

SAIA® PLC

Automates programmables

Manuel du logiciel niveau 2+3



SOFTWARE NIVEAU 2 + 3

- PARTIE G INTRODUCTION**
- PARTIE H JEU DES INSTRUCTIONS ADDITIONNELLES
DU PROCESSEUR DE BIT NIVEAU 2**
- PARTIE I JEU DES INSTRUCTIONS DU
PROCESSEUR DE MOTS NIVEAU 3**
- PARTIE K SORTIE DE TEXTE ET COMMUNICATION
(NIVEAUX 2 + 3)**
- PARTIE L EXEMPLES DE PROGRAMMATION**

TABLE DES MATIERES

		Page
PARTIE G	INTRODUCTION	
G 1	Compatibilité au sein de la famille PCA	1G
G 1.1	Vue d'ensemble de la famille d'automates SAIA°PLC	2G
G 2	Vue d'ensemble des manuels de la famille PCA	3G
G 3	Structure des registres du processeur avec les niveaux ② et ③	4G
G 4	Jeu d'instructions du SAIA°PLC niveau de logiciel ①	5G
G 5	Instructions complémentaires du niveau de logiciel ①H	6G
G 6	Vue d'ensemble du jeu d'instruction élargi du processeur de bits niveau ②	7G
G 7	Vue d'ensemble du jeu d'instruction du processeur de mots niveau ③ (seulement PCA23)	8G
PARTIE H	JEU DES INSTRUCTIONS ADDITIONNELLES NIVEAU ②	
H 1	Instructions à une ligne	1H
H 2	Instructions PAS à deux lignes	2H
	PAS 16/17	2H
	PAS 19	5H
	PAS 23	5H
	PAS 24	5H
	PAS 50	6H
	PAS 54/56	7H
	PAS 55/57	12H
	PAS 58	16H
H 3	Instructions PAS à dix lignes	20H
	PAS 190 et PAS 19	20H
	PAS 200 à 212	28H
	PAS 250/251	41H
PARTIE I	INSTRUCTIONS DU PROCESSEUR DE MOTS NIVEAU ③ (seulement PCA232 ou PCA231)	
I 1	Changement du processeur de bits au processeur de mots, jeu d'instruction du processeur de mots	1I
I 2	Instructions arithmétiques	4I
I 3	Instructions pour le traitement des registres A	7I
I 4	Vue d'ensemble des instructions de transfert	9I
I 5	Instructions de comptage et de saut	12I
I 6	Diverses instructions	14I

PARTIE K SORTIE DE TEXTE ET COMMUNICATION (Niveaux ② et ③)

K 1	L'interface série	1K
K 1.1	Qu'est-ce qu'une interface série?	1K
K 1.1.1	Qu'est-ce que le code ASCII?	2K
K 1.1.2	Transmission série de données, unité Baud, parité	4K
K 2	La boucle de courant 20mA (également nommé interface TTY, interface à courant de ligne ou boucle de courant)	5K
K 2.1	PCA2.M22 et M32	5K
K 2.2	PCA14 et PCA02	6K
K 3	Fonction de la lampe ERROR placée sur la face frontale	8K
K 4	Assigination de l'interface sur le CPU du PCA	9K
K 5	Entrée/sortie de texte	11K
K 5.1	Organisation de la mémoire de textes	11K
K 5.2	Sortie d'un texte	11K
K 5.3	Entrée de textes	13K
K 5.4	Entrée/sortie de texte à 8 bits	22K
K 6	Transmission de données par l'interface série	23K
K 6.1	Introduction, mode de communication	23K
K 6.2	Définition du mode C	27K
K 6.2.1	Assigination pour le mode C	30K
K 6.3	Définition du mode N	31K
K 6.3.1	Assigination pour le mode N	34K
K 6.3.2	Les télégrammes du mode N	36K
K 6.4	Définition du mode P	41K
K 6.4.1	Assigination pour le mode P	42K
K 6.4.2	Les télégrammes du mode P	42K
K 6.4.3	Ecriture de données dans le PCA	46K
K 6.4.4	Lecture de données du PCA	52K
K 6.5	Assigination pour les modes combinés	57K
K 6.6	Vue générale du PAS 100 et des variantes de fonctionnement	58K

PARTIE L EXEMPLES DE PROGRAMMATION

Exemple 1	Fonction d'enclenchement et sortie de texte à une heure prédéterminée (Utilisation de l'instruction PAS 50)	2L
Exemple 2	"Check-Sum" du programme-système et programme utilisateur avec activation du "Watchdog" (Utilisation du PAS 30...38)	6L
Exemple 3	Signes spéciaux pour textes	8L
Exemple 4	Jeu de balle	10L
Exemple 5	Echange de données entre deux PCA14 dans le mode N par l'interface série	12L

Exemple 6	Boucle de réglage PID rapide avec l'instruction PAS 202 et le PCA14 (Moteur-tachymètre)	15L
Exemple 7	Réglage manuel/PID avec PAS 211 et PCA14 (Module analogique 12 bits PCA1.W32)	17L
Exemple 8	Registre à décalage rotatif avant/arrière avec remise à zéro (Utilisation de l'instruction PAS 250)	19L
Exemple 9	Registre à décalage BCD avant/arrière (Utilisation du PAS 250)	21L
Exemple 10	Registre FIFO (Utilisation de l'instruction PAS 251)	23L
Exemple 11	Signalisations de dérangements (Mémorisation dans le registre FIFO par PAS 251)	26L
Vue d'ensemble alphabétique des instructions 2 + 3		29L

PARTIE G INTRODUCTION

- G 1** **Compatibilité au sein de la famille PCA**
- G 1.1** **Vue d'ensemble de la famille d'automates SAIA° PCA**
- G 2** **Vue d'ensemble des manuels de la famille SAIA PCA**
- G 3** **Structure des registres du processeur avec les
niveaux 2 et 3**
- G 4** **Jeu d'instructions du SAIA°PLC niveau de logiciel 1**
- G 5** **Instructions complémentaires du niveau de logiciel 1H**
- G 6** **Vue d'ensemble du jeu d'instruction élargi du
processeur de bits niveau 2**
- G 7** **Vue d'ensemble du jeu d'instruction du processeur
de mots niveau 3**

PARTIE G INTRODUCTION

G 1 Compatibilité dans la famille PCA

Le logiciel de la famille PCA est complètement compatible vers le haut. Chaque nouvelle série de PLC contient toutes les instructions des modèles antérieurs. Ceci permet à chaque client possédant déjà un "vieux programme d'une ancienne version de CPU, de pouvoir aussi le faire "tourner" sur un automate plus performant.


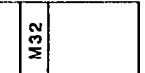




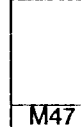
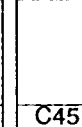
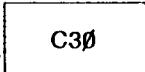
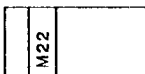

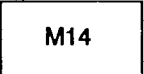
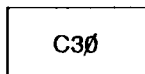
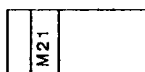

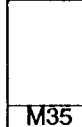
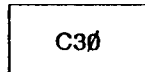

Cette compatibilité reste valable à l'intérieur d'une même série (par exemple PCA1) entre les différentes version d'un processeur.

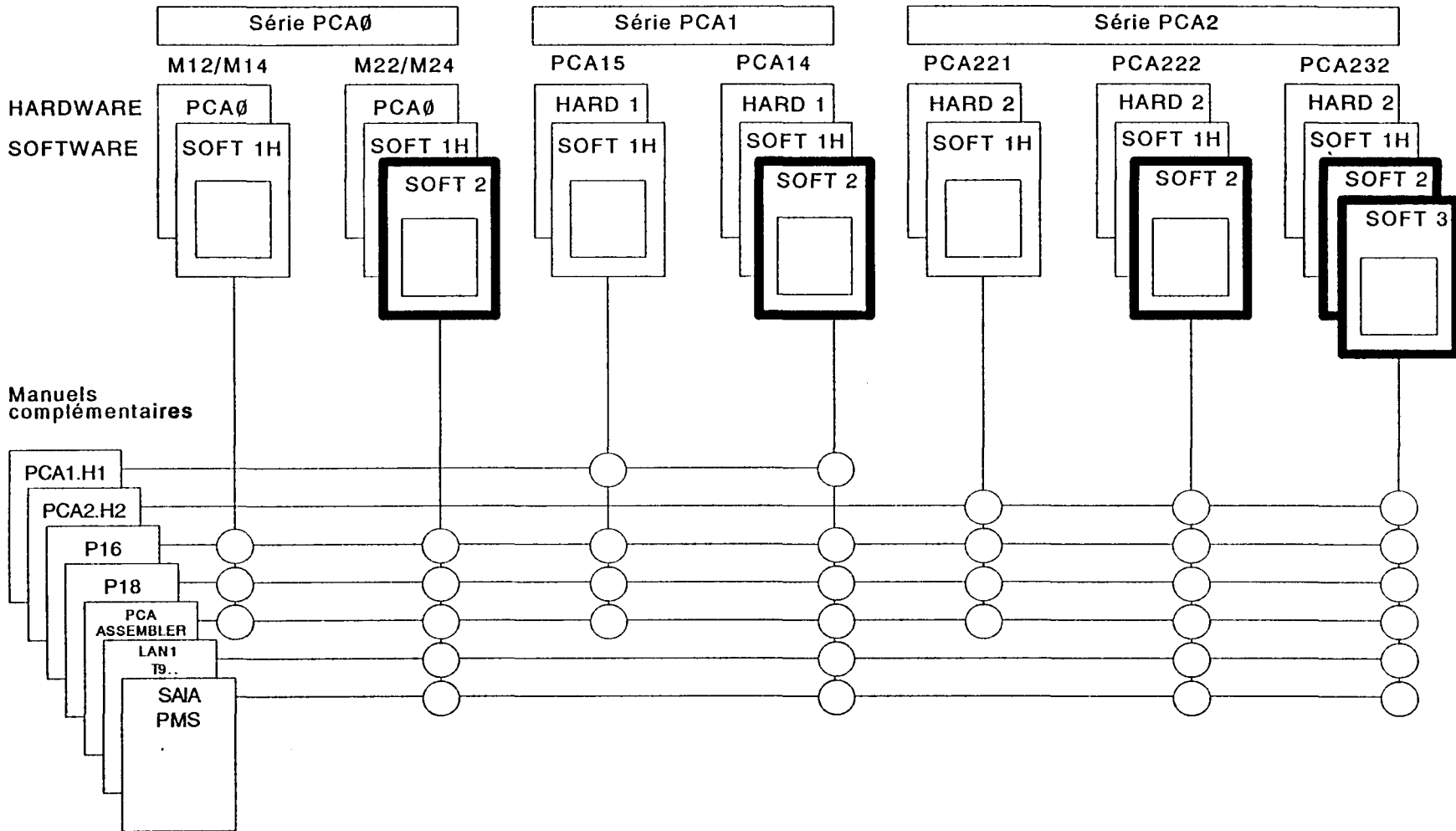
Des différences peuvent malgré tout surgir en cas d'emploi de modules d'entrées/sorties complexes.

Notez aussi qu'il existe aussi différentes version de processeur à chaque nouvelle amélioration correspond une nouvelle "firmware-version" qui peut, le cas échéant, améliorer les performances de votre automate programmable. La représentation SAIA de votre pays vous donnera toute informations nécessaires.

Cette compatibilité vers le haut est bâtie comme suit:

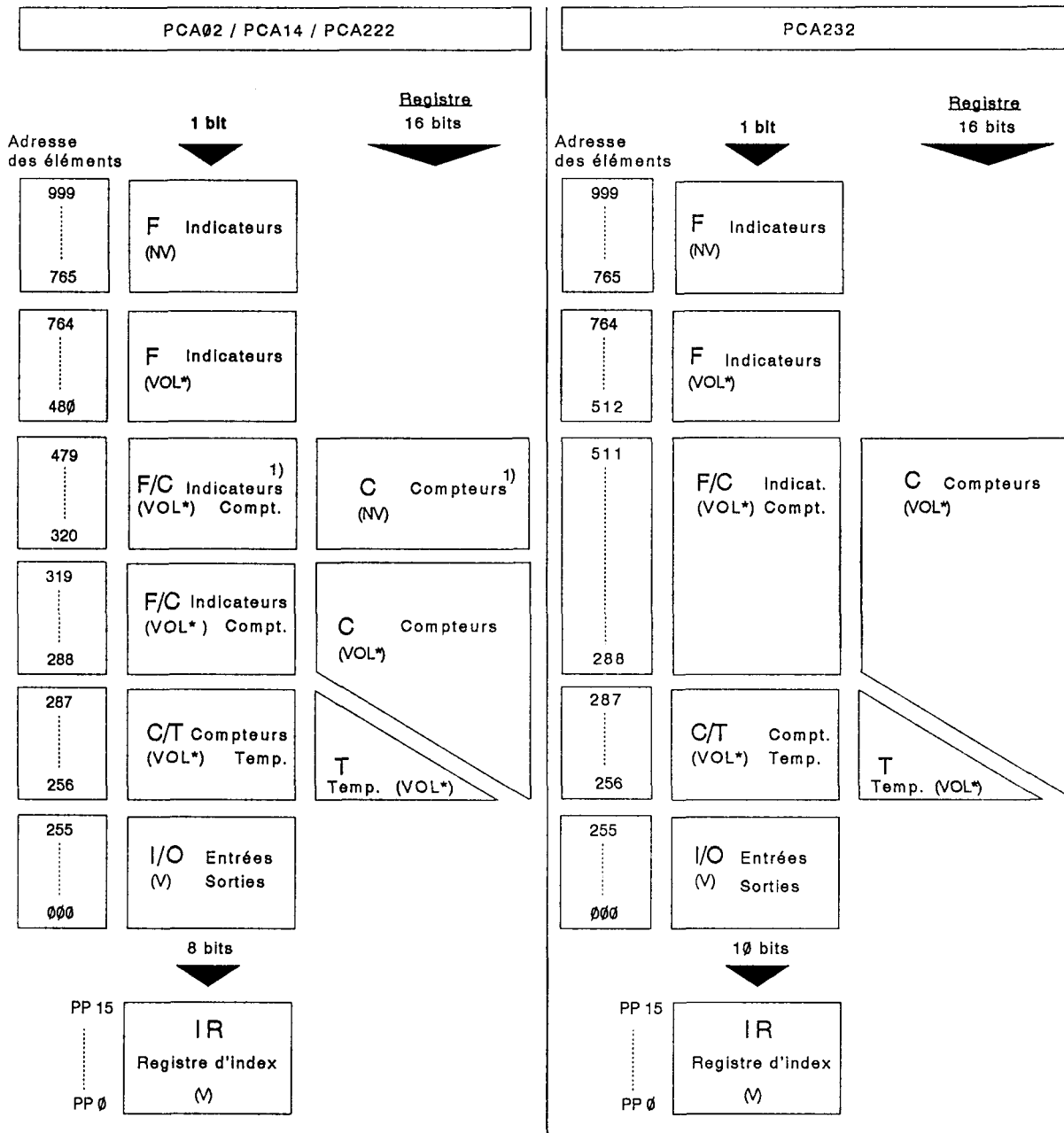
Niveau	Série PCA0	Série PCA1	Série PCA2
③ Instructions du processeur de mot			PCA232 (231)
② Instructions additionnelles	PCA02 ← → PCA14		PCA222
①H Instructions complémentaires	PCA01 ← → PCA15		PCA221
① Instructions de base		(PCA13) ← → (PCA210)	

Niveau d'"intelligence"	Série PCA0	PCA1	PCA2
Niveau du logiciel ③ Niveau de logiciel 2 + 32 instructions de mots pour . arithmétique \pm 9 digits . transfert de données . horodateur hardware			Mémoire utilisateur: 8K pas de programme  + 8K caract. de texte/données  + 8K octets de données PCA232 256 ou 512 E/S
Niveau du logiciel ② Niveau du logiciel 1H + interface sériele + horodateur + registre de données + instructions paramétriques (soft-interrupt, FIFO, PID)	Seulement OEM:   PCA0.M12 PCA0.M14 Mémoire utilisateur: max. 4K pas de programme max. 4K caractères de texte/données	    PCA141 PCA146 PCA147 + C45 32(56) 64(112) 128(4) E/S Mémoire utilisateur: max. 8K pas de programme max. 8K car. de texte/données	Mémoire utilisateur: max. 8K pas de programme  max. 8K caract. de texte/données  PCA222 256 ou 512 E/S
Niveau du logiciel ①H Niveau du logiciel 1 + arithmétique + fonctions de transfert + check-sum	Standard:   PCA0.M12 PCA0.M14 24/32 E/S 48/64 E/S Mémoire utilisateur: 4K pas de programme		Mémoire utilisateur: 8K pas de programme   PCA221 256 ou 512 E/S
Niveau du logiciel ① 32 instructions de base pour . fonctions de temporisation . fonction de comptage . registre d'index . sous-programmes . programmes parallèles		  PCA130 PCA135 32(56) 64(1 E/S Mémoire utilisateur: max. 2K pas de programme	Mémoire utilisateur: 2K pas de programme   PCA210 256 ou 512 E/S



G 2 Vue d'ensemble des manuels de la famille PCA

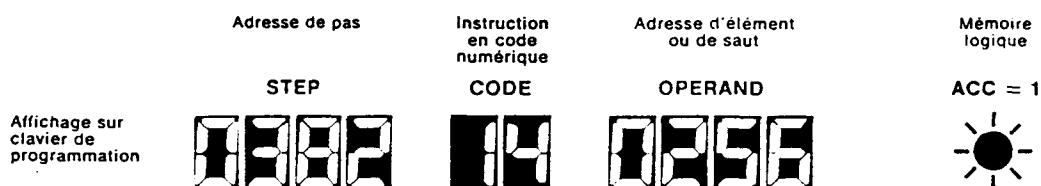
G 3 Structure des registres du processeur avec les niveaux ② et ③



(V) volatils
 (NV) non volatils
 (VOL*) volatils, commutables non volatils avec le pont NVOL

1) Augmentation du domaine des registres compteur jusqu'à C479 dès les versions firmware suivantes PCA02: V6.130, PCA14: V6.034, PCA222: V6.230.

G 4 Jeu d'instructions du SAIA®PLC niveau de logiciel ① (niveau de base)

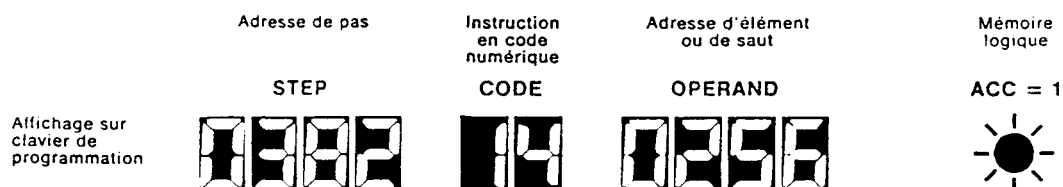


		Code numérique	Code mnémotechnique	Instruction en anglais	Description
	Instructions de combinaison logique	Ø1	STH	Start High	Début d'une combinaison logique avec élément scruté sur { High Low
		Ø2	STL	Start Low	
		Ø3	ANH	AND High	Fonction ET entre l'accumulateur et l'élément scruté sur { High Low
		Ø4	ANL	AND Low	
		Ø5	ORH	OR High	Fonction OU entre l'accumulateur et l'élément suivant scruté sur { High Low
		Ø6	ORL	OR Low	
		Ø7	XOR	Exclusive OR	Combinaison OU exclusif avec l'élément
		Ø8	NEG	Negate Accu	Complémente l'accu (résultat de la comb.)
		Ø9	DYN	Dynamic Control	Dynamise la combinaison logique (l'accu n'est influencé qu'au 1er cycle)
	Instructions de commutation	1Ø	OUT	Set Output with Status of Accu	Positionne la sortie ou l'indicateur avec l'état de l'accu
		11	SEO	Set Output	Enclenche la sortie ou l'indic. avec mémorisat.
		12	REO	Reset Output	Déclenche la sortie ou l'indicateur
		13	COO	Complement Output	Scrute la sortie ou l'indicateur et le positionne dans l'état contraire
	Instructions de temporisation et de comptage	14*	STR*	Set Timer	Positionne le temporisateur sur la valeur choisie et le fait partir
		15*	SCR*	Set Counter	Positionne le compteur sur la valeur choisie
		17	INC	Increment Counter	Augmente } la position du compteur de 1
		18	DEC	Decrement Counter	
	Instructions de saut	2Ø	JMP	Unconditional Jump	Saute à une adresse de pas, sans condition
		21	JIO	Jump if Accu is One	Saute si { accu = 1 accu = Ø sur adresse de pas
		22	JIZ	Jump if Accu is Zero	
		23	JMS	Jump to Subroutine	Saute dans un sous-programme
		24	RET	Return from Subrout.	Revient d'un sous-programme
	Instructions d'attente	25	WIH	Wait if High	Attend aussi longtemps que l'élément est sur { High Low
		26	WIL	Wait if Low	
	Instructions auxiliaires	ØØ	NOP	No Operation	Sans effet
		19	SEA	Set Accu	Positionne l'accu = 1
		16	SEI	Set Index	Pos. le registre d'index sur la valeur choisie
		27	INI	Increment Index	Augmente } le registre d'index de 1
		28	DEI	Decrement Index	
		29**	PAS**	Program Assignment	Assigne un programme parallèle
		3Ø	DOP	Display Operand	Affiche un opérande choisi
		31	DTC	Display Timer or Counter	Affiche la position d'un temporisateur ou d'un compteur

* Instruction à 2 lignes (la deuxième contient la valeur choisie)

** Instruction à 2 lignes (la deuxième contient l'adresse de début du programme)

G 5 Instructions complémentaires du niveau de logiciel (1H)

