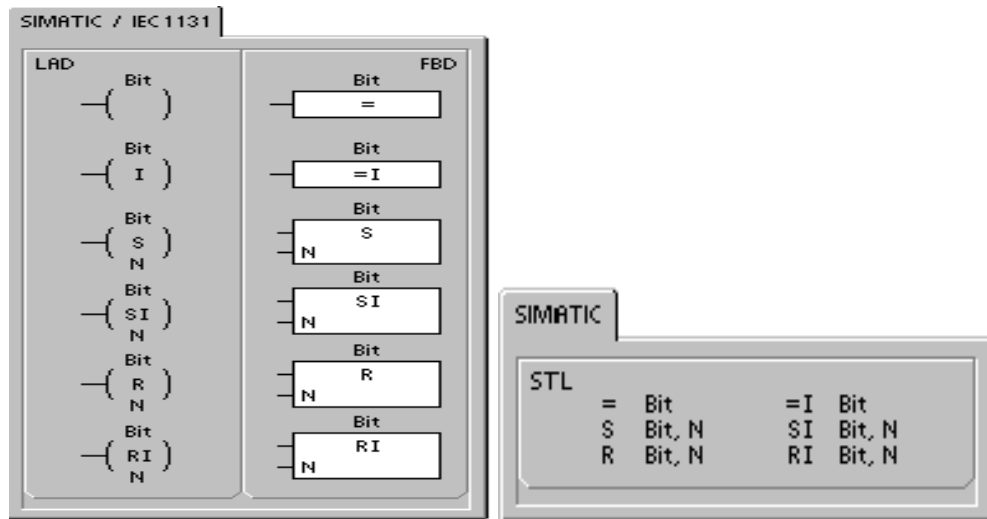
Le contact direct à fermeture est fermé (activé) lorsque le bit de l'entrée physique est égal à 1 et le contact direct à ouverture est fermé (activé) lorsque le bit de l'entrée physique est égal à 0. En LIST, l'opération LDI charge directement la valeur de l'entrée physique en haut de la pile et les opérations AI et OI combinent, et ce directement, la valeur de l'entrée physique à la valeur supérieure de la pile selon ET et OU respectivement. L'opération LDNI charge directement la négation de la valeur de l'entrée physique en haut de la pile et les opérations ANI et ONI combinent, et ce directement, cette valeur inversée à la valeur supérieure de la pile selon ET et OU respectivement.

C) Opération NOT

L'opération NOT change l'état de l'entrée de flux de signal (elle inverse la valeur supérieure de la pile, de 0 à 1 ou de 1 à 0).



IV. 3.6.1) Bobines



A) Sortie

L'opération Sortie (=) écrit la nouvelle valeur du bit de sortie dans la mémoire image. Lors de l'exécution de l'opération Sortie, le S7-200 met le bit de sortie dans la mémoire image à 1 ou à 0. En CONT et en LOG, le bit indiqué est posé égal au flux de signal. En LIST, la valeur supérieure de la pile est copiée dans le bit indiqué.

B) Sortie directe

L'opération Sortie directe (=I) écrit la nouvelle valeur à la fois dans la sortie physique et dans l'adresse correspondante de la mémoire image lors de l'exécution de l'opération. Lorsque l'opération Sortie directe est exécutée, la sortie physique (bit) est directement posée égale au flux de signal. En LIST, l'opération de sortie directe copie directement la valeur supérieure de la pile dans le bit de sortie physique indiqué. Le "I" - pour immédiat - signifie que la nouvelle valeur est écrite à la fois dans la sortie physique et dans l'adresse correspondante de la mémoire image lors de l'exécution de l'opération. Pour les opérations indirectes en revanche, la nouvelle valeur est écrite dans la mémoire image uniquement.

C) Mettre à 1, Mettre à 0

Les opérations Mettre à 1 (S) et Mettre à 0 (R) mettent à 1 (activent) ou mettent à 0 (désactivent) le nombre N de sorties indiquées, en commençant à l'adresse Bit indiquée. Vous pouvez mettre 1 à 255 sorties à 1 ou à 0.

Si, pour l'opération Mettre à 0, le bit correspond à un bit de temporisation (T) ou de compteur (C), l'opération met le bit de temporisation ou de compteur à 0 et efface la valeur de comptage ou de temporisation en cours.