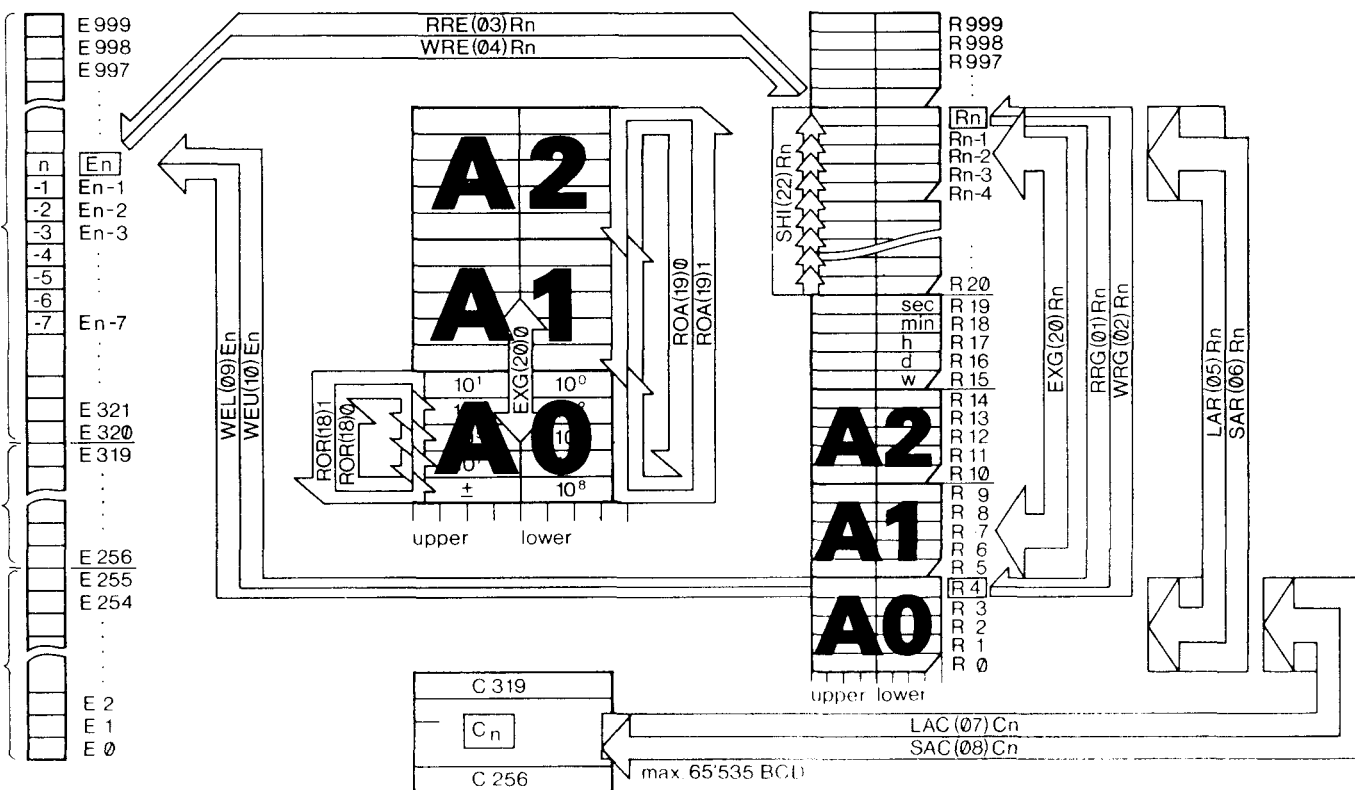


	Code mnémo.	Code num.	Instruction en anglais	Description
Instructions de transfert	STR SCR	14	Set Timer	
		15	Set Counter	
		19	} 2ème ligne	Lire 5 x 4 bits BCD
		20		Sortir 5 x 4 bits BCD
		21		Sortir 8 bits binaires
		22		Sortir 12 bits binaires
		23		Sortir 16 bits binaires
		24		Lire 8 bits binaires
		25		Lire 12 bits binaires
		26		Lire 16 bits binaires
		31		Trans. compteur -->compteur ou reg. d'index -->compteur
Instructions arithmè- tiques	SCR	15	Set Counter	
		27	} 2ème ligne	Additionne +
		28		Soustrais -
		29		Multiplie x
		30		Divise :
Instructions d'indexage	SEI	16	Set Index	Positionne registre d'index sur la valeur choisie
	INI	27	Increment-Index	Augmente < le registre
	DEI	28	Decrement-Index	Diminue < d'index de 1
	Code mnémo.	Code num.	Operand	Description
Instructions spéciales (à deux lignes)	PAS	29	18	Modification du nombre de prog. parallèles actifs
	PAS	29	30...38	Check-Sum

G 6 Vue d'ensemble du jeu d'instruction élargi du processeur de bits niveau (2)

Code num.	Code mnémo. (2e ligne)	Opérand	Description		Description Page
00	NOP	1111	Activation forcée de la sortie série		1H
29	PAS (00)	16 0	Instructions PAS Exécution exclusive d'un programme parallèle (avec blocage du déroulement des temporisateurs, de l'horodateur et de la ligne série) 2e ligne toujours 0	Instructions PAS à 2 lignes	2H
29	PAS (00)	17 0	Fin de l'exécution exclusive 2e ligne toujours 0		2H
29	PAS (00)	19 0	Fin de gestion d'interrupt 2e ligne toujours 0		5H
29	PAS (00)	23 0...818	Instruction pour la transmission d'un texte 2e ligne indique le numéro de début du texte, la transmission s'effectuera jusqu'au signe <NUL> dans le texte		5H
29	PAS	24	Envoi immédiat d'un télégramme		5H
29	PAS (00)	30 0	Contrôle du programme système par <check-sum>		6L
29	PAS (xx)	31...38 xxxx	Contrôle du programme utilisateurs par <check-sum> (31...38 = 1erK...8eK) 2e ligne contient la somme de référence		
29	PAS (xy)	50 Cn	Echanges de données entre horodateur et registre de compteurs xy indique la fonction à effectuer: Cn indique le temporisateur/compteur contenant la valeur à inscrire dans le registre de l'horodateur resp. dans lequel la valeur lue dans le registre de l'horodateur doit être déposée.		6H
29	PAS	54	Echange de données entre registre de comptage et mémoire de texte		7H
29	PAS	56	Echange de données entre registre de comptage et mémoire de donnée		7H
29	PAS	55	Echange de données entre registre de mot et mémoire de texte (PCA231 et PCA232)		12H
29	PAS	57	Echange de données entre registre de mot et mémoire de données (seulement PCA232)		12H
29	PAS	58	Echange de données entre module de mémoire RAM (PCA1.R25) et registre de comptage (seulement PCA14)		16H
29	PAS	100	Instruction à 10 lignes définissant les paramètres de transmission de la ligne série et le mode T, E, C, N ou P	Instructions PAS à 10 lignes	58K
29	PAS	190	Instruction pour la gestion d'interrupt (réaction très rapide due au travail exclusif en tant que programmes parallèles 15)		20H
29	PAS	200 : 212	Instruction définissant les paramètres pour effectuer 32 boucles de régulation PID		28H
29	PAS	250	Instruction pour la définition et la gestion d'un registre à décalage ou rotation		41H
29	PAS	251	Instruction pour la définition et la gestion d'un registre FIFO		41H

G 7 Vue d'ensemble du processeur de mots niveau ③ (seulement PCA23)
 NOP 1248 provoque le changement du processeur de bits au processeur de mots



Code num.	Code mnémo.	Opérand	Instruction en anglais	Description	Format des données
-----------	-------------	---------	------------------------	-------------	--------------------

Instructions de transfert de données

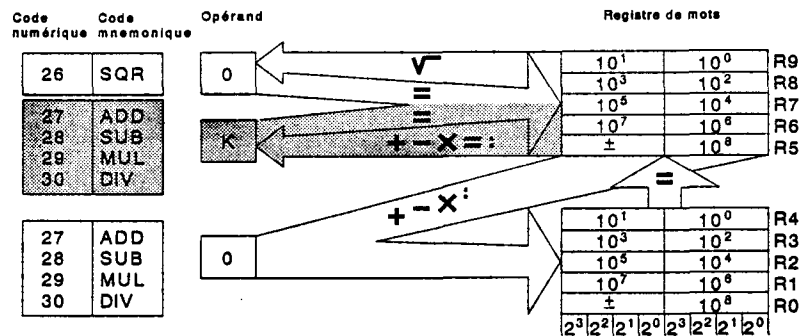
01	RRG	Rn	Read Register	Lire le mot en Rn et le mémoriser dans R4	8 bits	i
02	WRG	Rn	Write into Register	Ecrire en Rn le mot R4	8 bits	i
03	RRE	Rn	Read Register and write in Elem.	Lire le mot en Rn et le mémoriser dans les 8 éléments En...En-7, adressés par A1	8 bits	i
04	WRE	Rn	Write Register with Elements	Ecrire en Rn le contenu de 8 éléments En...En-7 adressés par A1	8 bits	i
05	LAR	Rn	Load A0 with Registers	Chargement de A0 avec le bloc de reg. Rn	5x8 bits	i
06	SAR	Rn	Store A0 into Registers	Mémorisation de A0 dans le bloc de reg. Rn	5x8 bits	i
07	LAC	Cn	Load A0 with Counter	Chargement de A0 avec le compt./temp. Cn	BCD	i
08	SAC	Cn	Store A0 into Counter	Mémorisation de A0 dans le compteur Cn	BCD	i
09	WEL	En	Write Elements with { Lower } Digit	Ecrire dans les éléments { R4 (10 ⁰)	4 bits	i
10	WEU	En	Write Elements with { Upper } Digit	EN...EN-3 le digit de { R4 (10 ¹)		

Traitement des registres arithmétiques

18	ROR	0/1	Rotate Register	Rotation R4...R3...R2...R1...R0 ou R0...R1...R2...R3...R4		
19	ROA	0/1	Rotate A	Rotation A0...A1...A2 ou A2...A1...A0		
20	EXG	0 Rn	Exchange A1 with A0 Exchange A1 with Rn...Rn-4	Echange A1 avec A0 Echange A1 avec Rn...Rn-4	5x8 bits 5x8 bits	i

(i) = indexable

Code numér.	Code mnémon.	Opérand	Instruction en anglais	Description	Format des données
Instructions d'arithmétiques					
25	CMP	En	Compare A0 with A1	Comparaison de A0 avec A1 A1 > A0 ----> En = 1 A1 = A0 ----> En-1 = 1 A1 < A0 ----> En-2 = 1	BCD i
26	SQR	0	Square Root of A1	$\sqrt{A1}$ ----> A1, chiffres entiers seulement	BCD
27	ADD	0 K	Add A0 to A1 Add K to A1	A1 + A0 ----> A1, <carry> A1 + K ----> A1, <carry> (K = 1...2047)	BCD BCD
28	SUB	0 K	Subtract A0 from A1 Subtract K from A1	A1 - A0 ----> A1, <carry> A1 - K ----> A1, <carry> (K = 1...2047)	BCD BCD
29	MUL	0 K	Multiply A1 by A0 Multiply A1 by K	A1 . A0 ----> A1, <carry> A1 . K ----> A1, <carry> (K = 1...2047)	BCD BCD
30	DIV	0 K	Divide A1 by A0 Divide A1 by K	A1 : A0 ----> A1, reste dans A0 <carry> A1 : K ----> A1, reste dans A0 <carry> (K = 1...2047)	BCD BCD



Traitement des registres arithmétiques

14	CLA	0/1/2	Clear A	Effacement des registres A0 ou A1 ou A2	
15	LAI	K	Load A immediately	Si K ≤ 99 ----> chargement de R4 avec données Si K = 100...2047 --> charge. de A0 avec données	
16	DBN	0	Decimal to binary	Conversion de A0 BCD en binaire	
17	BND	0	Binary to decimal	Conversion de A0 binaire en BCD	

Instructions de comptage et de saut

11	INR	Rn	Increment Register	Augm. de 1 la valeur BCD de Rn et mettre <carry>	i
12	DER	Rn	Decrement Register	Dimin. au cas où le resultat est < 0	i
13	SNC	0 En	Skip if no <carry> Skip if En = 0	Sauter l'instruction suivante si { <carry> = 0 En = 0 }	i
24	SEW	0 En	Skip to EWP if no <carry> Skip to EWP if En = 0	Sauter toutes les instructions jusqu'à EWP ou NOP 1248 si { <carry> = 0 En = 0 }	i i

Instructions diverses

00	NOP	1248	-	Changement du processeur de bits au processeur de mots et positionnement ACCU = 1	
00	NOP	0	No operation	Aucune opération	
21	CLK	En	Clock source	Assignment d'une source d'impulsion de temps	
22	SHI	Rn	Shift registers	Décalage des mots entre R20 et Rn d'une adresse vers le haut	i
23	TXT	Txn	Text	Sortie de textes	i
31	EWP	0	End Word Processor	Fin du travail dans le processeur de mots	

(i) = indexable

PARTIE H JEU DES INSTRUCTIONS ADDITIONNELLES NIVEAU 2

H 1 Instructions à une ligne NOP 1111

H 2 Instructions PAS à deux lignes
PAS 16/17
PAS 19
PAS 23
PAS 24
PAS 50
PAS 54/56
PAS 55/57
PAS 58

H 3 Instructions PAS à 10 lignes
PAS 190 (et PAS 19)
PAS 200...212
PAS 250/251

PARTIE H JEU DES INSTRUCTIONS ADDITIONELLES NIVEAU 2

H 1 Instructions à une seule ligne

NOP 1111 Traitement de la ligne sérieielle forcé

L'instruction NOP 1111 provoque un traitement immédiat du récepteur de la ligne sérieielle et peut être introduit dans un programme en nombre quelconque.

L'utilisation de cette instruction est convenable quand on risque d'avoir dans le programme des instructions-bit consécutives (par exemple quelques SCR) qui ne provoquent pas un changement du PP et par conséquent pas un traitement de la ligne sérieielle.

SCR 260

31 262

SCR 260

28 281

NOP 1111; NOP 1111 provoque le traitement forcé de la ligne sérieielle

SCR 260

30 4

.

.

.

NOP 1111 est effectué dans n'importe quel état de l'ACCU et il n'a pas d'effet sur celui-ci.

H 2 Instructions PAS à deux lignes

PAS 16, PAS 17 Exécution exclusive (hors base de temps)

Pour pouvoir émettre des impulsions précises dans le temps, ou pour obtenir le traitement très rapide d'une petite partie de programme, il est souvent souhaitable que seul ce programme parallèle particulier soit actif.

PAS 16 provoque le blocage de tous les autres PP, de l'avance des temporisateurs, de la ligne série et de l'horodateur software.

PAS 17 provoque le déblocage de l'état ci-dessus, c'est-à-dire que les PP ainsi que les temporisateurs et l'horodateur redémarrent à partir de leur situation à l'instant du blocage. La ligne série est à nouveau active.

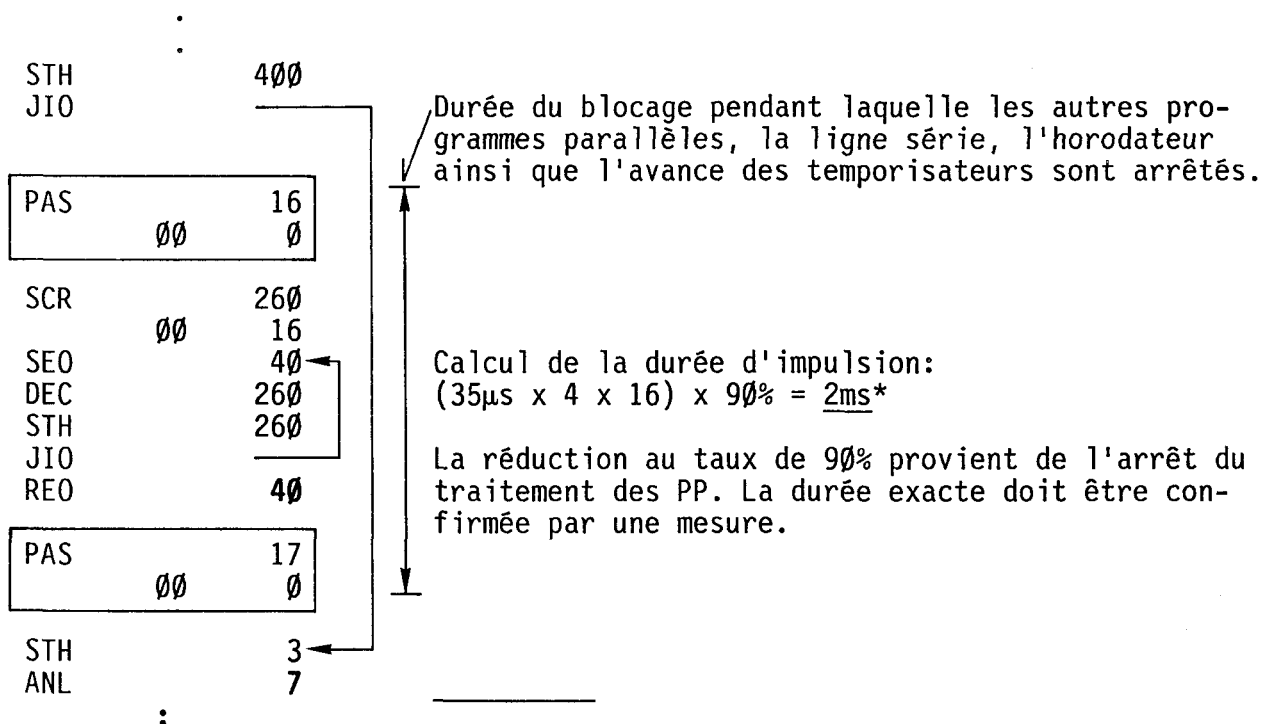
Format de l'instruction

PAS (29)	16	instruction de blocage,
00	0	la 2ème ligne étant toujours 0

PAS (29)	17	instruction de déblocage,
00	0	la 2ème ligne étant toujours 0

1. Exemple pour une impulsion de temps précise de 2ms*

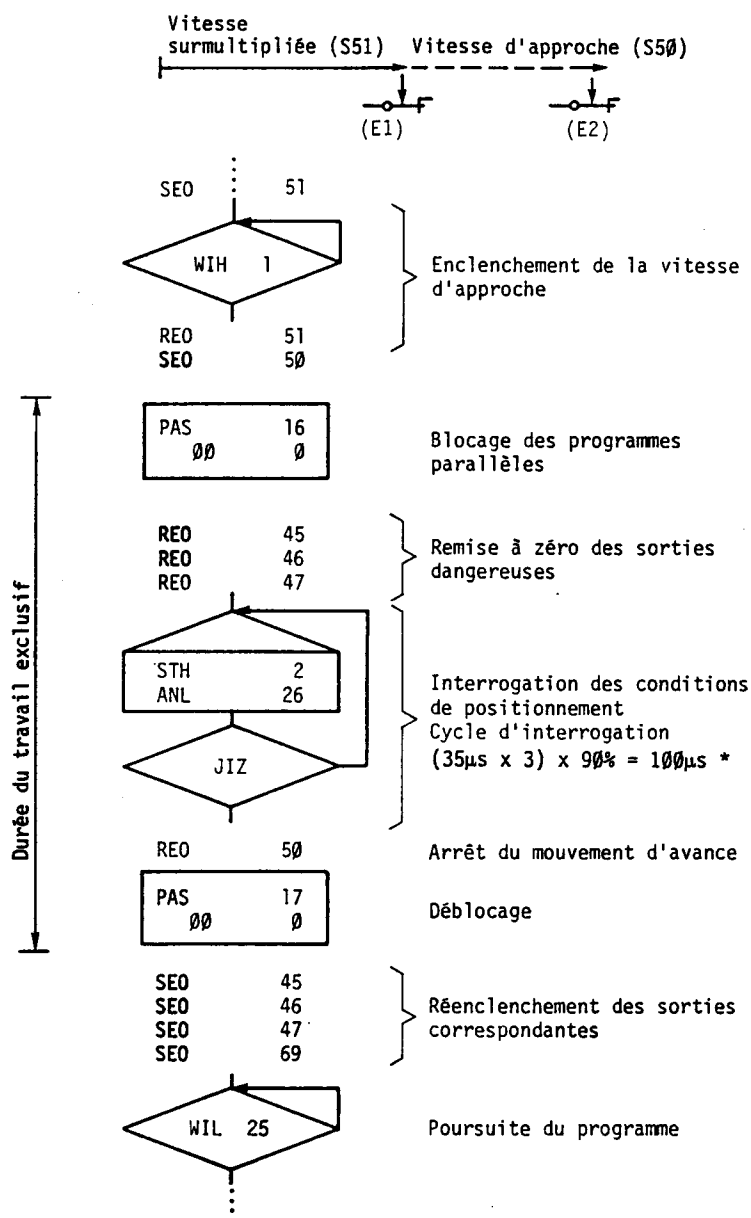
Un programme se compose de plusieurs programmes parallèles. Une boucle de programme contient la partie suivante:



*) Temps valable pour PCA232. Pour PCA02, PCA14 et PCA222 l'impulsion a une durée double.

2. Exemple pour temps de réaction très courts

En vitesse d'approche, un chariot doit être positionné avec une grande précision par un interrupteur de signalisation (E2).



Dans cet exemple, la scrutation de E2 se trouve dans une boucle à 3 instructions. Selon où se trouve le travail au moment de l'ouverture de E2, 4 à 6 instructions doivent être travaillées jusqu'au REO 50. Ce qui correspond à un temps de réaction de 125...190 μs .*

Le temps de réaction total s'obtient en ajoutant aux temps précédents les retards dûs aux interfaces d'entrée et de sortie. Dans le cas d'une alimentation stabilisée, ces retards sont cependant constants et peuvent être pris en compte dans le processus.

*) Temps valable pour PCA232. Le temps de réaction est double pour PCA02, PCA14 et PCA222.

Remarques

- Le temps de blocage des PP dépend du processus et peut durer plusieurs secondes; dans l'exemple précédent, toutes les fonctions de surveillance sont hors service pendant cette durée. Si cela doit être évité, ces fonctions doivent être incluses dans la boucle d'interrogation.
- Si un blocage dure plus de 90ms, l'horloge interne ainsi que les temporisateurs retarderont de la durée correspondante (pour une base de temps de 1/10s).
- Pendant la durée du blocage, aucun transfert d'informations séries par l'intermédiaire des interfaces n'est admis, car des erreurs de transmission pourraient en résulter.

PAS 19 Fin de gestion d'interrupt
(voir PAS 190)

PAS 23 Sortie de texte du processeur de bit
(voir le chapitre K 5 - Entrée/sortie de textes)

PAS 24 Sortie immédiate de télégrammes

L'instruction PAS 24 rend l'émission d'un caractère ou d'un télégramme indépendante de la base de temps. Cette nouvelle possibilité raccourcit le temps de réaction de la communication série.

Explication

D'une part, le programme utilisateur signifie au système que l'émission d'un caractère ou d'un télégramme est désirée en plaçant l'élément TBY sur "High". D'autre part, le programme système n'analyse l'état de l'élément TBY qu'à des instants définis, lorsque la ligne série n'est pas active en émission soit:

- chaque 1/10 resp. 1/100 ms, dépend du choix de la base de temps
- à l'émission et à la réception de caractères spécifiques: ACK, ENQ, derniers caractères de télégramme (ETX resp. BCC)
- après l'instruction de fin de routine de gestion d'interruption PAS 19
- au lancement de texte par les instructions PAS 23 et TXT (PCA 232)
- à l'exécution de l'instruction PAS 24

Utilisation

SEO	509	Positionner le TBY (Transmit Busy)
PAS	24	
00	0	Sortie immédiate du télégramme

Le PAS 24 est exécuté indépendamment de l'état de l'ACCU et ne l'influence pas.

PAS 50 Echanges entre horodateur et compteurs

Sous P05 (mode de fonctionnement "MAN BIT") nous avons vu comment lire et écrire par le P05 dans l'horodateur sous la plage d'adresse 4000...4007. Nous expliquons ici comment accéder à l'horodateur par le programme utilisateur. Ceci afin de pouvoir effectuer une remise à l'heure par le programme utilisateur, ou encore utiliser l'horodateur pour des fonctions en temps réel.

Grâce à une instruction PAS à deux lignes, l'accès à l'horodateur est possible en lecture ou écriture.

	CODE	OPERAND
1ère ligne	PAS (29)	50
2ème ligne	XX	ccc
0 ≙ écriture de la date et heure 1 ≙ lecture de la date et heure		Compteur contenant la valeur à inscrire ou recevant la valeur lue
Adresse en mode MAN BIT		Valeurs du compteur ccc
(4000)	0 ≙ semaine de l'année	1...53
(4001)	1 ≙ jour de la semaine	1...7
(4002)	2 ≙ année	0...99
(4003)	3 ≙ mois	1...12
(4004)	4 ≙ date (No jour)	1...28, 29*, 30, 31
(4005)	5 ≙ heures	0...23
(4006)	6 ≙ minutes	0...59
(4007)	7 ≙ secondes	0...59

Il est aussi possible (voir K 5.3 "Signification des signes spéciaux") de sortir par la ligne série, sous différentes formes, la date et l'heure.

Exemple:

Remise à 0 des minutes et secondes à chaque heure pleine, effectuée avec le flanc ascendant d'une impulsion de synchronisation provenant de l'entrée 7.

```

STH      7;      Entrée de synchronisation
DYN      400
JIZ
SCR      300      } Remise à 0 du compteur de référence
00      0
PAS      50      ** } Remise à 0 des secondes
07      300
SCR      300      } Mise à 30 du compteur de référence
00      30
PAS      50      ** } Mise à 30 des minutes
06      300
:

```



Partie L
Exemple 1

*) Les années bissextiles ne sont pas prises en compte dans l'horodateur logiciel, par contre un horodateur hardware en tient compte.

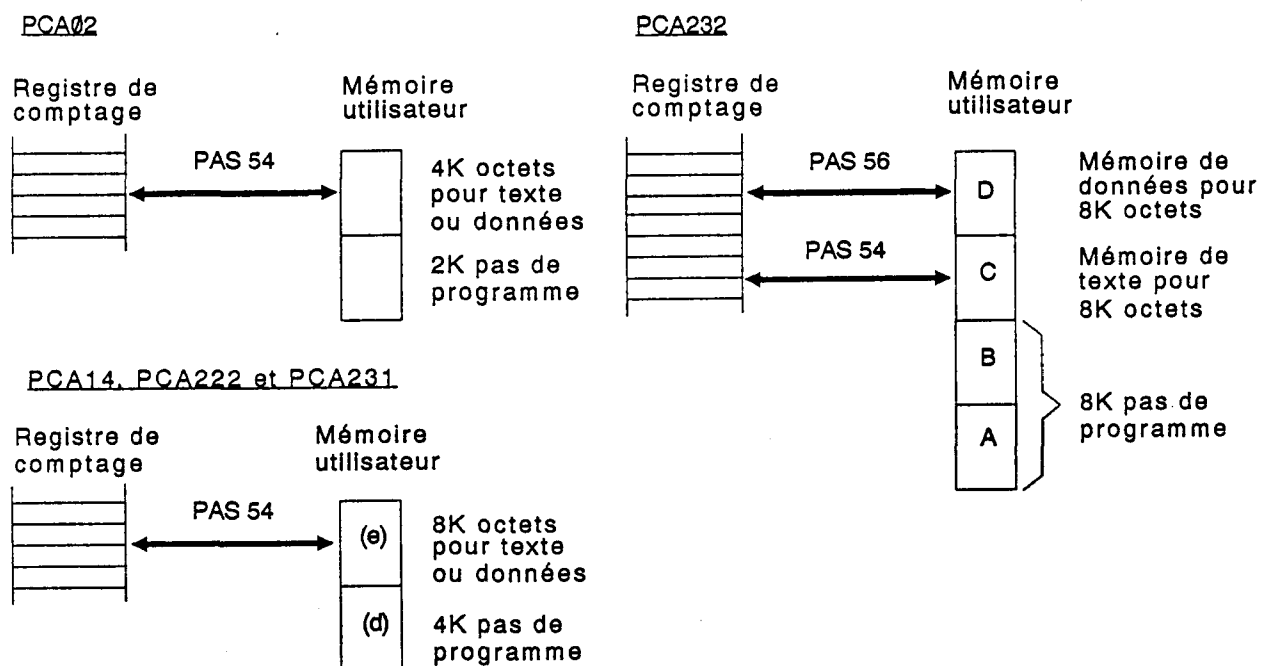
**) L'instruction PAS 50 est effectuée indépendamment du contenu de l'ACCU et ne l'influence pas.

PAS 54 Echange de données entre registre de comptage et mémoire de texte

PAS 56 Echange de données entre registre de comptage et mémoire de données

Comme le montre la vue d'ensemble ci-dessous, un nombre important de données peuvent, par ces 2 instructions être transmises au SAIA[®]PCA, puisque l'on utilise au moins une partie de la mémoire de texte. Sur le PCA232, c'est la mémoire de donnée qui est utilisée, elle est sélectionnée par l'instruction PAS 56. Afin d'assurer la compatibilité vers le haut, il est aussi possible de travailler avec la mémoire de texte du PCA232 respectivement avec l'instruction PAS 54.

Vue d'ensemble

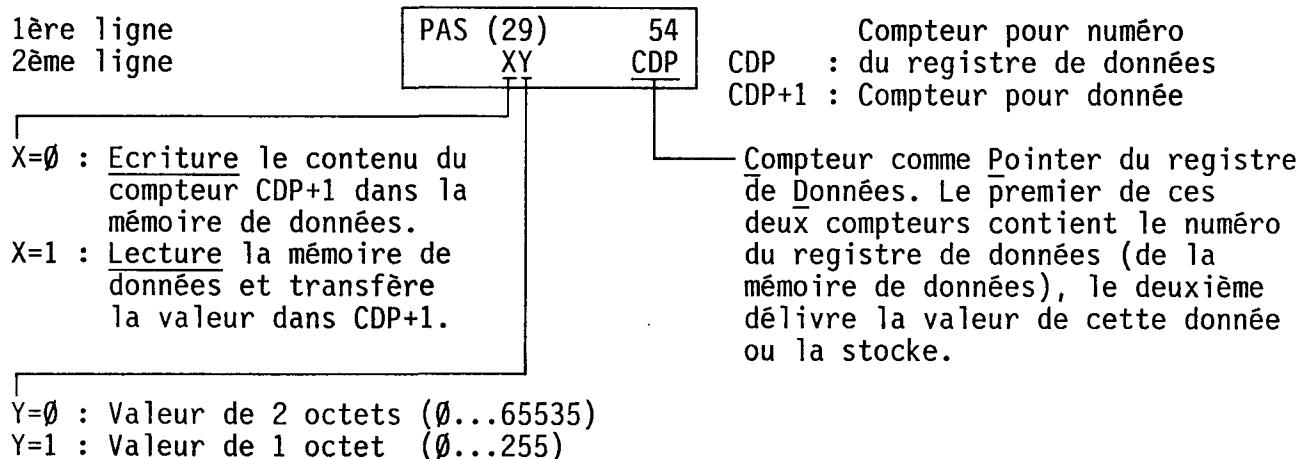


Il faut tenir compte des points suivants:

- Si l'on veut y introduire des données, il faut que le domaine mémoire du mémoire de textes resp. du registre de données soit en RAM.
- Le chapitre "Modes de fonctionnement" décrit, sous "Texte" resp. "MAN BCD", comment on peut lire ou écrire dans la mémoire de données de manière manuelle à l'aide d'appareil de programmation (voir manuels Hardware PCA1 ou PCA2).
- Dès maintenant, nous utiliserons uniquement l'expression mémoire de données en lieu et place de mémoire texte ou mémoire de données.
- Toutes les descriptions et exemples suivants se rapportent à l'instruction PAS 54. L'instruction PAS 56 peut être utilisée de manière similaire sur le PCA232.

PAS 54 Transfert entre mémoire de données et registre de comptage

Format de l'instruction:



PAS 54 est effectuée indépendamment de ACCU et ne le modifie pas.

L'exécution de l'instruction PAS 54 lance les actions suivantes:

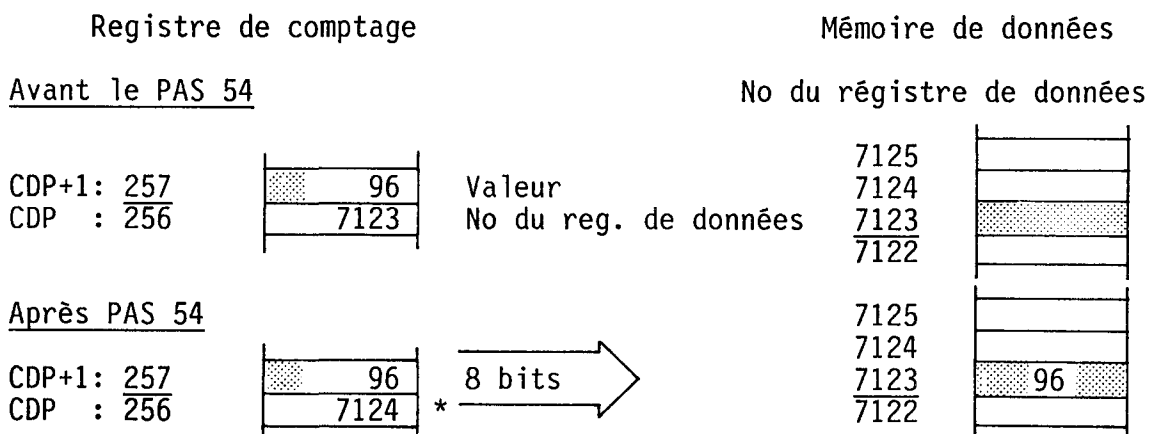
- Lors de l'écriture, le ou les 2 octets se trouvant dans CDP+1 sont transférés dans la mémoire de données à l'adresse contenue dans CDP.
- Lors de la lecture, un ou deux registres de données se trouvant dans le CDP sont transférés de la mémoire de données dans CDP+1.
- Après l'exécution, le contenu du compteur CDP est automatiquement augmenté de 1 (pour 1 octet) ou de 2 (pour 2 octets), ceci afin d'être immédiatement prêt pour le transfert suivant.

Exemples

- Transfert du contenu d'un compteur (1 octet) dans la mémoire de données

Le compteur 257 contient la valeur 96 et il faut la déposer dans un octet de la mémoire de données à l'adresse de caractère 7123.

SCR (15)	256	} Charger CDP avec le numéro du registre de données
00	7123	
PAS (29)	54	Ecrire 1 octet du CDP+1 (C257) en mémoire de données
01	256	



*) Le contenu de CDP a été augmenté de 1.

- Transfert du contenu d'un compteur (2 octets) dans la mémoire de données

Le compteur 314 contient la valeur 3280 et il faut le déposer dans la mémoire de données aux adresses de caractère 7466 et 7467.

SCR (15)	313	} Charger le CDP avec le numéro de registre de données plus haut
00	7467	
PAS (29)	54	} Ecrire 2 octets du CDP+1 (C314) dans la mémoire de données
00	313	

Registre de comptage

Avant le PAS 54

CDP+1: 314	3280	} Valeur No du registre de données
CDP : 313	7467	

Après PAS 54

CDP+1: 314	3280	} 16 bits
CDP : 313	7469	

Mémoire de données

No du registre de données

7468		} 3)
7467		
7466		
7465		
7468		} 2) (x 256) (x 1)
7467	12	
7466	208	
7465		

- Transfert d'une valeur (2 octets) de la mémoire de données dans un compteur

La valeur 56477 mémorisée dans les registres de données numéros 1930 et 1931 doit être déposée dans le compteur 275.

SCR (15)	274	} Charger le CDP avec le numéro du registre de données le plus haut
00	1931	
PAS (29)	54	} Ecrire 2 octets en CDP+1 (C275)
10	274	

¹⁾ Le contenu de CDP a été augmenté de 2.

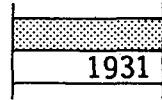
²⁾ La valeur est mémorisée sous format binaire soit (12 x 256 + 208 = 3280). Le moniteur du CPU permet, avec P10/P05, de présenter ces valeurs comme valeurs de 2 octets combinées (voir "Mode de fonctionnement").

³⁾ Pour des valeurs de 2 octets choisir si possible un nombre du registre de données impair.

Registre de comptage

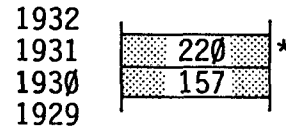
Avant le PAS 54

CDP+1: 275
CDP : 274



Mémoire de données

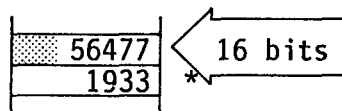
No du registre de données



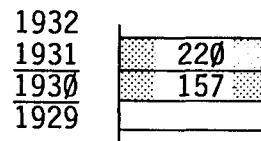
*) Valeur 56'477 = 220 x 256 + 157

Avant le PAS 54

CDP+1: 275
CDP : 274



No du registre de données



*) Le contenu de CDP a été augmenté de 2.

- Il s'agit de transférer 32 compteurs de C288...C319 avec des valeurs de 0...9999 dans la mémoire de données à partir du numéro du registre de données 4001.

SEI 0

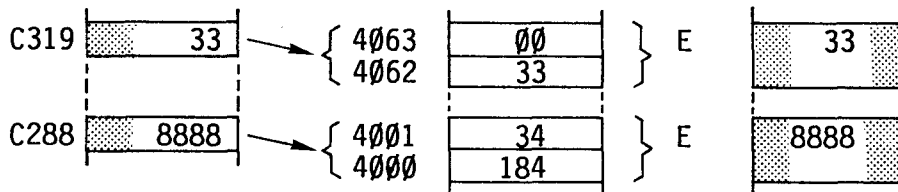
SCR 260 } Charger le compteur CDP ayant le numéro
00 4001 } de registre de données le plus bas

SCR 261 } Compteur de transfert (CDP+1)
31 1288 } pour C288...C319

PAS 54
00 260

Ecrire 2 octets de CDP+1 (C261) en mémoire de données

INI 31
JIO



Inter-
prétation
1 octet
binaire

Inter-
prétation
2 octets
binaire

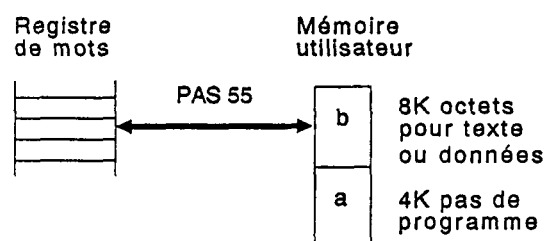
PAS 55 Echange de données registre de mot <---> mémoire de texte
(PCA231 et PCA232)

PAS 57 Echange de données registre de mot <---> mémoire de donnée
(seulement PCA232)

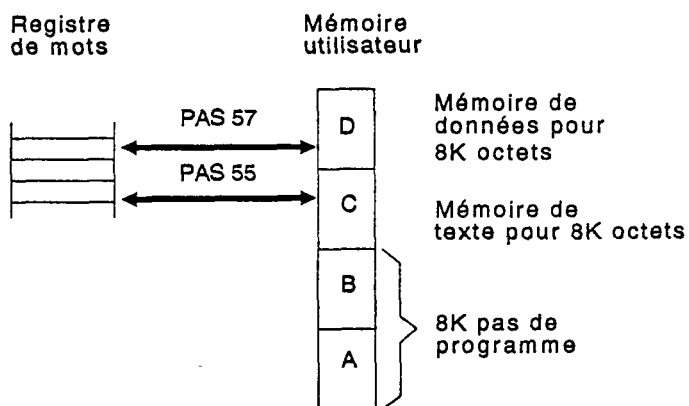
Par analogie à PAS 54/56, les instructions PAS 55/57 sont utilisées pour actionner le transfert vers le registre de mot du PCA23.

Nous décrirons ici uniquement l'instruction PAS 57. Le PAS 55 a exactement la même fonction, mais avec la mémoire de texte.

PCA231



PCA232



Format de l'instruction

1ère ligne	PAS (29)	57	CDP : Compteur pour le numéro du registre de données
2ème ligne	X0	CDP	CDP+1 : Compteur pour l'adresse du registre de mots

X=0 : Ecriture du contenu du bloc de registres (Pointer CDP+1) de la mémoire de données (Pointer CDP)

X=1 : Lecture de la mémoire de données (Pointer CDP) et transfert dans le bloc de registres (Pointer CDP+1)

Compteur comme Pointer de Données
Le premier de 2 compteurs contient le No de registre (du mémoire de données), le deuxième contient l'adresse bloc de registres de mots.

L'exécution de l'instruction PAS 57 lance les actions suivantes:

- Lors de l'écriture (X=0), le contenu du bloc de registres (5 registres = 9 BCD + signe) est transféré comme valeur 5 x 8 bits dans la mémoire de données à partir du registre de données selon CDP.
- Lors de la lecture (X=1), le contenu de 5 caractères de la mémoire de données est transféré dans un bloc de registres selon CDP+1.
- Après l'exécution, le contenu des compteurs CDP et CDP+1 est automatiquement augmenté de 5, ceci afin d'être immédiatement prêt pour le transfert suivant.

Exemples

- Transfert du contenu d'un bloc de registres de mots dans la mémoire de données

Le bloc de registres R515...R519 doit être transféré dans les registres de données 4115...4119 de la mémoire de données.

SCR (15) 280 } Charger le CDP avec le
 00 4119 } numéro du registre (MSD)

SCR (15) 281 } Charger le CDP+1 avec
 519 } l'adresse du registre de mots (MSD)

PAS (29) 57
 00 280 Transfert de la valeur du bloc de
 registre dans la mémoire de données

Avant PAS 57 Registre de
 compteur

CDP+1: 281	519
CDP : 280	4119

Registre de mots Mémoire de données

Adresse
du registre

No du registre

R519	89
R518	67
R517	45
R516	23
R515	+1

4119	
18	
17	
16	
4115	

Après PAS 57

CDP+1: 281	524 *
CDP : 280	4124 *

R519	89
R518	67
R517	45
R516	23
R515	+1

BCD
(+123'456'789)

4119	137	89
18	103	67
17	69	45
16	35	23
4115	01	01

Inter- Inter-
prétation interprétation
binaire BCD

L'indice + est pris comme 0. Les autres valeurs sont transférées telles quelles, néanmoins, elles sont interprétées de manière binaire lors d'une interrogation manuelle. Pour les conversions manuelles en format BCD, voir le chapitre "Modes de fonctionnement" (manuels Hardware PCA1, PCA2). Lors d'une transmission ultérieure en retour dans le registre de mot, ces données seront de nouveau lues en format BCD avec les valeurs originales.

*) Les compteurs CDP et CDP+1 ont été augmentés de 5.

Pour le transfert de blocs de registre, il est avantageux de choisir des adresses dont le dernier chiffre est un 4 ou un 9.

- Transfert du contenu de 5 octets de la mémoire de données dans un bloc de registres de mots

Le contenu des registres de données 4220...4224 doit être transféré dans le bloc de registres de mots R620...R624 **.

SCR (15) 280 } Charger le CDP avec le numéro
00 4224 } du registre de données le plus haut (MSD)

SCR (15) 281 } Charger le CDP+1 avec l'adresse
00 624 } du bloc de registres la plus haute (MSD)

PAS (29)	57
10	280

Transfert de la valeur de la mémoire de données dans le bloc de registres

Avant PAS 57 Registre de comptage Reg. de mots Mémoire de données

CDP+1: 281	624	R624		4224	33	21
CDP : 280	4224	R623		23	67	43
		R622		22	101	65
		R621		21	135	87
		R620		4220	153	99
					Inter- préta- tion binaire	Inter- préta- tion BCD

Après PAS 57

CDP+1: 281	629 *	R624	2 1	4224	33	21
CDP : 280	4229 *	R623	4 3	23	67	43
		R622	6 5	22	101	65
		R621	8 7	21	135	87
		R620	- 9	4220	153	99
			BCD (-987'654'321)		Inter- préta- tion binaire	Inter- préta- tion BCD

*) Le contenu des compteurs CDP et CDP+1 a été augmenté de 5.

**) Pour le traitement de blocs de registres, choisir si possible des adresses dont le dernier chiffre est un 4 ou un 9.

- Transfert de tout le registre de mots R20...R999 dans la mémoire de données à partir du registre de données 6120.

R20...R999 sont 196 blocs de registre.