D) Mettre à 1 directement, Mettre à 0 directement

Les opérations Mettre à 1 directement et Mettre à 0 directement mettent à 1 (activent) ou mettent à 0 (désactivent), et ce directement, le nombre N de sorties indiquées, en commençant à l'adresse Bit indiquée. Vous pouvez mettre directement à 1 ou à 0 de 1 à 128 sorties.

Le "I" - pour immédiat - signifie que la nouvelle valeur est écrite à la fois dans la sortie physique

et dans l'adresse correspondante de la mémoire image lors de l'exécution de l'opération. Pour les opérations indirectes en revanche, la nouvelle valeur est écrite dans la mémoire image uniquement.

II. 3.7) Opérations d'horloge

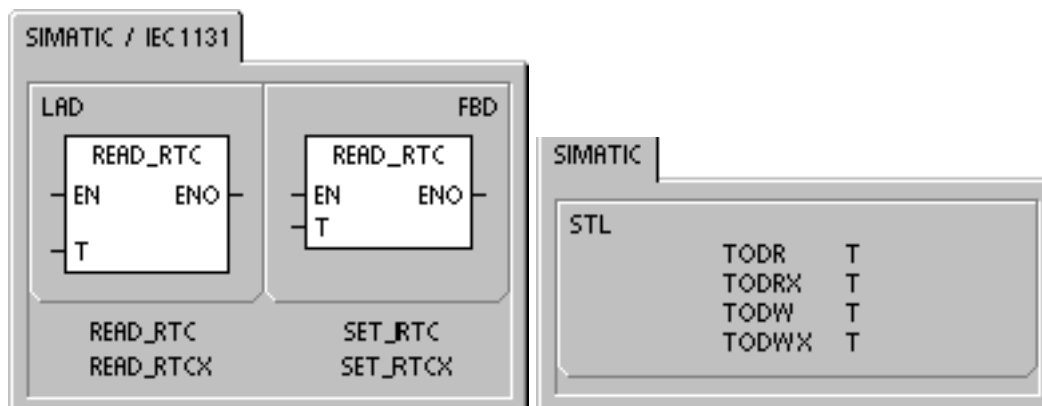
II. 3.7.1) Lire horloge temps réel, Générer horloge temps réel

L'opération Lire horloge temps réel (TODR) lit la date et l'heure en cours de l'horloge temps réel et les charge dans une mémoire tampon de datation de huit octets commençant à l'adresse T. L'opération Générer horloge temps réel (TODW) écrit dans l'horloge temps réel la date et l'heure en cours, en commençant à l'adresse de la mémoire tampon de datation de huit octets indiquée par T.

Vous devez coder toutes les valeurs de date et d'heure en format décimal codé binaire (par exemple, 16#97 pour l'année 1997). La figure 6-4 montre le format de la mémoire tampon de datation (T) de 8 octets.

L'horloge temps réel est initialisée comme suit en cas de coupure de courant prolongée ou de perte de mémoire :

Date : 01-Jan-90
Heure : 00:00:00
Jour : dimanche



Remarque

La CPU S7-200 ne vérifie pas si le jour de la semaine correspond bien à la date. Elle peut accepter des dates incorrectes comme le 30 février. C'est à vous de vous assurer que la date est correcte.

N'utilisez pas l'opération TODR/TODW à la fois dans le programme principal et dans un programme d'interruption. En effet, si vous le faites et que l'opération d'horloge temps réel soit en cours d'exécution quand se produit l'interruption qui exécute également l'opération d'horloge temps réel, cette opération ne peut pas être exécutée dans le programme d'interruption. Le bit SM4.3 sera mis à 1 pour signaler qu'il y a eu deux tentatives simultanées d'accès à l'horloge (erreur bénigne 0007).

L'horloge temps réel dans le S7-200 utilise seulement les deux chiffres les moins significatifs pour l'année. Ainsi, l'an 2000 est représenté par "00". L'automate S7-200 n'utilise

en aucune façon les données concernant l'année. Toutefois, les programmes utilisateur exécutant des opérations arithmétiques ou de comparaison avec la valeur de l'année doivent prendre en compte cette représentation à deux chiffres et le changement de millénaire.