SCR 289 } Charger le compteur de boucles
00 196 } avec le nombre des blocs de registre

SCR 290 } Charger CDP avec le numéro du
00 6124 } registre de données le plus bas + 4

SCR 291 } Charger CDP+1 avec l'adresse du
00 24 } registre de mots la plus basse + 4

PAS 57
00 290

Ecrire sur mémoire de données

DEC 289
STH 289

JIO

Après PAS 57

Registre de mots

R999	99
998	98
997	97
996	96
R995	95
R24	24
23	23
22	22
21	21
R20	20



Mémoire de données

7099	153	99
98	152	98
97	151	97
96	150	96
95	149	95
6124	36	24
23	35	23
22	34	22
21	33	21
6120	32	20

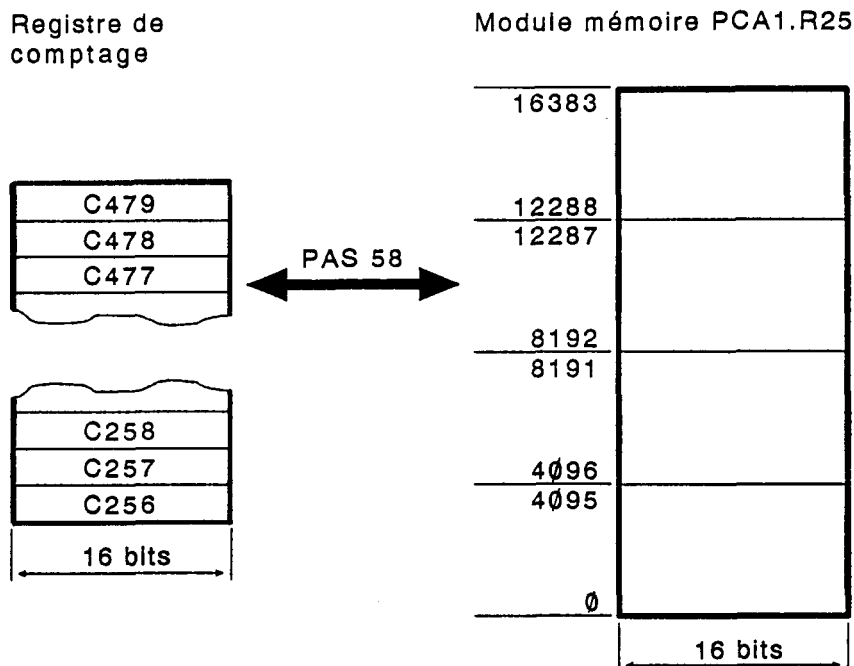
Inter-
préta-
tion
binaire

Inter-
préta-
tion
BCD

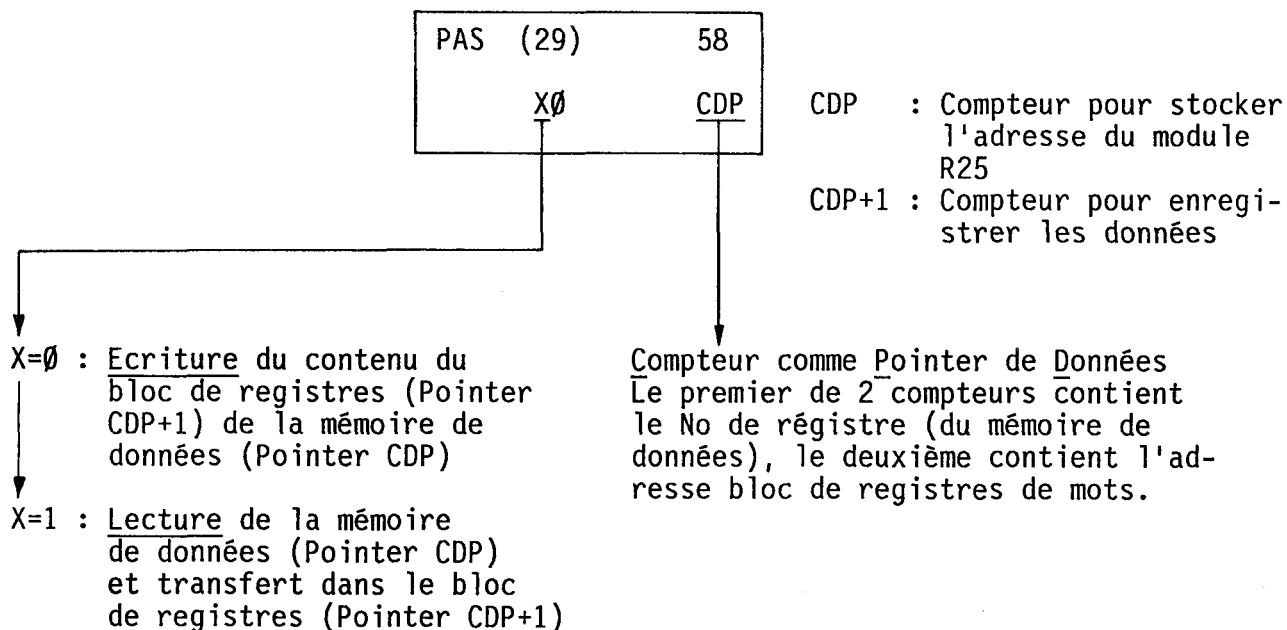
PAS 58 Echange de données entre le module de mémoire à RAM PCA1.R25 et le registre de comptage (seulement PCA14 dès version V6.036)

Le module PCA1.R25 permet de stocker 16 K mot (32 K byte) dans une mémoire à batterie tampon. L'instruction PAS 58 assure le transfert de donnée entre la mémoire de données et le registre de comptage.

Vue d'ensemble



Format de l'instruction:



PAS 58 est exécuté indépendamment de l'état de l'accumulateur et ne change pas son état.

L'instruction PAS 58 exécute les fonctions suivantes:

- Ecriture (X=0): Transfert du contenu du CDP+1 (1 mot = 2 octets) dans la mémoire du module R25; l'adresse de cette mémoire est contenue dans le CDP.
- Lecture (X=1): Transfert du contenu d'une mémoire du module R25 dans le compteur CDP+1; l'adresse de la mémoire du PCA1.R25 est contenue dans le compteur CDP.
- Après l'exécution de l'une de ces 2 opérations, le contenu du compteur CDP est automatiquement incrémenté de 1. Ainsi le pointeur (CDP) se trouve placé au bon endroit pour le prochain transfert.

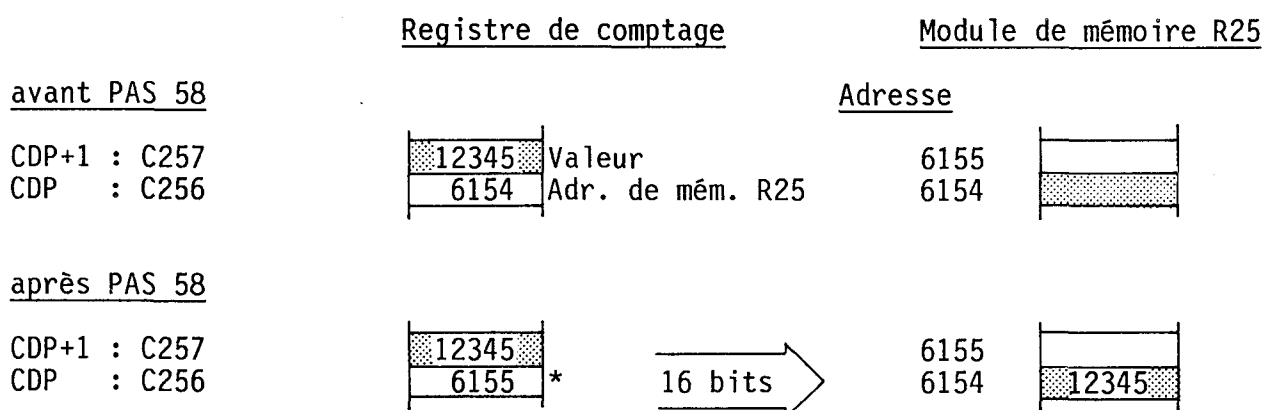
Exemples

- Le contenu du compteur C257 (ex: valeur 12345) doit être transféré dans les registres de données du module PCA1.R25 à l'adresse 6154.

SCR (15) 256 ; Charger le compteur CDP avec l'adresse
00 6154 de la mémoire du R25

PAS (29)	58
00	256

; Transfert de la valeur du CDP+1 (C257) dans le module mémoire PCA1.R25



*) La valeur du CDP est incrémentée de 1.

- Lire la valeur (par ex: 54321) du module R25, à l'adresse 14300 et la transférer dans le compteur C275.

SCR (15) 274 ; Charger le compteur CDP avec le contenu
de l'adresse du PCA1.R25
06 2012 (6 x 2048 + 2012 = 14300)

PAS (29)	58
10	274

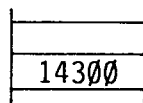
; Lire la valeur dans le module mémoire PCA1.R25
et la transférer dans le compteur CDP+1 (C275)

Registre de comptage

Module mémoire R25

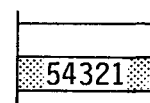
avant PAS 58

CDP+1 : C275
CDP : C274



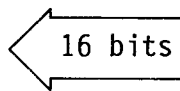
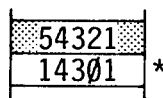
Adresse

14301
14300

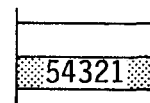


après PAS 58

CDP+1 : C275
CDP : C274



14301
14300



*) La valeur du CDP est incrémentée de 1.

- Il faut transférer les 160 compteurs suivants C320...C479 dans le module mémoire R25 à partir de l'adresse 4001.

```

SEI      0
SCR      260 ; Charger le compteur CDP avec le contenu de l'adresse
           00 4001 du module mémoire R25

```

```

→ SCR      261 ; Compteur de transfert (CDP+1) pour C320...C479
   31 1320

```

PAS	58
00	260

```

INI      159
JIO

```

- Le contenu des 160 adresses du module R25 doit être transféré dans les compteurs C320...C479.

```

SEI      0
SCR      260 ; Charger le compteur CDP avec le contenu de
           00 4001 l'adresse du R25

```

PAS	58
10	260

```

→ SCR      1320 ; Transférer le contenu du compteur intermédiaire
   31 261      (CDP+1) sur les compteurs C320...C479

INI      159
JIO

```

H 3 Instructions PAS à dix lignes

PAS 190 et PAS 19 Gestion d'interrupt (IM = Interrupt Management)

Principe de fonctionnement

Par un élément, généralement une entrée, un programme d'interrupt peut être activé exclusivement. Ce programme de gestion d'interrupt, assigné par PAS 190, sera automatiquement placé en même temps que PP15.

L'élément INTRQ (Interrupt Request) peut être défini actif soit sur le niveau "H" ou "L".

Lorsque l'élément INTRQ est actif et l'élément INTA (Interrupt Acknowledge, une sortie en général) est au niveau "L", au prochain changement de programme parallèle, le programme de gestion d'interrupt sera activé exclusivement, donc le travail de tous les PP est interrompu.

Dès que ce programme a terminé le travail exclusif, il faut libérer tous les programmes parallèles assignés précédemment par PAS 19 afin de revenir à la situation normale.

L'instruction PAS 190 est une instruction à 10 lignes qui est placée en début de programme comme l'assignation.

1	PAS	(29)	190	:	Assignation d'un interrupt
2		00	ssss	*	Adresse du début du programme d'interrupt
3		00	En	:	Interrupt Request (entrée INTRQ)
4		00	000x	:	Niveau INTRQ ("H" = 1, "L" = 0)
5		00	En	:	Interrupt Acknowledge (sortie INTA)
6		00	0	:	
7		.	.	:	
8		.	.	:	
9		.	.	:	
10		00	0	:	

} Zéros

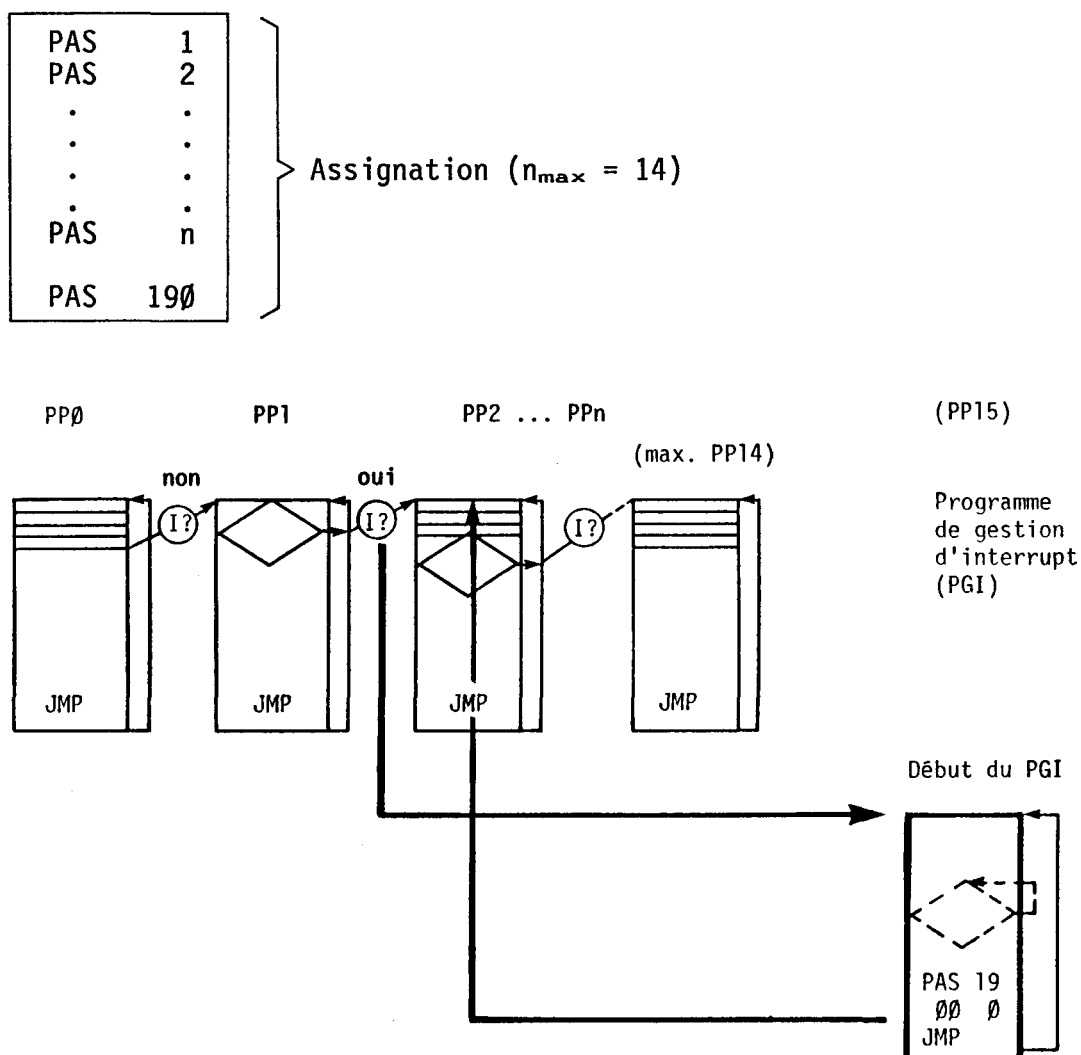
La déclaration de fin de travail exclusif se fait à l'aide de l'instruction PAS 19 (instruction à 2 lignes).

1	PAS	(29)	19	:	Fin du travail exclusif du programme d'interrupt (End ISR)
2		00	0	:	Zéro

*) L'emploi de la fonction IAD/KAD (INSERT et KILL) du programme "CI" du PCA-Assembler ne changera pas automatiquement cette adresse de pas.

Détail de fonctionnement

a) Structure de programmes



A chaque changement de programmes parallèles, les critères d'interrupt sont scrutés. Dans le cas où la scrutation reçoit un résultat positif, seul le programme de gestion d'interrupt sera travaillé.

Lorsque les conditions de retour à la situation normale sont remplies, il faut avant le PAS 19 procéder selon les besoins comme suit:

- Les programmes parallèles doivent continuer leur déroulement au point où l'interrupt est survenu.
- Certains programmes parallèles doivent recommencer au début donc doivent être réassignés.

b) Temps de réaction

La scrutation des critères d'interrupt est faite à chaque changement de programmes parallèles. Le temps de réaction entre la fermeture du "contact d'interrupt" et le départ du programme de gestion d'interrupt est pratiquement le temps de retardement de "l'entrée d'interrupt". Avec les entrées standards, au temps de 8ms, lors d'une structure de programme défavorable, il faut ajouter environ 1ms (PCA232) resp. 2ms (PCA02/14/222) pour le travail interne de l'automate.

c) Gestion des temporisateurs et interface série

Pendant le travail exclusif, la base de temps est active. De ce fait, les temporisateurs et la ligne série sont utilisables normalement. Le programme de gestion d'interrupt peut également contenir des fonctions de temps et effectuer de la transmission de textes ou d'informations en direction d'un autre système.

d) Relation entre INTRQ, INTA et le programme de gestion d'interrupt (PGI)

Pour le démarrage du programme de gestion d'interrupt, les critères suivants doivent être remplis:

- INTRQ (normalement une entrée) doit se trouver sur "H" resp. "L" selon la définition de la 4ème ligne de l'assignation d'interrupt (PAS 190).
- INTA (normalement une sortie) doit être sur "L".

INTA sera automatiquement positionné sur "H" par le programme système. Après environ 1ms, le travail exclusif du PGI commence. Le temps de travail du PGI doit être plus court que le temps entre deux changements de INTRQ de "H" à "L" à "H".

En ce qui concerne le REO INTA, 2 cas ayant des conséquences très différentes sont à considérer:

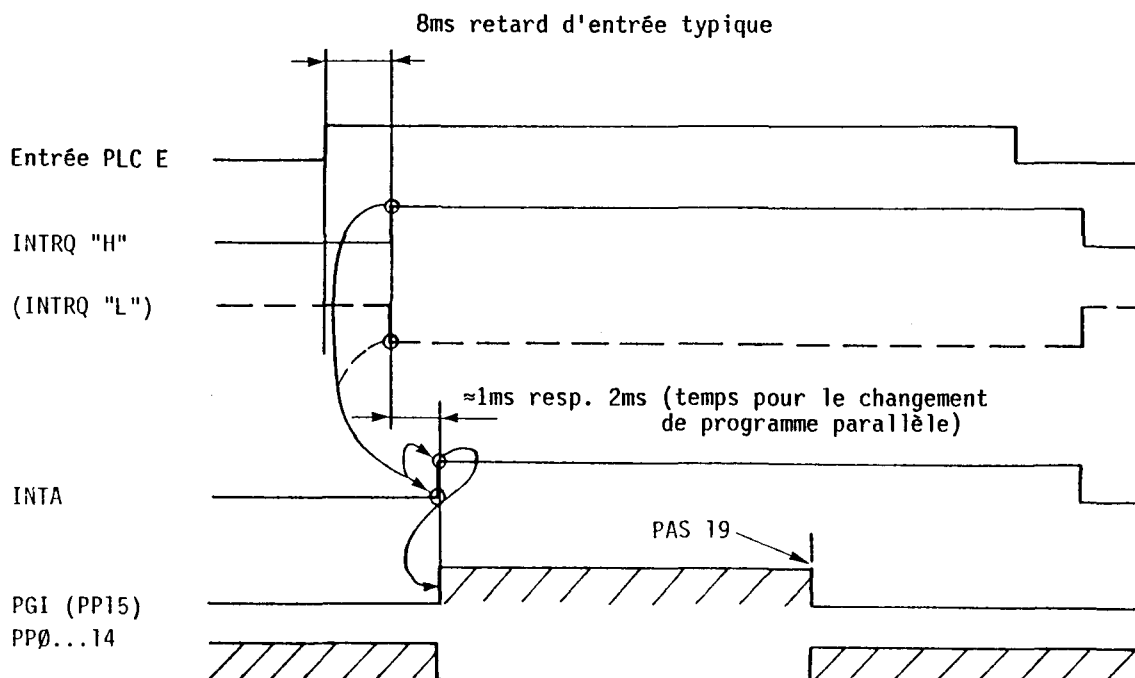
REO INTA se trouve $\left\{ \begin{array}{l} 1) \text{ à l'extérieur de PGI} \\ 2) \text{ à l'intérieur de PGI} \end{array} \right.$

1) REO INTA à l'extérieur de PGI (cas général) se trouvant dans un programme cyclique.

a) INTRQ actif "H" : STL INTRQ
REO INTA

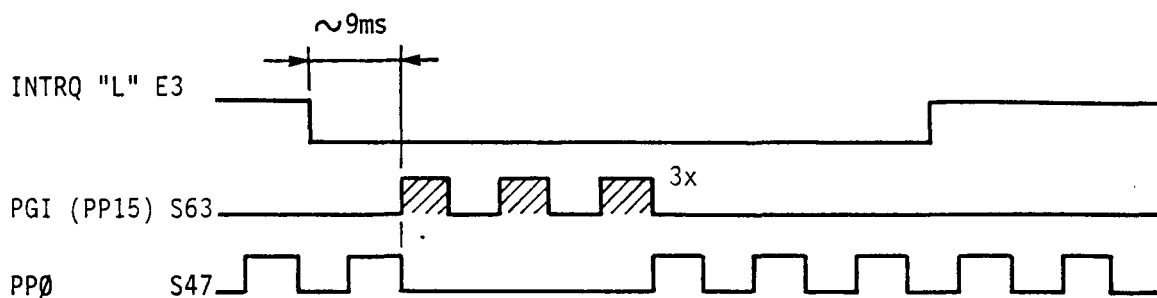
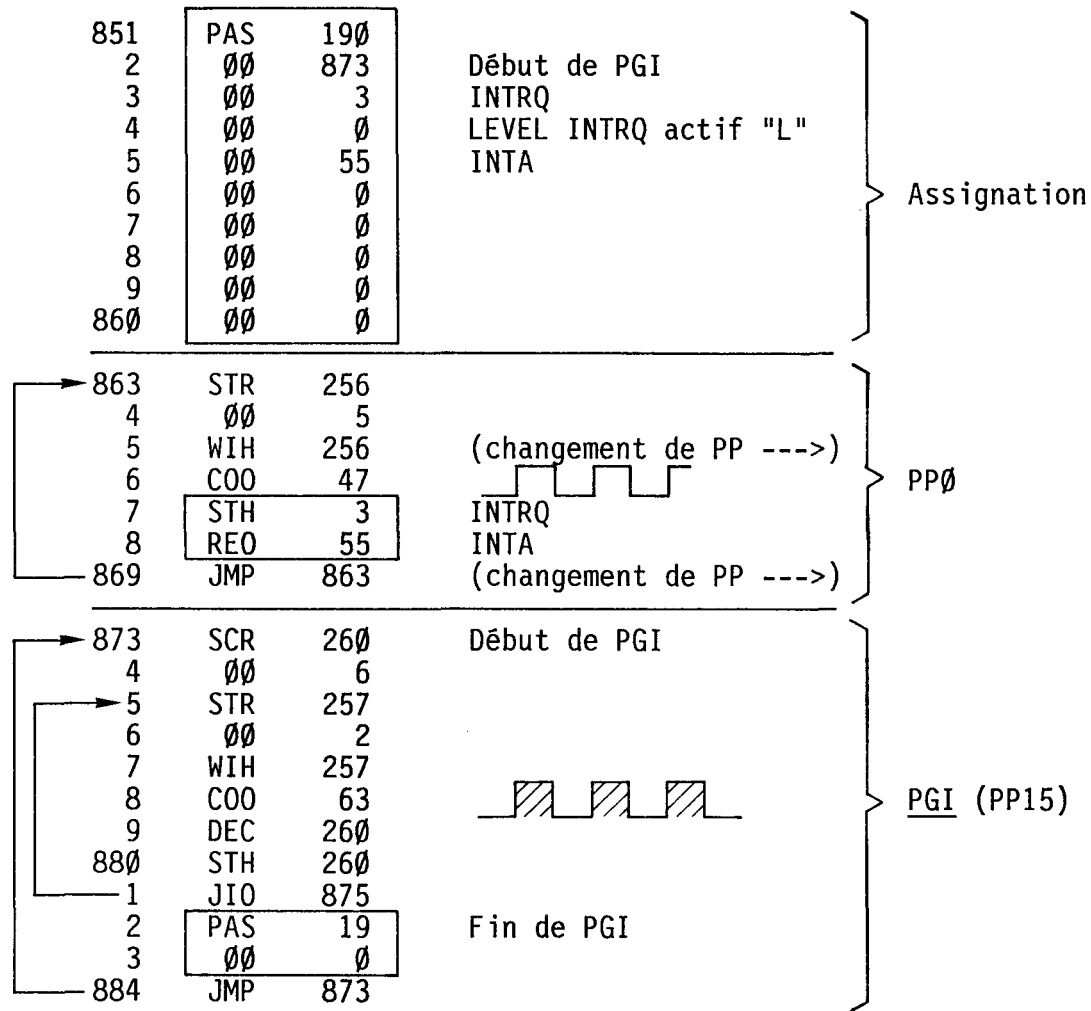
b) INTRQ actif "L" : STH INTRQ
REO INTA

Dans ce cas, le PGI ne sera travaillé qu'une seule fois lorsque INTRQ passe à "H" resp. à "L".

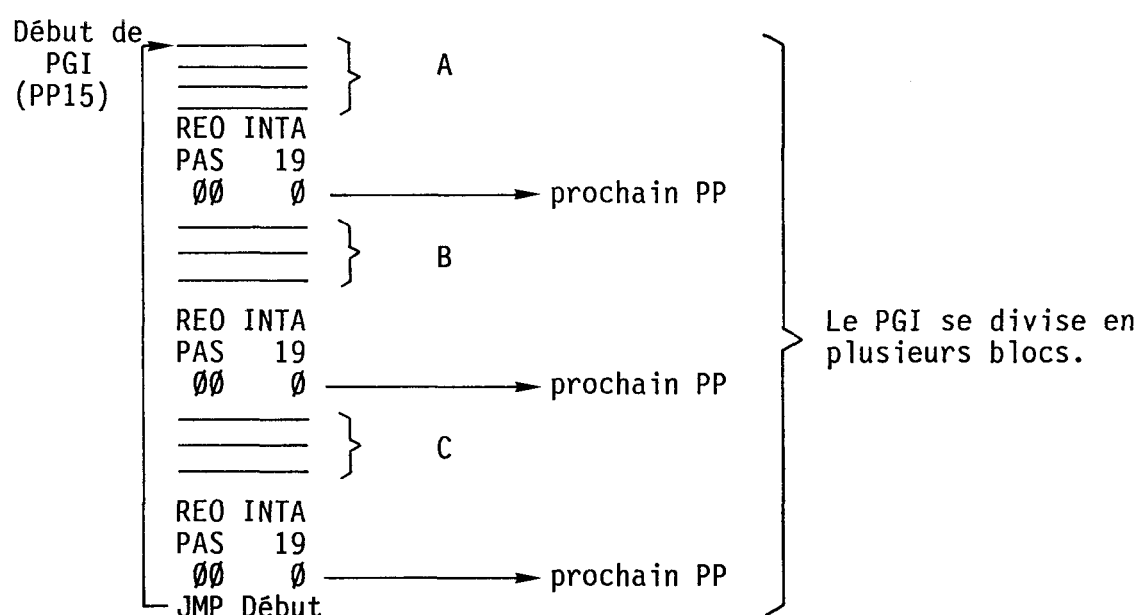
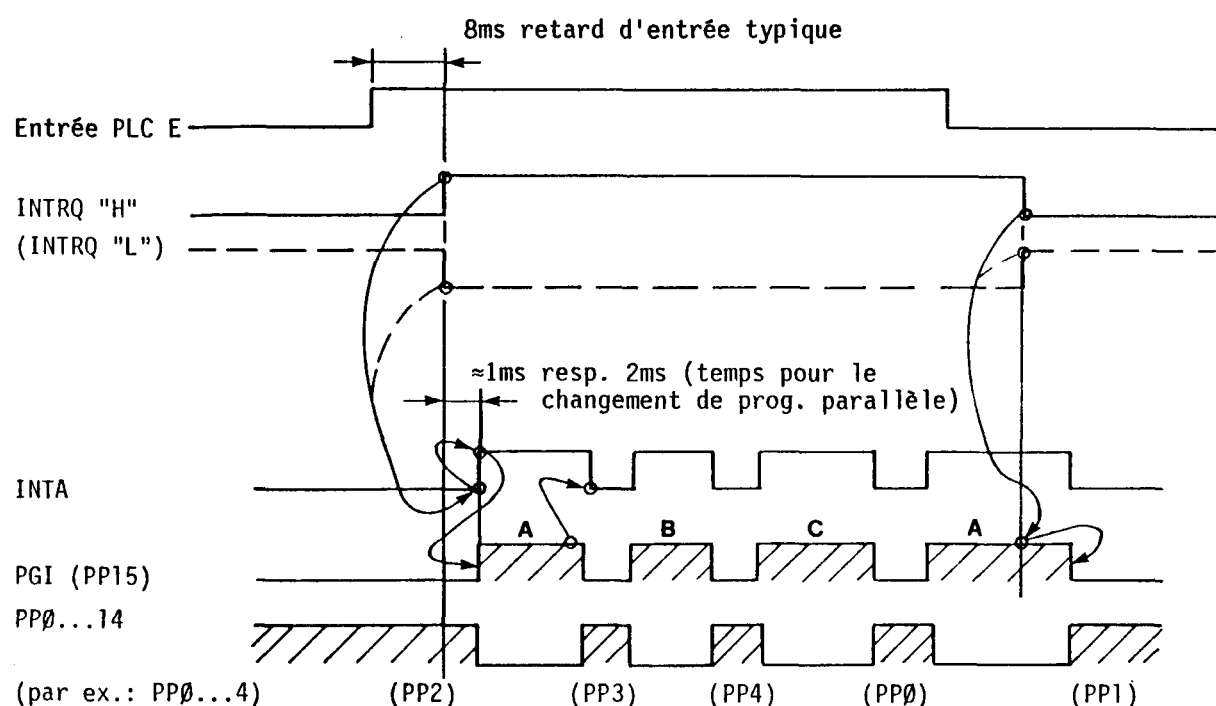


Exemple pour le travail unique et exclusif de PGI (REO INTA à l'extérieur de PGI, INTRQ actif "L")

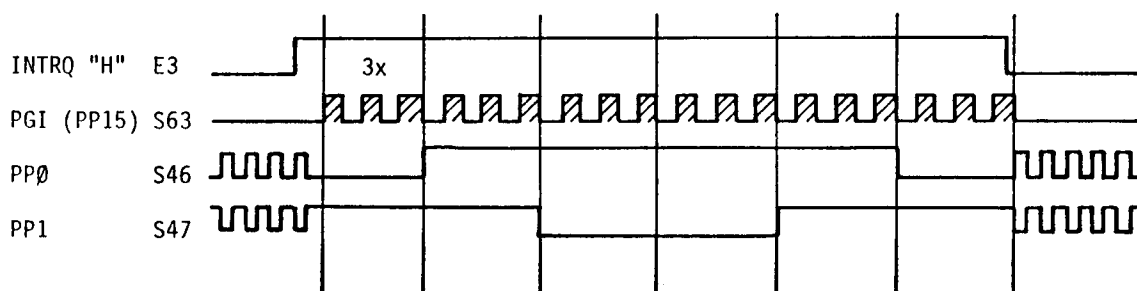
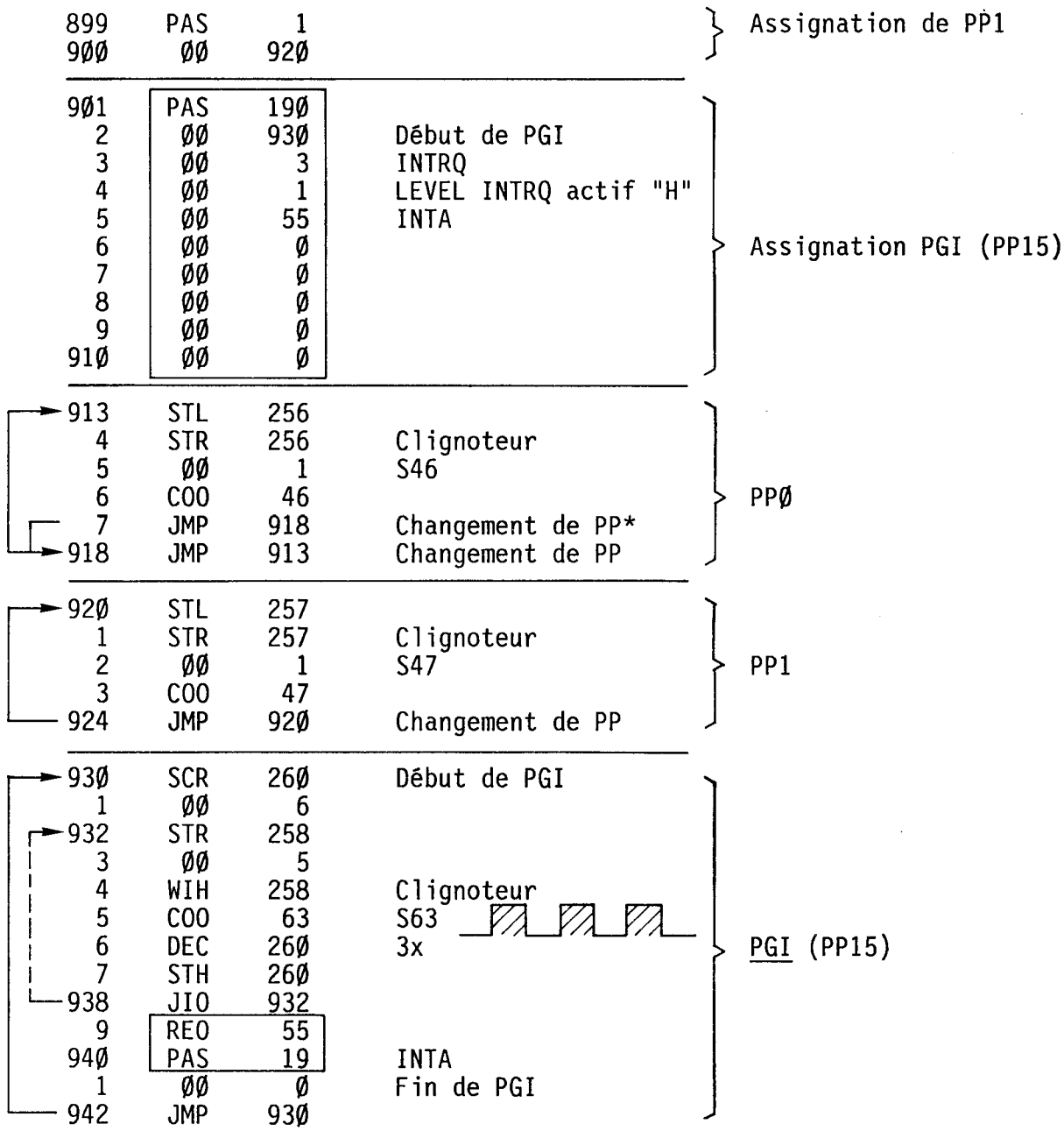
Exemple didactique:



- 2) REO INTA se situe dans le PGI (PP15), le PGI sera travaillé aussi longtemps que INTRQ sera "H" resp. "L". L'instruction PAS 19 définit la fin du travail exclusif du PGI et permet au programme parallèle suivant de se dérouler. Au prochain changement de PP, le PGI sera à nouveau travaillé. De ce fait le PGI peut se composer de plusieurs blocs qui seront traités avec de plus grande priorité que les autres programmes (par ex.: actions critiques en temps).



Exemple de traitement d'un PGI à grande priorité plusieurs fois (exemple didactique) REO INTA effectué dans le PGI, INTRQ actif "H"

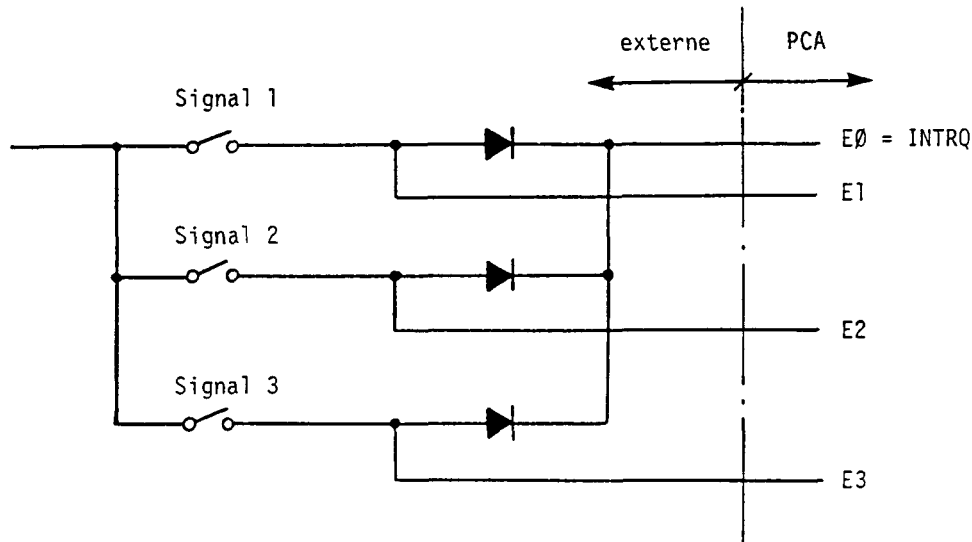


*) Par l'introduction d'un JMP artificiel, un changement de programme parallèle supplémentaire est provoqué.

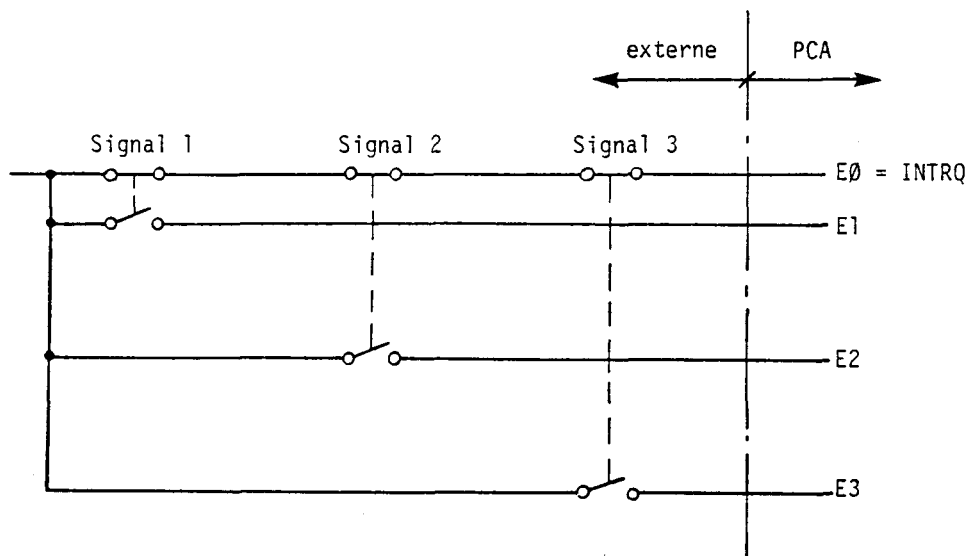
e) Priorités d'interrupts différentes

Lorsque divers niveaux de priorité ou que différents PGI doivent être exécutés selon l'entrée activée, il faut procéder comme suit:

- Pour INTRQ actif "H":



- Pour INTRQ actif "L":



Les différents signaux d'interrupt seront connectés chacun à une entrée.

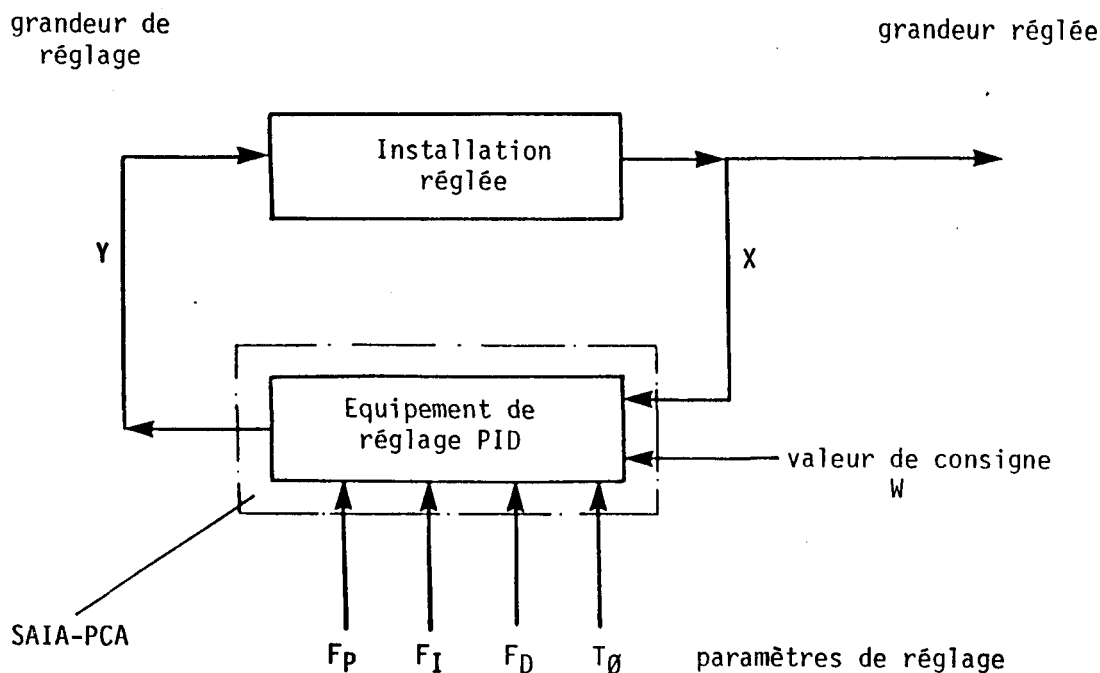
Lorsque INTRQ devient "H", une entrée recevra par l'intermédiaire d'une matrice de diodes le signal d'interrupt commun (par ex.: E0).

Lorsque INTRQ devient "L", il faut par signal d'interrupt 2 contacts. Par la liaison en série du deuxième contact de chaque signal, une entrée pour le signal d'interrupt commun est formée. Cette dernière devra être définie comme INTRQ dans le PAS 190.

Dans le PGI, les divers signaux seront scrutés selon leur priorité et permettront de travailler la partie de programme correspondante.

Généralités

L'implantation pratique de régulateurs PID requiert des connaissances techniques de régulation. Dans la description suivante, seul le principe de régulation ainsi que le traitement logiciel avec l'instruction PAS 200...212 sont mentionnés.



La figure ci-dessus représente le couplage typique d'une régulation en boucle fermée. A la sortie de l'installation réglée la grandeur réglée est ramenée à l'entrée de l'équipement de réglage et est comparée à la valeur de consigne. Le résultat de cette comparaison, influencé par le facteur de proportionnalité, les constantes de temps d'intégration et de dérivée, donne la valeur de réglage. Celle-ci est amenée à l'entrée de l'installation réglée pour la commander judicieusement.

Si l'on fait appel à un ordinateur numérique tel qu'une SAIA-PCA l'équipement de réglage se compose :

- d'une entrée analogique avec convertisseur analogue-digital
- de l'unité de calcul (CPU)
- d'une sortie analogique précédée du convertisseur digital-analogique

Chaque processeur de la série PCA a des registres (aussi appelé Databloc) pour un maximum de 32 boucles de régulation. Cela signifie que l'on peut activer jusqu'à 32 boucles de réglage pour autant que le nombre des E/S soit suffisant. Ces 32 boucles de réglage sont indépendantes les unes des autres et exécutées en technique DDC (Direct Digital Control).

Pour ce faire, le CPU du PCA utilise l'algorithme connu suivant:

$$Y_1 = F_P \times [(W - X_1) + F_I \cdot S_1 + F_D (X_0 - X_1)] \quad \text{avec } S_1 = S_0 + (W - X_1)$$

Légende:

W = valeur de consigne

X = valeur réglée

Y = valeur de réglage

F_P = facteur de proportionnalité = 1 / X_P

X_P = domaine proportionnel

F_I = facteur intégral = T₀ / T_I

T₀ = temps d'intervalle de scrutation

T_I = temps de dosage d'intégration

= T_N = temps de réponse

F_D = facteur différentiel = T_D / T₀

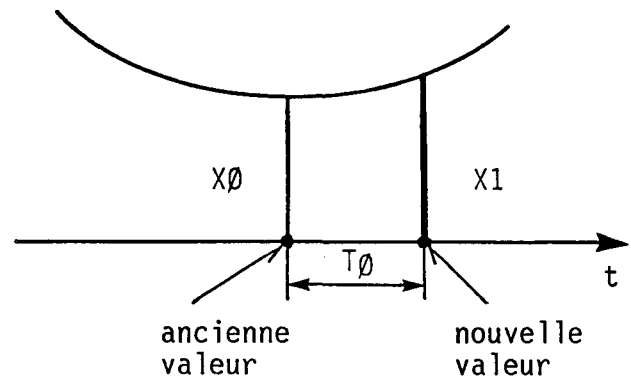
T_D = constante de temps de dérivée

= T_V = temps de retard

Les indices 0 et 1 signifient:

0 = dernière valeur travaillée

1 = nouvelle valeur



La formule ci-dessous peut aussi être utilisée avec le PCA232 (dès la version V7.120) ainsi la valeur de consigne est prise en compte plus rapidement:

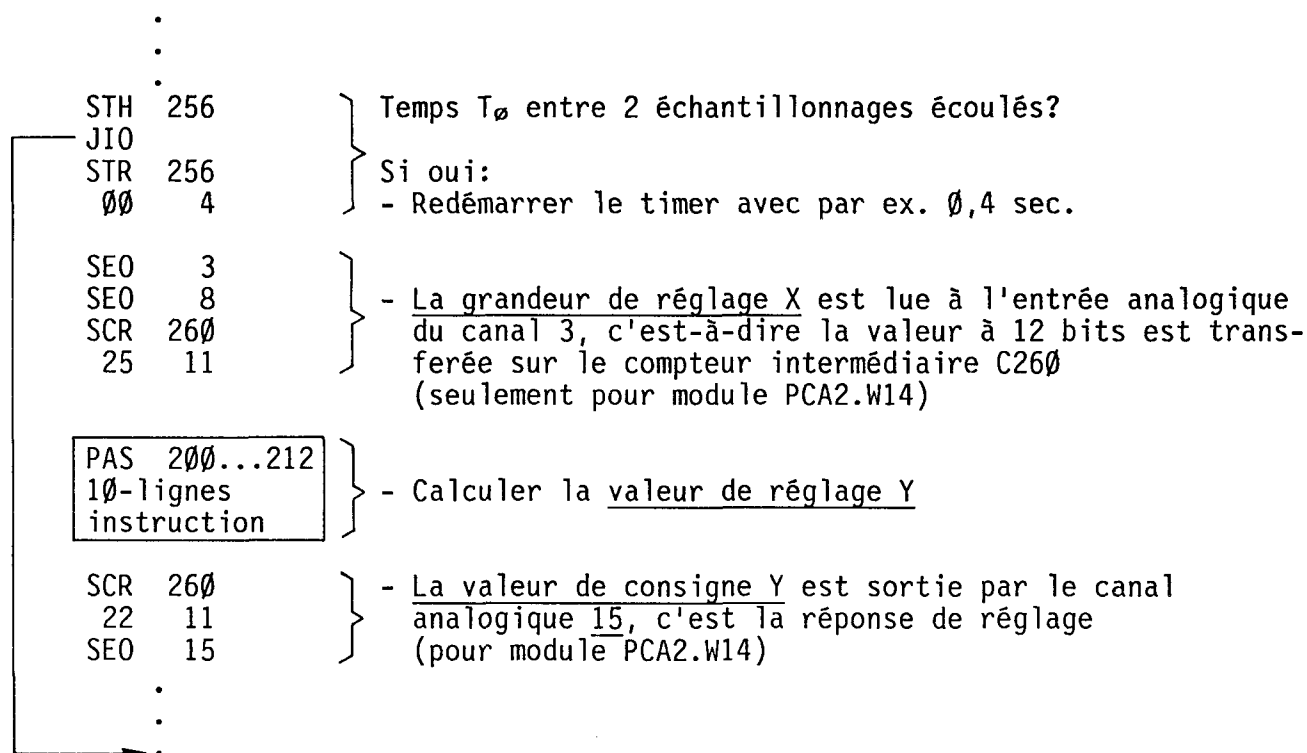
$$Y_{1W} = F_P \left\{ (W_1 - X_1) + F_I \times S_1 + F_D [(X_0 - X_1) - (W_0 - W_1)] \right\}$$

Structure du programme

Par l'utilisation des instructions paramétrisables PAS 200...212 le programme système du PCA calcule une nouvelle valeur de réglage Y_1 sur la base des anciennes valeurs X_0 , Y_0 , S_0 et des paramètres contenus dans l'instruction PAS.