Les années bissextiles sont prises en compte jusqu'en 2096.

II. 3.7.2) Lire horloge temps réel (étendu)

L'opération TODRX (Lire horloge temps réel, étendu) lit la configuration en cours pour la date, l'heure et l'heure d'été dans l'AP et charge cette configuration dans une mémoire tampon de 19 octets en commençant à l'adresse indiquée par T.

II. 3.7.3) Régler horloge temps réel (étendu)

L'opération TODWX (Régler horloge temps réel, étendu) écrit la configuration en cours pour la date, l'heure et l'heure d'été dans l'AP en commençant à l'adresse, indiquée par T, de la mémoire tampon de 19 octets.

Vous devez coder toutes les valeurs de date et d'heure en format décimal codé binaire (par exemple, 16#02 pour l'année 2002). La figure 6-9 montre le format de la mémoire tampon de datation (T) de 19 octets.

L'horloge temps réel est initialisée comme suit en cas de coupure de courant prolongée ou de perte de mémoire :

Date : 01-Jan-90
Heure : 00:00:00
Jour : dimanche

II. 3.8) Opération Calcul PID

L'opération Calcul PID (PID) exécute un calcul de boucle PID sur la boucle LOOP référencée en se basant sur les entrées et les informations de configuration figurant dans TABLE (TBL).

L'opération de boucle PID (régulateur à actions proportionnelle, intégrale et dérivée) permet le calcul du régulateur PID. La valeur supérieure de la pile doit être à 1 (le courant circule) pour valider le calcul PID. Cette opération comporte deux opérandes : TBL contient l'adresse de début de la table pour la boucle de régulation et LOOP indique le numéro (compris entre 0 et 7) de la boucle de régulation.

Vous pouvez utiliser huit opérations PID dans un programme. Si deux opérations PID ou plus indiquent le même numéro de boucle, les calculs PID interféreront les uns avec les autres et le résultat sera imprévisible, et ce même si les adresses des tables sont différentes.

La table pour la boucle contient neuf paramètres servant à la commande et à la surveillance de l'opération de boucle. Il s'agit de la valeur en cours et de la valeur précédente de la mesure (variable du processus), de la consigne, de la grandeur réglante, du gain (coefficient d'action proportionnelle), de la période d'échantillonnage, du temps d'intégration, du temps de dérivation et de la somme intégrale (bias).

Pour pouvoir effectuer le calcul PID avec la période d'échantillonnage désirée, il faut exécuter l'opération PID soit à l'intérieur d'un programme d'interruption cyclique, soit à l'intérieur du programme principal à une fréquence gérée par une temporisation. La période d'échantillonnage est un paramètre d'entrée de l'opération PID que vous indiquez dans la table pour la boucle de régulation.

Une fonction de mise au point automatique a été intégrée à l'opération PID. Vous en trouverez une description détaillée au chapitre 15. Le panneau de commande de mise au point PID fonctionne uniquement avec des boucles PID créées par l'assistant PID.

II. 4) Calcul d'un bilan de consommation

La CPU S7-200 possède une alimentation interne fournissant du courant à la CPU elle-même, aux modules d'extension, ainsi qu'à d'autres équipements consommant du courant 24 V. Les informations ci-après doivent vous aider à déterminer combien d'énergie ou de courant la CPU S7-200 peut mettre à la disposition de votre configuration.

II. 4.1) Besoins en courant

Chaque CPU S7-200 fournit du courant 5 V et 24 V.

- Chaque CPU S7-200 a une alimentation de capteur 24 V pouvant fournir du courant continu en 24 V aux entrées locales ou aux bobines de relais sur les modules d'extension.

Si les besoins en courant 24 V dépassent le courant fourni par la CPU, vous pouvez ajouter une alimentation 24 V externe afin de fournir ce courant aux modules d'extension.

Vous devez connecter à la main l'alimentation 24 V aux entrées ou aux bobines de relais.

- La CPU fournit également du courant 5 V pour les modules d'extension lorsqu'un tel module est connecté. Si les besoins en courant 5 V des modules d'extension dépassent le courant fourni par la CPU, vous devez supprimer des modules d'extension jusqu'à ce que leurs besoins soient couverts.