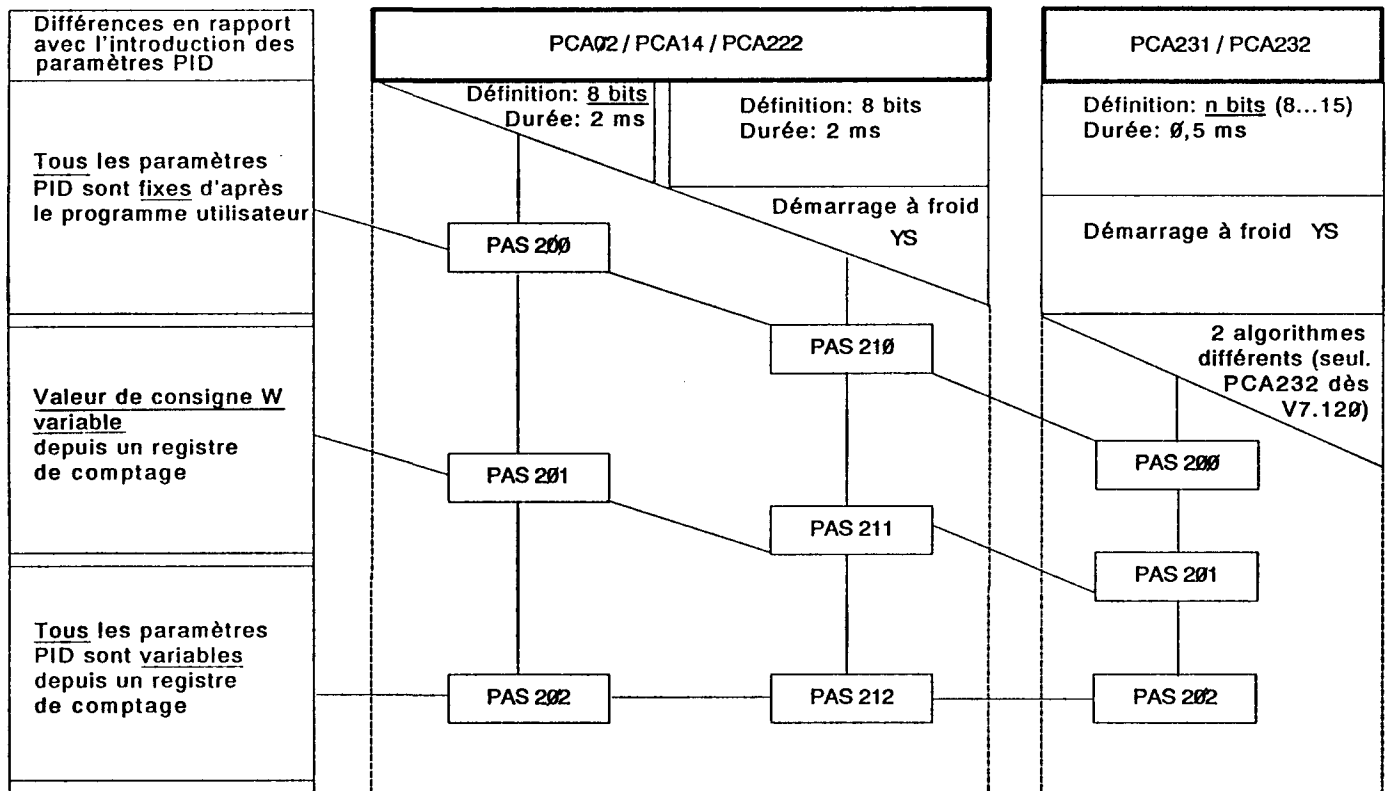
L'instruction PAS 200...212 n'est pas une instruction d'assignation, mais comme indiqué dans l'exemple suivant, elle doit être traitée périodiquement.

Exemple:



Description des instructions paramétrables PAS 200...212

Les instructions PAS 200...212 se différencient les unes des autres par les processeurs. Comme vous le voyez dans la table suivante il n'y a que des légères différences entre les CPU à 8 bits (PCA02, PCA14 et PCA222) et les CPU à 16 bits (PCA231 et PCA232). Le nombre de paramètres variables varie aussi légèrement dans chaque groupe d'instruction PAS. Ainsi le groupe supérieur PAS 200, 210 n'a aucun paramètre variable tandis que le groupe inférieur PAS 202, 212 a la possibilité d'adapter tous les paramètres, par lecture dans les registres de comptage. Ceci permet à l'utilisateur de stocker les paramètres de réglage en mémoire ou de les faire varier aisément par des roues codeuses BCD.



PAS 200, 201, 202

Pour CPU PCA02 / PCA14 / PCA222
(Algorithme rapide pour régulation
avec module analogique à 8 bits)

Description détaillée des paramètres pour PAS 200 et 201

Aperçu			Commentaire	Domaine de nombre
1	PAS(29)	200	; Régulation PID	200 ou 201
2	00	DBLK	; No du bloc de données	No 31...0
3	00	W	; Valeur de consigne	Valeur 0...255*
4	00	X, Y	; Adresse du compteur pour valeur réglée/de réglage	ADR 256...319
5	XX	F _I	; Facteur d'intégration (T_0/T_I)	Valeur 0...32767
6	XX	F _D	; Facteur différentiel (T_D/T_0)	Valeur 0...32767
7	XX	F _P	; Facteur proportionnel ($1/X_P$)	Valeur 0...32767
8	00	DR	; Zone morte	Valeur 0...255
9	00	0	} Zéro	
10	00	0		

*) pour PAS 201: Adr. 256...319

- 1ère ligne

PAS 200: La valeur de consigne W de la ligne 3 est fixe.

PAS 201: La valeur consigne W est lue dans le compteur adressé à la ligne 3.

- 2ème ligne Bloc de données

La capacité maximum est de 32 boucles de régulation. De façon à utiliser des registres internes différents pour les valeurs X_0 , Y_0 et S_0 il faut définir pour chaque boucle un numéro de bloc de données 31...0.

Afin de ne pas atteindre les registres de comptage (jusqu'à C479) il est recommandé de débiter à l'adresse DBLK31 (voir aussi "organisation des registres" page 4G).

- 3ème ligne Valeur de consigne W

. Pour PAS 200, la valeur à 8 bits (0...255) est à introduire.

. Pour PAS 201, introduire l'adresse du compteur (256...319) dans lequel la valeur (0...255) sera déposée. Le PAS 201 sera utilisé dans le cas où la valeur de consigne peut être modifiée.

- 4ème ligne Compteur auxiliaire pour X resp. Y

Adresse du compteur dans lequel la valeur réglée X_1 est déposée avant l'appel du PAS 200 (201) et la valeur de réglage Y_1 est lue après le travail de celui-ci. Dans l'exemple c'est le compteur 260 qui est utilisé. L'adresse de ce compteur doit être donnée à la ligne 4.

- 5...7ème lignes F_I , F_D , F_P

Dans la pratique ces valeurs sont généralement comprises entre 0,1 et 7,99. Pour obtenir une précision suffisante le PCA14 travaille avec le multiple 256 de ces valeurs, c'est à dire:

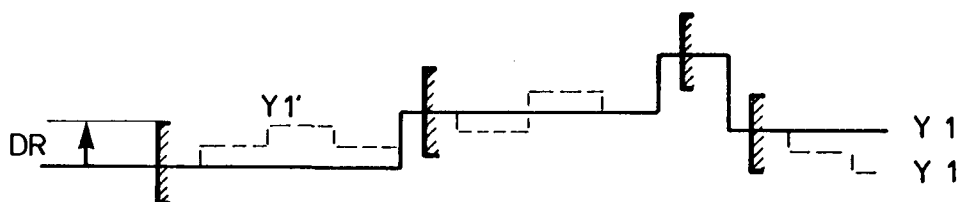
Valeur pratique	Valeur à introduire dans l'opérand
0.1	25
1.0	256
7.99	2045
24.25	6208

C ---> 0 3 / 0064

Une valeur d'opérand supérieure à 2047 peut aussi être introduite directement. Le PLC stocke le modulo dans l'opérand et le multiplicateur de 2048 dans le code. Pour convertir la valeur en nombre réel appuyez sur la touche C (convert).

- 8ème ligne Zone morte DR (Dead Range)

Avec la définition d'une zone morte, les variations de la valeur réglée plus petite que la tolérance (zone morte) ne sont pas prises en compte (valeur de réglage Y1 non modifiée).



Lorsque la valeur $Y1'$ reste dans la zone morte DR, aucune modification de la valeur de réglage $Y1$.

La valeur de la zone morte peut être définie de 0...255 et est considérée comme une valeur de consigne. Elle influence les variations positives et négatives.

Remarque:

Les instructions PAS 200, 201 et 202 sont indépendantes de l'état de l'ACCU et sont donc toujours travaillées. Le temps nécessaire pour ce travail est d'environ 2 ms.

Description des paramètres pour PAS 202

Il est parfois nécessaire d'avoir un accès facile à tous les paramètres afin de pouvoir modifier leurs valeurs selon les besoins durant le travail. Pour cela il faut utiliser le PAS 202.

Aperçu:

1	PAS(29)	202	:	Régulation PID
2	00	DBLK	:	No du bloc de données
3	00	FIC	:	1ère adresse d'un groupe de 6 compteurs consécutifs où seront mémorisés W, X/Y, F _I , F _D , F _P , DR
4	00	0	}	Zéro
5	00	0		
6	00	0		
7	00	0		
8	00	0		
9	00	0		
10	00	0		

- 3ème ligne FIC (First Counter)

Avec par exemple, l'adresse 260 sur cette ligne, la répartition sera la suivante:

		Contenu du registre C
C 260	Valeur de consigne W	0...255
C 261	Valeur réglée X1 valeur de réglage Y1	0...255
C 262	F _I facteur d'intégration	0...32767
C 263	F _D facteur différentiel	0...32767
C 264	F _P facteur proportionnel	0...32767
C 265	DR zone morte	0...255

Les valeurs que les compteurs doivent contenir ont la même signification que les valeurs positionnées directement dans le PAS 200 et 201.

*) Pour obtenir ces valeurs qui seront lues dans les compteurs, il faut multiplier les valeurs pratiques de F_I, F_D et F_P par 256.

Partie L
Exemple 6



PAS 210, 211, 212

Pour les séries PCA02 / PCA14 / PCA222

PAS 200, 201, 202

Pour les séries PCA231 / PCA232Résolution réglable de 8...15 bits

Pour des raisons historiques dues à l'évolution et l'amélioration constante de nos produits, nous possédons maintenant différentes instructions ayant la même fonction. Les processeurs développés après le PCA14, c'est-à-dire le PCA2.M32 et M31 ont eu dès leurs apparitions la possibilité de régler la résolution. Avec l'introduction des modules analogiques à 12 bits pour le PCA1 il a fallu, alors utiliser une nouvelle instruction PAS. Pour continuer d'assurer la compatibilité des anciens programmes une instruction PAS différente a dû être trouvée c'est à dire le PAS 210, 211 et 212.

Par la suite, nous ne mentionerons que les instructions PAS 210, 211 et 212 (pour les CPU PCA02 / PCA14 / PCA222). Mais, par analogie les autres instructions PAS 200, 201 et 202 peuvent être utilisées de façon similaire avec les séries PCA231 / PCA232 (voir table de la page 31H).

Il n'y a que 2 points différents:

- Temps de réaction pour PAS 200...212

PCA02 / 14 / 222	: environ 8 ms
PCA231 / 232	: environ 0,5 ms

Malgré le long temps de réaction de 8 ms, la vitesse de commutation (9600 baud) n'est pas affectée car la sortie sérielle est activée chaque 1ms.

- Le PCA232 a dès la Firmware version V7.120 un autre algorithme à disposition où la valeur de consigne est prise en compte beaucoup plus rapidement (voir ligne 9 de l'instruction PAS).

Description détaillée des paramètres pour PAS 210 et 211
pour PCA02 / PCA14 / PCA222 (resp. 200 ou 201 pour PCA231 / PCA232)

Aperçu			Commentaire	Domaine de valeur
1	PAS(29)	210	; Régulation PID	210 ou 211
2	n	DBLK	; No du bloc de données et n bits de resolution	31... 0
3	00	W	1); Valeur de consigne	Valeur 0... 4095 2)3)
4	00	X, Y	; Adresse du compteur pour valeur réglée/de réglage	ADR 256... 319 2)
5	XX	F _I	; Facteur d'intégration (T ₀ /T _I)	Valeur 0...32767
6	XX	F _D	; Facteur différentiel (T _D /T ₀)	Valeur 0...32767
7	XX	F _P	; Facteur proportionnel (1/X _P)	Valeur 0...32767
8	00	DR	; Zone morte	Valeur 0... 4095 2)
9	0X	YS	; Adresse du compteur pour la valeur initiale YS	ADR 256... 319 2)
10	00	0	; 0	

- 1ère ligne

PAS 210: La valeur de consigne W de la ligne 3 est fixe.

PAS 211: La valeur de consigne W est lue dans le compteur adressé à la 3ème ligne (C256...C319).

- 2ème ligne Résolution à n bit et bloc de données

Dans le code, le nombre de bits n à traiter pour les valeurs W, X, Y, YS et DR est indiqué avec:

n = 08 4) $\hat{=}$ 8 bits ---> 0... 255
 12 5) $\hat{=}$ 12 bits ---> 0... 4095
 15 $\hat{=}$ 15 bits ---> 0...32767

La capacité maximum est de 32 boucles de régulation. De façon à utiliser des registres internes différents pour les valeurs X₀, Y₀ et S₀, il faut définir pour chaque boucle un numéro de bloc de données (0...31) dans l'opérand. Afin de ne pas atteindre les registres de comptage (jusqu'à C479), il est recommandé de débiter à l'adresse DBLK31 (voir aussi "Organisation des registres" à la page 4G).

- 3ème ligne Valeur de consigne W

- . Pour PAS 210, la valeur à 12 bits (0...4095)³⁾ est à introduire directement.
- . Pour PAS 211, introduire l'adresse du compteur (256...319) dans lequel la valeur (0...4095)³⁾ sera déposée. Le PAS 211 sera utilisé dans le cas où la valeur de consigne peut être modifiée.

- 4ème ligne Compteur auxiliaire pour X resp. Y

Adresse du compteur dans lequel la valeur réglée X₁ est déposée avant l'appel du PAS 210 (211) et la valeur de réglage Y₁ est lue après le travail de celui-ci. Dans l'exemple c'est le compteur 260 qui est utilisé.

1) Pour PAS 211: Adr. de compteur 256...319

2) La valeur dans l'opérand ou dans le compteur est dépendante de la 2ème ligne.

3) Valable pour n = 12 (ou 00) dans la 2ème ligne.

4) Pour le PCA14, la valeur n = 8 peut être aussi n = 0 pour 8 bits.

5) Pour le PCA232, la valeur n = 12 peut être aussi n = 0 pour 12 bits.

- 5...7ème ligne F_I , F_D , F_P

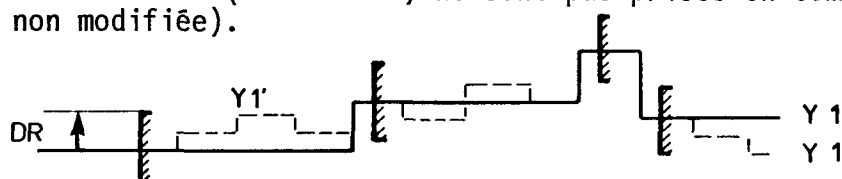
Dans la pratique ces valeurs sont généralement comprises entre 0,1 et 7,99. Pour obtenir une précision suffisante, on travaille avec le multiple 256 de ces valeurs.

Valeur pratique	Valeur à introduire dans l'opérand
0,1	25
1,0	256
7,99	2045
24,25	6208 [C] ---> 0 3 / 0064

Une valeur d'opérand supérieure à 2047 peut aussi être introduite directement. Le PLC stocke le modulo dans l'opérand et le multiplicateur de 2048 dans le code. Pour convertir la valeur en nombre réel appuyez sur la touche [C] (convert) de l'appareil de programmation P10/P05.

- 8ème ligne Zone morte DR (Dead Range)

Avec la définition d'une zone morte, les variations de la valeur réglée plus petite que la tolérance (Zone morte) ne sont pas prises en compte (valeur de réglage Y_1 non modifiée).



Lorsque la valeur calculée $Y1'$ reste dans la zone morte DR, il n'y a aucune modification de la valeur de réglage Y_1 .

La valeur de la zone morte peut être définie de 0 et 4095, et est considérée comme valeur absolue. Elle influence les variations positives et négatives.

- 9ème ligne Valeur d'initialisation Y_S (valeur de départ)

Dans le cas où les blocs de données 0...31 sont définis non-volatils, lors de la première utilisation (ou pour une autre boucle de régulation), il faut tout d'abord positionner une valeur de départ.

Dans la 9ème ligne dans l'opérand le compteur adressé CYS doit contenir la valeur Y_S . Lors de l'initialisation, les valeurs suivantes sont utilisées (et déposées dans le bloc de données):

$$\text{Bloc de données: } X_0 = X; Y_0 = Y_S; S_0 = \frac{Y_S/F_P - (W - X)}{F_I}$$

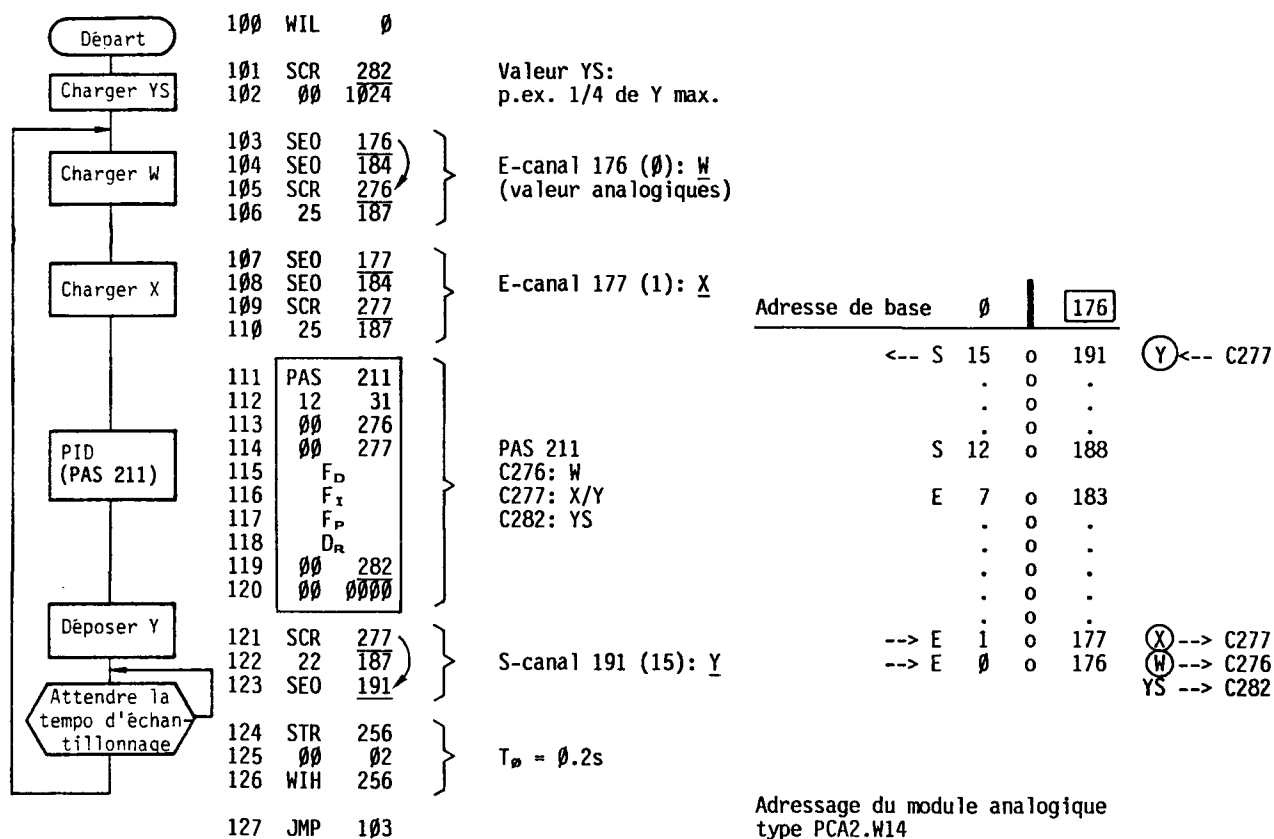
$$\text{Valeur de réglage: } Y = Y_S$$

Après la première exécution du PAS, le compteur CYS est automatiquement mis à 0 et n'est plus consulté lors des exécutions suivantes. A part l'initialisation, il est également possible d'utiliser le compteur CYS pour le passage sans accoup de manuel à automatique ou lors de modification de paramètres (voir exemple de programme 7).

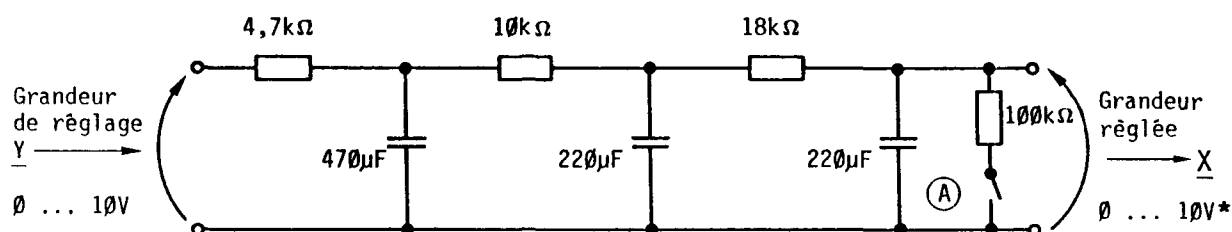
Dans le code de la ligne 9, la valeur 00 sélectionne l'algorithme normal pour le calcul de la valeur de consigne Y . La valeur 01 permet alors le choix de l'autre algorithme avec "saut de valeur de consigne" (pour PCA232 dès la V7.120; voir aussi 29H).

Etant donné que l'emploi des modules analogiques de la gamme PCA2 est plus aisé que ceux de la gamme PCA1, nous utiliserons ici, le CPU PCA2.M22 et le module analogique PCA2.W14 avec l'adresse de base 176.

Exemple où le mode manuel est inclu:



Exemple de régulation: Avec le programme ci-dessus, il est possible d'effectuer la régulation PID sur le circuit de simulation ci-après:

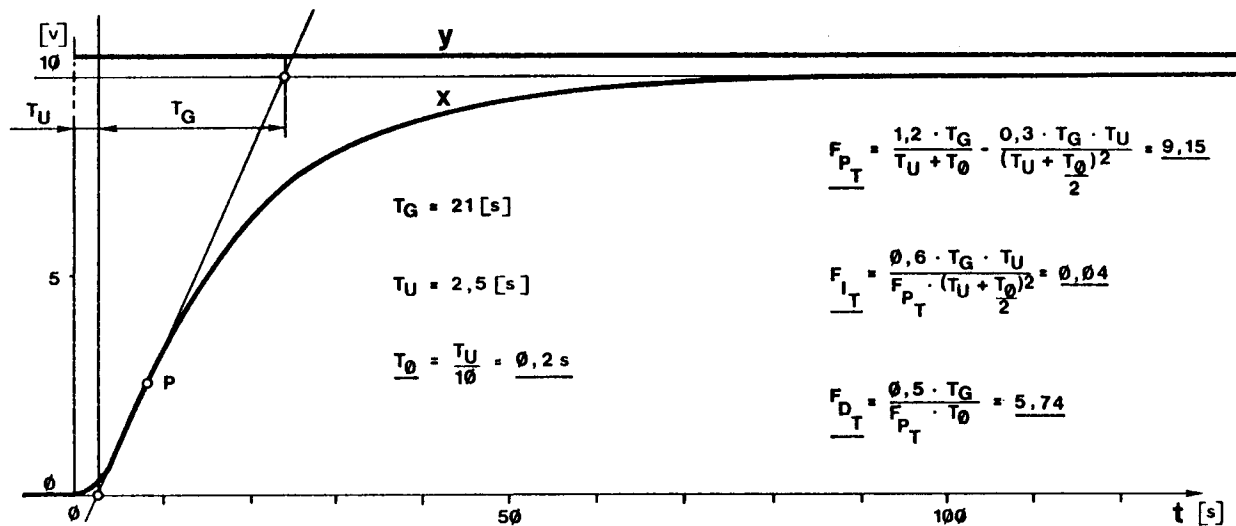


A la fermeture du commutateur (A), une perturbation d'environ 20% est provoquée.

A l'aide de la courbe réponse (pour Y variant 0 ---> 10V), les paramètres F_P, F_I, F_D et T₀ peuvent, selon la formule de Takahashi, être calculés.

*) L'instrument de mesure pour X doit avoir une impédance d'entrée de 1MΩ au minimum.

Courbe de réponse d'une boucle de réglage



Pour l'introduction dans le PAS 211, les valeurs calculées ci-dessus doivent encore être multipliées par 256:

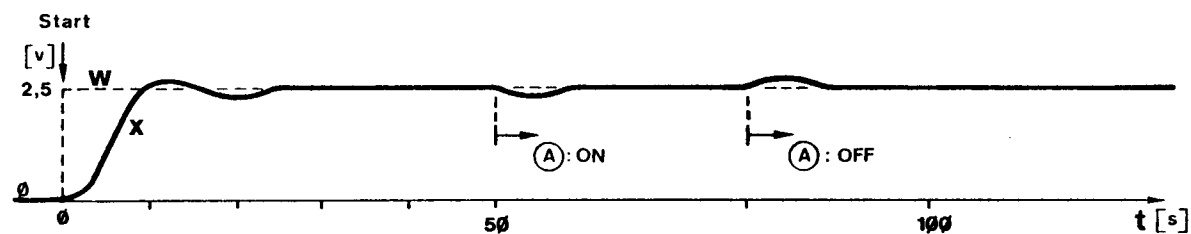
$$F_I = F_I \times 256 = 10$$

$$F_D = F_D \times 256 = 1469$$

$$F_P = F_P \times 256 = 2341$$

PAS	211	
12	31	; 12 bits, bloc de données 31
00	276	; Compteur pour W
00	277	; Compteur pour X/Y
00	10	; F_I
00	1469	; F_D
01	293	; F_P (\boxed{C} = E E 2341)
00	00	; DR
00	282	; Compteur pour YS
00	00	; --

Comportement du circuit avec les paramètres calculés ci-dessous et une valeur de consigne $W = 1024$ (2,5V):



(A) : Grandeur de la perturbation simulée ($\approx 20\%$).

Description de l'instruction PAS 212

Pour la mise en service d'une boucle de réglage, il est avantageux de pouvoir accéder à tous les paramètres et de pouvoir les modifier à volonté pendant le travail. Le PAS 212 répond à ce besoin (PAS 202 pour PCA231/232).

Aperçu:

1	PAS(29)	212	; Régulation PID
2	n	DBLK	; No du bloc de données et nombre de bits n
3	00	FIC	; 1ère adresse d'un groupe de 7 compteurs consécutifs pour les valeurs W, X/Y, F _I , F _D , F _P , DR, YS
4	00	0	} Zéro
5	00	0	
6	00	0	
7	00	0	
8	00	0	
9	00	0	
10	00	0	

- 2ème ligne voir PAS 210
- 3ème ligne FIC (First Counter)

Avec, par exemple, l'adresse 276 sur cette ligne, la répartition sera la suivante:

Chiffres de C		
C 276	Valeur de consigne W	0... 4095 ²⁾
C 277	Compteur auxiliaire pour X1 ou Y1	0... 4095 ²⁾
C 278	F _I facteur d'intégration	0...32767
C 279	F _D facteur différentiel	0...32767
C 280	F _P facteur proportionnel	0...32767
C 281	DR Zone morte	0... 4095 ²⁾
C 282	YS Valeur d'initialisation	0... 4095 ²⁾

Les valeurs que les compteurs doivent contenir ont la même signification que les valeurs positionnées directement dans les PAS 210 et 211.

Important:

Toutes les instructions PAS 200...212 sont indépendantes de l'état de l'ACCU et sont donc toujours exécutées.

¹⁾ Pour obtenir ces valeurs qui seront lues dans les compteurs, il faut multiplier les valeurs pratiques de F_I, F_D et F_P par 256.

²⁾ Valeur pour le module analogique à 12 bits.

PAS 250/251 Registre à décalage, à rotation et FIFO (First In First Out)
PAS 250 Registre à décalage et à rotation

L'instruction **PAS 250** à 10 lignes permet de définir et de résoudre le décalage et la rotation d'informations mémorisées sur indicateurs et sorties. En effet, les registres de sorties lisibles sont aussi utilisables. L'instruction PAS est formée comme suit:

1	PAS	(29)	250	:	Activation de la fonction paramétrique	
2		00	SADD	:	Adresse de début du registre SADD	1)
3		00	DADD	:	Adresse de fin du registre DADD	1)
4		00	NNN	:	Nombre de bits d'une information (N bits)	2)
5		00	0	}	Zéro	
6		00	0			
7		00	0			
8		00	0			
9		00	0			
10		00	0			

Selon où se trouve l'adresse la plus basse soit dans SADD ou DADD, la rotation ou le décalage se fera en avant ou en arrière. L'instruction paramétrisable PAS 250 sera effectuée quelque soit le contenu de l'ACCU et ne l'influence pas.

1) Les 2 adresses de début SADD (source address) et de fin DADD (destination address) sont indexables.

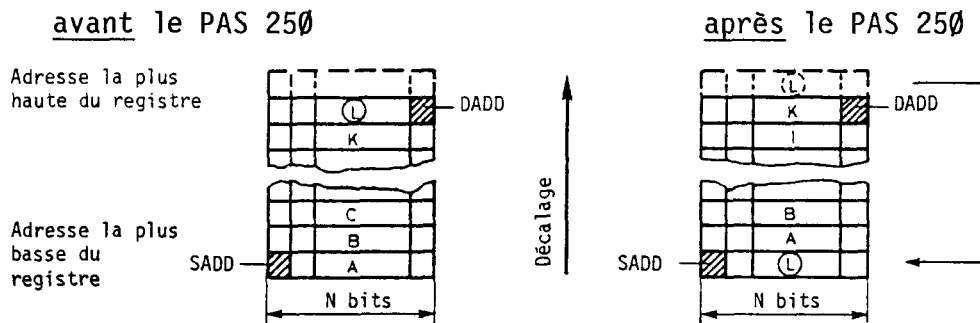
2) La taille d'une information peut être comprise entre 1...255 bits.

NNN = 1...255: Valeure fixe

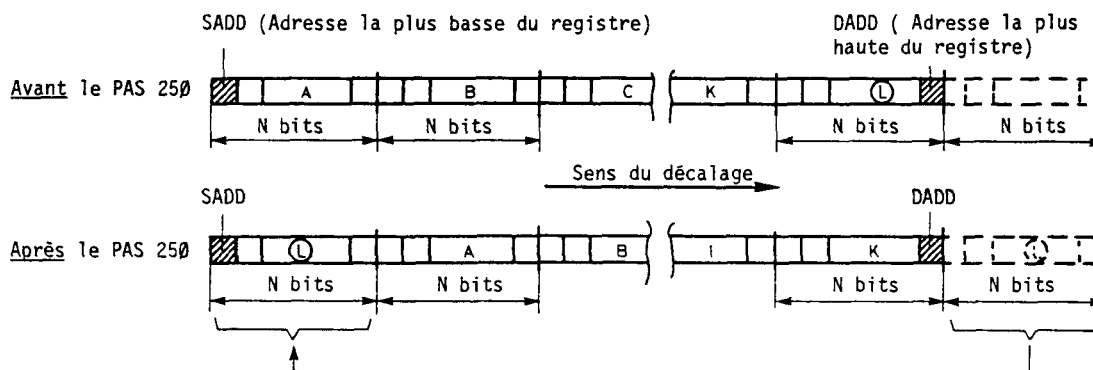
NNN = 256...511: L'information N bits se trouve dans le compteur correspondant.

Fonctionnement

a) $SADD < DADD \hat{=}$ décaler dans le sens avant

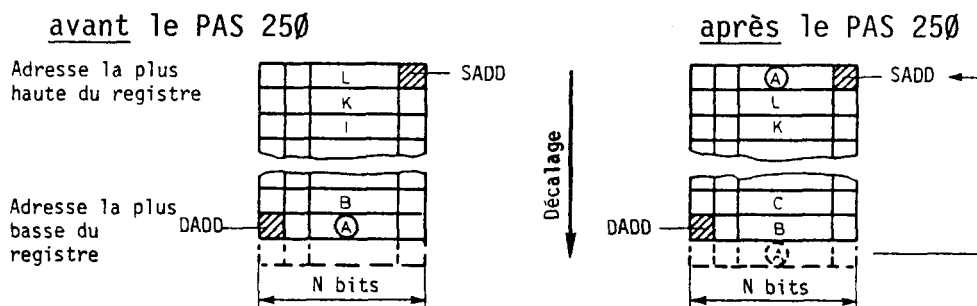


Le même registre représenté différemment:



Ces représentations montrent qu'il faut à partir de l'adresse DADD réserver N bits. Ces bits correspondent à un pas du registre et sont utilisés comme mémoire-tampon.

b) $SADD > DADD \hat{=}$ décaler dans le sens arrière



La représentation montre qu'il faut avant l'adresse DADD réserver N bits. Ces bits correspondent à un pas du registre et sont utilisés comme mémoire-tampon.

L'instruction PAS 250 correspond à un registre rotatif (voir les figures ci-dessous). Lorsqu'il faut faire un registre à décalage linéaire, par exemple dans le cas a) (décaler dans le sens avant), l'information finale se trouve dans le champ DADD. Après décalage, il faut introduire la nouvelle information dans le champ SADD.

Partie L
Exemples 8 et 9



PAS 251 Introduction dans le registre FIFO (First in - First out)

Avec l'instruction à 10 lignes PAS 251, une information de N bits peut être introduite et décalée jusqu'à la dernière place libre du registre FIFO.

La taille du registre FIFO est définie par son adresse de début SADD et son adresse de fin DADD. L'instruction est constituée comme suit:

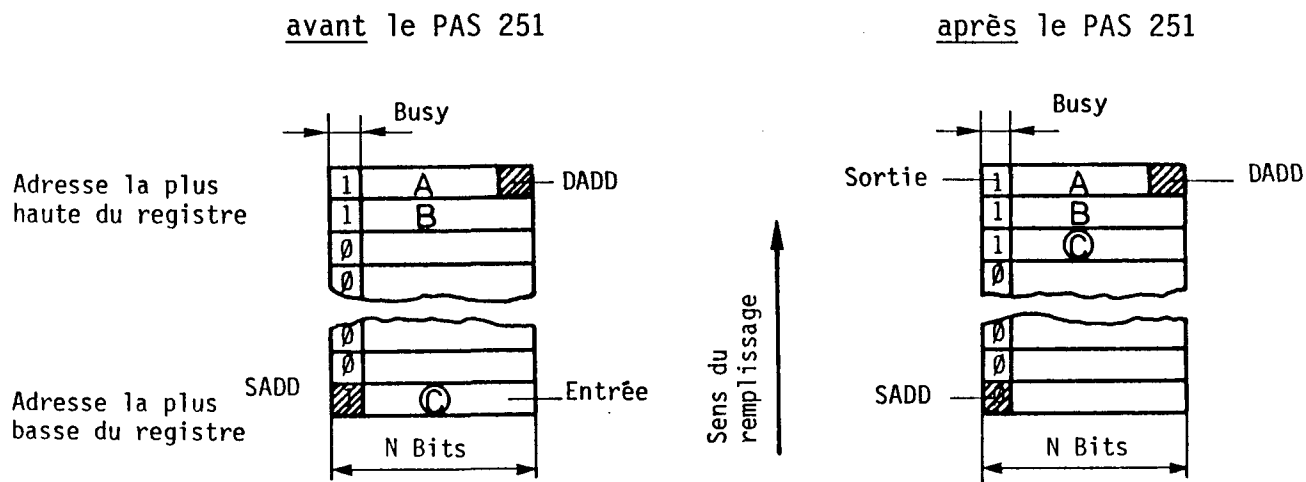
1	PAS	(29)	251	;	Activation de la fonction paramétrique	
2		00	SADD	;	Adresse de début du registre SADD	1)
3		00	DADD	;	Adresse de fin du registre DADD	1)
4		00	NNN	;	Nombre de bits d'une information (N bits)	2)
5		00	0	}	Zéro	
6		00	0			
7		00	0			
8		00	0			
9		00	0			
10		00	0			

SADD doit toujours être l'adresse la plus basse du registre et DADD la plus haute. L'instruction paramétrisable PAS 251 sera effectuée quelque soit le contenu de l'ACCU et l'état de l'ACCU est influencé en rapport du résultat de l'opération.

1) Les 2 adresses de début SADD (source address) et de fin DADD (destination address) sont indexables.

2) La taille d'une information peut être comprise entre 1...255 bits.

Fonctionnement

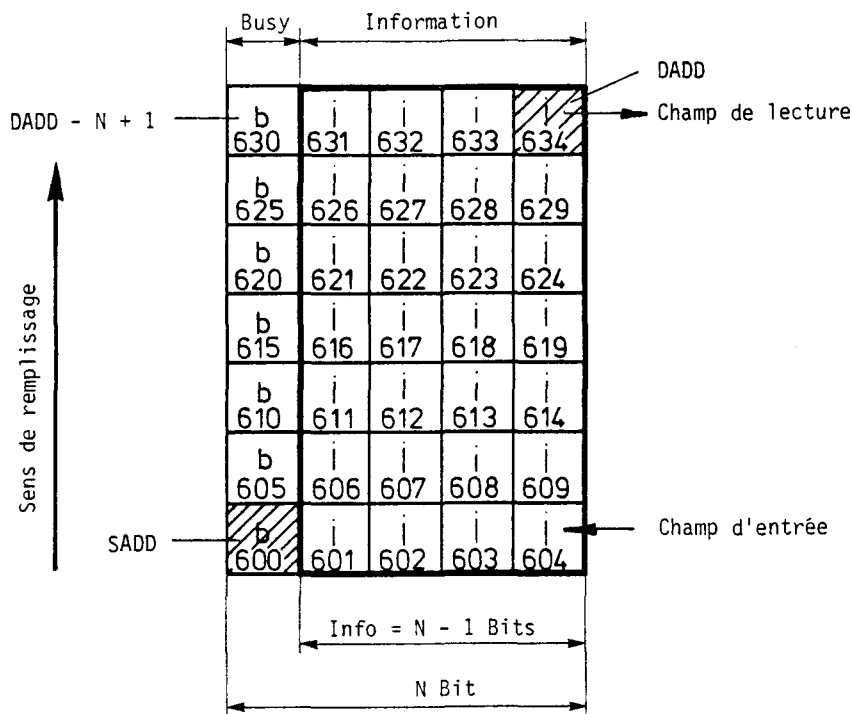


La figure ci-dessus représente la manière d'introduire une valeur de N bits dans la dernière place libre se trouvant dans le sens SADD ---> DADD.

Pour gérer avec sécurité cette instruction, il convient avant d'introduire chaque nouvelle valeur, de s'assurer que le registre n'est pas plein. Sans cette précaution, la nouvelle information détruirait la dernière information en la remplaçant. Pour pouvoir contrôler facilement l'état de remplissage du registre, il faut au début de chaque plage d'informations un bit réservé au "busy". Lorsque ce bit est à "H", la place correspondante est occupée.

Il suffit simplement de contrôler le bit "busy" de la place d'entrée avant d'introduire une nouvelle valeur dans le registre puis de prendre la décision correcte selon le résultat de ce test.

Le registre FIFO se présente comme suit:



$SADD = 600$
 $DADD = 634$
 $N = 5 \text{ bits}$
 $\text{Information} = 4 \text{ bits}$

Marche à suivre:

- a) Si la dernière valeur ne doit pas être effacée, il faut d'abord contrôler que $SADD$ (Busy-Bit) est "H".

$SADD = H$: FIFO plein
 $SADD = L$: place libre dans le FIFO

- b) L'utilisateur entre une information de largeur N bits-1 et place le bit $SADD$ "busy" sur "H".

- c) Exécuter le PAS 251.
Après le PAS 251, le bit $SADD$ est positionné sur "L" lorsque le registre FIFO n'est pas plein. S'il reste "H", la dernière place est occupée.

- d) Traitement des résultats donnés par le PAS 251

$ACCU = 1$: Exécution sans erreur détectée

$ACCU = \emptyset$: Erreur de dimensionnement du FIFO, le PAS 251 ne sera pas exécuté.

$SADD = H$: FIFO plein. L'introduction d'une nouvelle valeur provoque le remplacement de la valeur précédente.

$SADD = L$: FIFO libre, place libre pour d'autre(s) information(s).

$DADD - (N-1) = L$: Le FIFO est vide.

Notes:

PARTIE I INSTRUCTIONS DU PROCESSEUR DE MOTS NIVEAU 3
(seulement PCA232 ou 231)

- I 1 Changement du processeur de bits au processeur de mots,
 Jeu d'instruction du processeur de mots**
- I 2 Instructions arithmétiques**
- I 3 Instructions pour le traitement des registres A**
- I 4 Vue d'ensemble des instructions de transfert**
- I 5 Instructions de comptage et de saut**
- I 6 Diverses instructions**

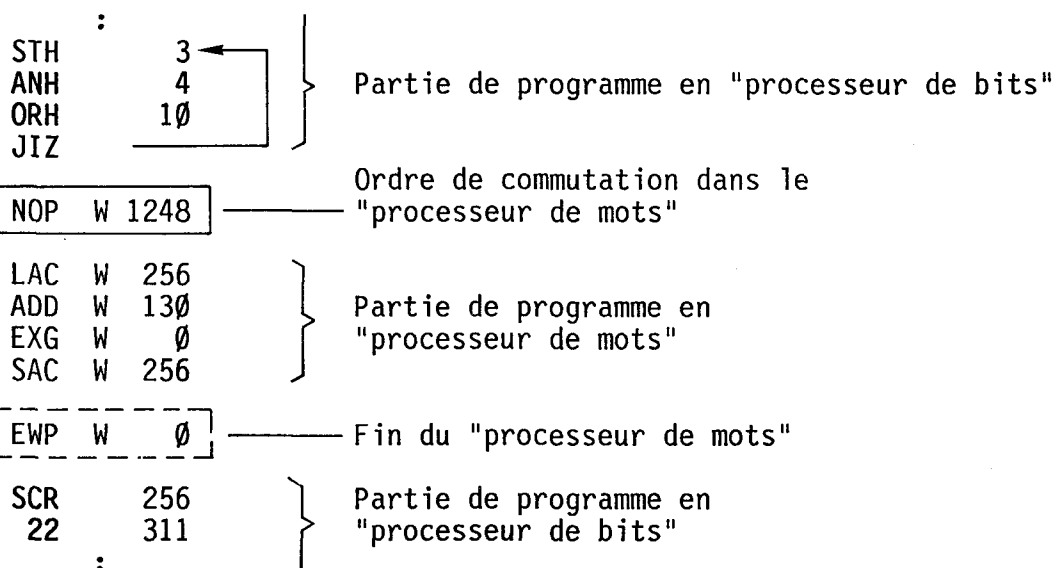
PARTIE I INSTRUCTIONS DU PROCESSEUR DE MOTS NIVEAU 3 (seulement PCA232 ou PCA231)

I 1 Changement du processeur de bits au processeur de mots, jeu d'instruction

NOP 1248 Changement du processeur de bits au processeur de mots

Les 5 bits disponibles pour le code des instructions ne permettent de définir que 32 instructions. L'ordre de commutation NOP 1248 donne accès au jeu des 32 autres instructions du processeur de mots.