Les caractéristiques techniques à l'annexe A donnent des informations sur le courant fourni par les CPU et sur les besoins en courant des modules d'extension.

Remarque

- Si le bilan de consommation CPU est déficitaire, vous ne pourrez peut-être pas connecter le nombre maximal de modules autorisés pour votre CPU.

- Connecter une alimentation 24 V externe en parallèle avec l'alimentation de capteur en courant continu S7-200 peut entraîner un conflit entre les deux alimentations, chacune cherchant à établir son propre niveau de tension de sortie préféré.

- Ce conflit peut réduire la durée de vie ou provoquer une défaillance immédiate de l'une ou des deux alimentations, ayant pour effet un fonctionnement imprévisible du système d'automatisation pouvant entraîner la mort, des blessures graves et des dommages matériels importants.

- L'alimentation de capteur CC S7-200 et toute alimentation externe doivent fournir du courant à des points différents, une seule connexion des conducteurs neutres étant autorisée.

II. 4.2) Calcul d'un exemple de bilan de consommation

L'exemple présenté au tableau B-1 montre comment calculer le bilan de consommation pour un automate S7-200 comprenant :

- une CPU 224 S7-200 CA/CC/Relais

- trois EM 223, 8 entrées CC/8 sorties relais

- un EM 221, 8 entrées CC

Cette installation comporte 46 entrées et 34 sorties au total.

Conseil

Remarque

La CPU a déjà alloué le courant nécessaire pour piloter les bobines de relais internes. Vous n'avez donc pas besoin d'inclure les besoins en courant des bobines de relais internes dans votre bilan de consommation.

Dans cet exemple, la CPU S7-200 fournit suffisamment de courant continu 5 V pour les modules d'extension, mais pas suffisamment de courant continu 24 V à partir de l'alimentation de capteur pour toutes les entrées et les bobines de relais d'extension. Les E/S requièrent 400 mA alors que la CPU S7-200 ne fournit que 280 mA. Il faut donc une source supplémentaire d'au moins 120 mA à 24 V pour opérer toutes les entrées et sorties 24 V présentes.

Les tables ci-dessous présentent le calcul du bilan de consommation pour un exemple de configuration :

Courant fourni par la CPU	5 V-	24 V-
CPU 224 CA/CC/relais	660 mA	280 mA

Moins

Besoins du système	5 V-	24 V-
CPU 224, 14 entrées		$14 \times 4 \text{ mA} = 56 \text{ mA}$
3 EM 223, puissance 5 V nécessaire	$3 \times 80 \text{ mA} = 240 \text{ mA}$	
1 EM 221, puissance 5 V nécessaire	$1 \times 30 \text{ mA} = 30 \text{ mA}$	
3 EM 223, 8 entrées chacun		$3 \times 8 \times 4 \text{ mA} = 96 \text{ mA}$
3 EM 223, 8 bobines de relais chacun		$3 \times 8 \times 9 \text{ mA} = 216 \text{ mA}$
1 EM 221, 8 entrées		$8 \times 4 \text{ mA} = 32 \text{ mA}$
Total des besoins	270 mA	400 mA

égale

Bilan de consommation	5 V-	24 V-
Excédent/déficit de courant	390 mA	[120 mA]

II. 4.2) Calcul de votre bilan de consommation

Servez-vous du tableau ci-dessous pour déterminer combien de courant la CPU S7-200 peut mettre à la disposition de votre configuration. Vous trouverez à l'annexe A des informations sur le courant fourni par votre modèle de CPU et sur les besoins en courant de vos modules d'extension.

Courant fourni par la CPU	5 V-	24 V-

Moins

Besoins du système	5 V-	24 V-
Total des besoins		

égale

Bilan de consommation	5 V-	24 V-
Excédent/déficit de courant		

Bibliographie :

Famille de produits	Documentation
S7-200	Guide de l'utilisateur du panneau tactile TP070 (anglais)
	TP170 micro, Manuel de mise en oeuvre (anglais)
	Communication interface point à point S7-200, Manuel (anglais/allemand)
	Manuel maître interface AC CP 243-2 SIMATIC NET (anglais)
	Module Internet CP 243-1 IT (avec documentation électronique sur CD)
	Module Ethernet CP 243-1 (avec documentation électronique sur CD)
	Automate programmable S7-200, Manuel système