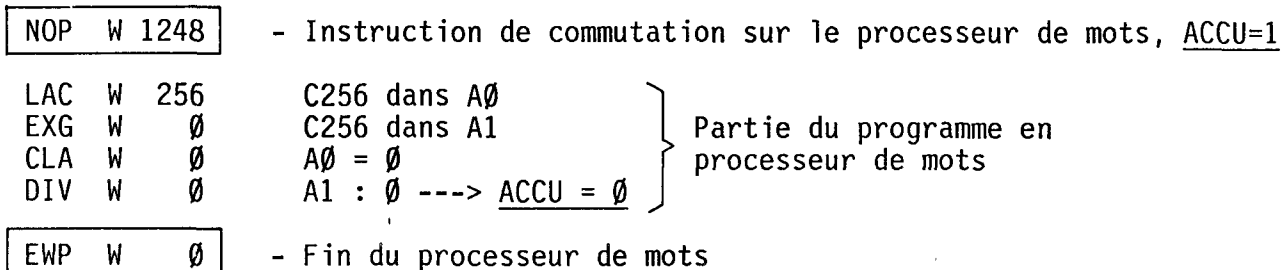
Exemple:



NOP 1248 est toujours exécuté, quel que soit l'état de l'ACCU.

NOP 1248 positionne l'ACCU = 1

Certaines instructions du processeur de mots positionnent l'ACCU = 0, si p.ex. des limites de capacité sont dépassées ou en cas de division par zéro. Par un contrôle de l'état de l'ACCU après EWP, il est possible de déterminer si les limites admissibles sont respectées.



JIZ Saut dans la routine de signalisation d'erreur

↓

Note: De NOP 1248 à JIZ, il n'y a pas de changement de PP. En cas de PP critiques du point de vue temps, il faut en tenir compte en subdivisant les programmes de processeur de mots en sections plus courtes.

Jeu d'instruction du processeur de mots

Code num.	Code mnémo.	Opérand	Instruction en anglais	Description	Format des données	
00	NOP	1248	---	Changement du processeur de bits au processeur de mots. Mis ACCU = 1		

Instructions de transfert de données

01	RRG	Rn	Read Register	Entrée du mot Rn et mémorisation dans R4	8 bits	i
02	WRG	Rn	Write into Register	Ecrire en Rn le mot R4	8 bits	i
03	RRE	Rn	Read Register and write in Elements	Déposer le contenu du mot Rn sur le 8 élément adressés par A1	8 bits	i
04	WRE	Rn	Write Elements into Register	Ecrire dans Rn le contenu des 8 éléments En...En-7 adressés par A1	8 bits	i
05	LAR	Rn	Load A0 with Registers	Chargement de A0 avec le bloc de registre Rn	5x8 bits	i
06	SAR	Rn	Store A0 in Registers	Mémoriser A0 dans le bloc de registre Rn	5x8 bits	i
07	LAC	Cn	Load A0 with Counter	Chargement de A0 avec le compteur Cn	BCD	i
08	SAC	Cn	Store A0 in Counter	Mémoriser A0 dans le compteur Cn	BCD	i
09	WEL	En	Write into { Lower } digit	Ecrire dans les élé- { R4 (10 ⁰)	4 bits	i
10	WEU	En	elements { Upper }	ments En...En-3 le digit { R4 (10 ¹)		

Instructions de comptage et de saut

11	INR	Rn	Increment Register	Augment. } 1 à la valeur BCD { > 99		i
12	DER	Rn	Decrement Register	Dimin. } de Rn et "carry" { < 0		i
13	SNC	0 En	Skip if no "carry" Skip if En = 0	Sauter l'instruction { "carry" = 0 suivante si En = 0		i
24	SEW	0 En	Skip to EWP if no "carry" Skip to EWP if En = 0	Sauter toutes les in- { "carry" = 0 structions jusqu'à EWP En = 0		i

Traitement des registres arithmétiques

14	CLA	Ø, 1, 2	Clear A	Effacement des registres AØ, A1 ou A2		
15	LAI	K	Load A immediately	si K ≤ 99 ----> chargement de R4 si K = 100...2047 ----> chargement de AØ		
16	DBN	Ø	Decimal to Binary	Convertir AØ de décimal en binaire		
17	BND	Ø	Binary to Decimal	Convertir AØ de binaire en décimal		
18	ROR	Ø, 1	Rotate Register	Rotation R4...RØ ou RØ...R4		
19	ROA	Ø, 1	Rotate A	Rotation AØ...A2 ou A2...AØ		
20	EXG	Ø	Exchange A1 with AØ	Echange A1 et AØ		
		Rn	Exchange A1 with Rn...Rn-4	Echange A1 et Rn...Rn-4		i

Instructions arithmétiques

25	CMP	En	Compare AØ with A1	Comparaison de AØ et A1 A1 > AØ ----> En = 1 A1 = AØ ----> En-1 = 1 A1 < AØ ----> En-2 = 1	BCD	i
26	SQR	Ø	Square Root from A1	$\sqrt{A1}$ ----> A1, chiffres entiers seulement	BCD	
27	ADD	Ø	Add AØ to A1	A1 + AØ ----> A1, "carry"	BCD	
		K	Add K to A1	A1 + K ----> A1, "carry" (K=1...2047)	BCD	
28	SUB	Ø	Subtract AØ from A1	A1 - AØ ----> A1, "carry"	BCD	
		K	Subtract K from A1	A1 - K ----> A1, "carry" (K=1...2047)	BCD	
29	MUL	Ø	Multiply A1 by AØ	A1 . AØ ----> A1, "carry"	BCD	
		K	Multiply A1 by K	A1 . K ----> A1, "carry" (K=1...2047)	BCD	
30	DIV	Ø	Divide A1 by AØ	A1 : AØ ----> A1, reste dans AØ "carry"	BCD	
		K	Divide A1 by K	A1 : K ----> A1, reste dans AØ "carry" (K=1...2047)	BCD	

Diverses instructions

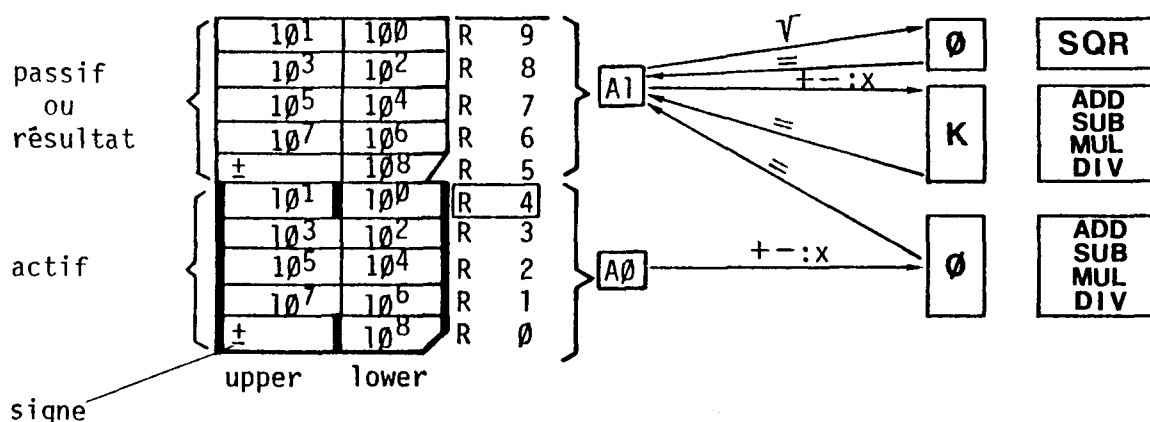
ØØ	NOP	Ø	No operation	Aucune opération		
21	CLK	En	Clock source	Assignment d'une source d'impulsion de temps		
22	SHI	Rn	Shift registers	Décalage de mots vers le haut, par pas de 1 entre R20 et Rn		i
23	TXT	Txn	Text	Sortie de texte		i
31	EWP	Ø	End word processor	Fin du travail dans le processeur de mots		

(i) = indexable

1 2 Instructions arithmétiques

Les opérations arithmétiques seront exécutées exclusivement entre les registres A1 et A0, respectivement une constante K. A1 reçoit l'opérateur passif et conserve le résultat de l'opération, alors que A0 reste inchangé (excepté dans la division).

Operand



Pour des opérations arithmétiques, les valeurs doivent être disponibles en format BCD. Des valeurs BCD positives et négatives entières sont traitées jusqu'à une capacité de $\pm 10^9 - 1$ ($\pm 999'999'999$). Ce faisant, le signe est placé avant la décimale 10⁸ avec la signification suivante:

0 = positive +
9 = negative -

Lorsqu'on lit un bloc de registre, tous les bits dans le champ du signe qui ne sont pas zéro, sont considérés comme négatifs.

Pour le transfert de données il est possible d'utiliser une combinaison quelconque de 10x4 bits.

ADD Addition

ADD: Add A0 (ou K) à A1

Code mnémotechnique	Code chiffré	Opérand	Opération
ADD	27	0	A1 + A0 ----> A1
		K = 1...2047	A1 + K ----> A1

Capacité de calcul: $\pm 999'999'999$ ou $\pm (10^9 - 1)$

En cas d'"overflow", "carry" = 1 et ACCU = 0. A1 contient exclusivement le reste.

SUB SoustractionSUB: Soustraire A0 (ou K) de A1 (Subtract A0 or K from A1)

Code mnémonique	Code chiffré	Opérand	Opération
SUB	28	0	$A1 - A0 \rightarrow A1$
		$K = 1 \dots 2047$	$A1 - K \rightarrow A1$

Capacité de calcul: $\pm 999'999'999$ ou $\pm(10^9-1)$

En cas d'"overflow", "carry" = 1 et ACCU = 0. A1 contient exclusivement le reste.

MUL MultiplicationMUL: Multiplier A1 par A0 (ou K) (Multiply A1 by A0 or by K)

Code mnémonique	Code chiffré	Operand	Opération
MUL	29	0	$A1 \times A0 \rightarrow A1$
		$K = 1 \dots 2047$	$A1 \times K \rightarrow A1$

Capacité de calcul: $\pm 999'999'999$ ou $\pm(10^9-1)$ En cas d'"overflow", "carry" = 1 et ACCU = 0. A1 contient la valeur max. $\pm 999'999'999$.**DIV** DivisionDIV: Diviser A1 par A0 (ou K) (Divide A1 by A0 or by K)

Code mnémonique	Code chiffre	Opérand	Opération
DIV	30	0	$A1 : A0 \rightarrow A1$
		$K = 1 \dots 2047$	$A1 : K \rightarrow A1$

Le reste de la division est déposé dans A0.

"Carry" devient 1 et l'ACCU = 0 en cas de division par 0. Dans ce cas-là, l'opération n'est pas effectuée et les données ne varient pas. Limite de capacité pour le résultat: $\pm 999'999'999$.

SQR Racine carréeSQR: Racine carrée de A1 (Square Root from A1)

Code mnémonique	Code chiffré	Opérand	Opération
SQR	26	Ø	$\sqrt{A1}$ ----> A1

Limite de calcul: $\sqrt{+999'999'999}$.

Le résultat ne sort qu'en chiffres entiers (p.ex. $\sqrt{3} = 1$). Le reste est perdu. Une précision plus élevée peut être atteinte lorsque le radian est multiplié d'abord par un multiple de 100 (p.ex. $\sqrt{3'00'00} = 173$).

Lorsque l'on tire la racine carrée d'un chiffre négatif, "carry" devient 1 et ACCU = Ø. A1 contient le radian négatif non varié.

CMP Comparer les chiffres A1 et AØCMP: Comparer A1 et AØ (A1 - AØ) (compare A1 with AØ; A1 - AØ)

Code mnémonique	Code chiffré	Opérand	Opération
CMP	25	Adresse d'élément En = 2...999(i)	A1 > AØ ----> En =1, carry=Ø A1 = AØ ----> En-1=1, carry=Ø A1 < AØ ----> En-2=1, carry=1

(i) = indexable

Le résultat de l'opération de comparaison est donné directement aux adresses de 3 éléments (indicateurs ou sorties) et se trouve donc à disposition du processus. L'opération ne modifie pas les valeurs des registres AØ et A1.

La comparaison se fait en respectant le signe jusqu'à $\pm 999'999'999$ ou $\pm(10^9-1)$. Si p.ex. +3 est supérieur à -500.

I 3 Instructions pour le traitement des registres A

CLA Effacer les registres A0, A1 ou A2

CLA: Effacer A (Clear A)

Code mnémonique	Code chiffré	Opérand	Description
CLA	14	0 ou 1 ou 2	Effacer A0, A1 ou A2

LAI Charger le registre A0 avec la constante K

LAI: Charger A0 immédiatement (Load A0 immediately)

Code mnémonique	Code chiffré	Opérand	Description
LAI	15	K (0...2047)	K = 0...99 ----> seul R4 est chargé K = 100...2047 ----> l'ensemble de A0 est chargé

Si la constante vaut par ex. 57, seul R4 est chargé, R0 à R3 restant inchangés. Mais si la constante vaut par ex. 225, A0 est entièrement chargé, c'est-à-dire que les 7 chiffres disponibles précédant 225 sont considérés comme nuls.

DBN, BND Conversion décimal ----> binaire ou binaire ----> décimal

DBN, BND: Décimal ----> binaire, binaire ----> décimal (DBN: decimal to binary)
(BND: binary to decimal)

Code mnémonique	Code chiffré	Opérand	Description
DBN	16	0	Convertir A0 de décimal en binaire (limité de capacité: ±999'999'999)
BND	17	0	Convertir A0 de binaire en décimal (limité de capacité: R4 à R1)

Les opérations arithmétiques ne seront effectuées correctement qu'avec des nombres sous forme BCD. Si par ex. des valeurs analogiques se présentent aux entrées sous forme binaire pure, elles devront être mises sous forme BCD avant leur traitement. Le même cas se présente, mais inversement, pour la sortie de valeurs destinées à un traitement analogique.

ROR Rotation des registresROR: Rotation des registres (Rotate registers)

Code mnémonique	Code chiffré	Opérand	Description
ROR	18	0	Rotation R4->R3->R2->R1->R0->R4
		1	Rotation R0->R1->R2->R3->R4->R0

Le fonctionnement de ces commandes figure dans le sommaire au début du chapitre suivant.

ROA Rotation AROA: Rotation A (Rotate A)

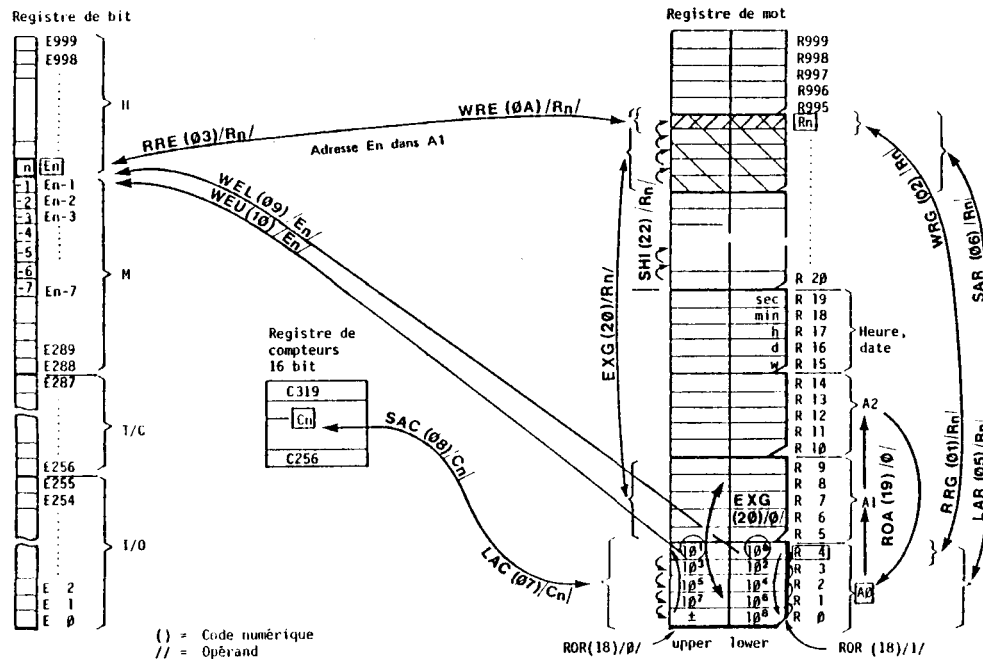
Code mnémonique	Code chiffré	Opérand	Description
ROA	19	0	Rotation A0 -> A1 -> A2 -> A0
		1	Rotation A2 -> A1 -> A0 -> A2

EXG Echanger A1 avec A0 ou avec le bloc de registre Rn-4...RnEXG: Echange A (Exchange A)

Code mnémonique	Code chiffré	Opérand	Description
EXG	20	0	Echanger A1 avec A0
		Rn de 14...999 (i)	Echanger A1 avec Rn-4...Rn R9 <---> Rn R8 <---> Rn-1 R7 <---> Rn-2 R6 <---> Rn-3 R5 <---> Rn-4

(i) = indexable

I 4 Vue d'ensemble des instructions de transfert



Code num.	Code mnémo.	Opérand	Instruction en anglais	Description	Format des données	(i)
-----------	-------------	---------	------------------------	-------------	--------------------	-----

Instructions de transfert de données

01	RRG	Rn	Read Register	Entrée du mot Rn et mémorisation dans R4	8 bits	i
02	WRG	Rn	Write into Register	Ecrire en Rn le mot R4	8 bits	i
03	RRE	Rn	Read Register and write in Elements	Déposer le contenu du mot Rn sur 8 éléments En...En-7 adressés par A1	8 bits	i
04	WRE	Rn	Write Elements into Register	Ecrire dans Rn le contenu des 8 éléments En...En-7 adressés par A1	8 bits	i
05	LAR	Rn	Load A0 with Registers	Chargement de A0 avec le bloc de registre Rn	5x8 bits	i
06	SAR	Rn	Store A0 in Registers	Mémoriser A0 dans le bloc de registre Rn	5x8 bits	i
07	LAC	Cn	Load A0 with Counter	Chargement de A0 avec le compteur Cn	BCD	i
08	SAC	Cn	Store A0 in Counter	Mémoriser A0 dans le compteur Cn	BCD	i
09	WEL	En	Write into Lower elements	Ecrire dans les éléments En...En-3 le digit	R4 (10 ⁰)	
10	WEU	En	Write into Upper elements	Ecrire dans les éléments En...En-3 le digit	R4 (10 ¹)	i

Traitement des registres arithmétiques

18	ROR	0, 1	Rotate Register	Rotation R4...R0 ou R0...R4		
19	ROA	0, 1	Rotate A	Rotation A0...A2 ou A2...A0		
20	EXG	0	Exchange A1 with A0	Echange de A0 et A1		
		Rn	Exchange A1 with Rn...Rn-4	Echange A1 et Rn-4...Rn		i

(i) = indexable

Le sommaire précédent montre clairement le fonctionnement de toutes ces instructions. Un transfert de données est possible non seulement à l'intérieur du registre de mots, mais encore vers les registres de bits et de comptage. L'on dispose de diverses instructions de longueurs 4, 8 ou 40 bits, qui concernent en général des données sous forme BCD.

Sous la forme binaire pure, les données peuvent être échangées entre tous les registres, excepté celui de comptage. Pour le traitement arithmétique, les données correspondantes doivent cependant être disponibles sous forme BCD (voir DBN ou BND).

RRG, WRG Transfert de données entre Rn et R4

RRG: Lecture du registre (Rn) (Read Register (Rn))

WRG: Ecriture dans le registre (Rn) (Write into register (Rn))

Code mnémo.	Code chiffré	Opérand	Description
RRG	01	Rn de	Lire le mot Rn (8 bits) et le mémoriser dans R4
WRG	02	0...999(i)	Ecrire en Rn le mot de R4 (8 bits)

Le format des données peut être BCD ou binaire.

LAR, SAR Transfert de données entre les blocs de registre Rn...Rn-4 et A0

LAR: Charger A0 avec le bloc de registres (Load A0 with registers)

SAR: Mémorisation de A0 dans le bloc de registres (Store A0 into registers)

Code mnémotique	Code chiffré	Opérand	Description
LAR	05	Rn de	Charger A0 avec le bloc de reg. Rn...Rn-4 (5 x 8 bits)
SAR	06	9...999(i)	Charger A0 dans le bloc de reg. Rn...Rn-4 (5 x 8 bits)

La différence avec les instructions RRG/WRG est que le transfert s'effectue pour tout le bloc de registres (5 x 8 bits ou 9 chiffres binaires). Les données peuvent également être sous forme binaire pure.

Comme déjà mentionné l'adresse Rn dans l'opérand concerne tout le bloc de registres de Rn...Rn-4. Pour une meilleure vue d'ensemble, il est recommandé de partager le registre de mots en blocs de 5 et de n'utiliser, pour ces instructions, que des adresses se terminant par 4 ou 9.

EXG: Echange A1 avec bloc de registre Rn...Rn-4: voir I 3

(i) = indexable

LAC, SAC Transfert de données entre les registres de comptage Cn et A0

LAC: Charger A0 avec compteur (Cn) (Load A0 with counter (Cn))

SAC: Mémorisation de A0 dans le compteur (Cn) (Store A0 with Counter (Cn))

Code mnémo.	Code chiffré	Opérand	Description
LAC	07	Cn de	Charger A0 avec le compteur Cn
SAC	08	256...511(i)	Mémorisation de A0 dans le Cn

Les données à transférer de A0 à Cn, ou inversement de Cn à A0, doivent être sous forme BCD.

Tenir compte de la limite de capacité de Cn: 65535. Par ce fait, le signe dans A0 ne peut pas être respecté.

Le registre de comptage est utilisé par le processeur de bits pour ses besoins en comptage ou temporisation. L'instruction LAC permet de lire, puis de transférer des valeurs de comptage ou de temporisation instantanée. Si, par contre, une valeur de temporisation doit être transférée de A0, le programme doit être établi comme suit:

```

SAC      256      Transfert A0 ---> C256 (dans le registre de comptage!)
.
.
EWP      0
.
.
STR (14) 256      Transfert de valeur dans le registre de
31      256      temporisation et démarrage de la temporisation

```

WEL, WEU Transfert d'un digit BCD dans un groupe d'éléments

WEL: Ecrire dans les éléments le digit inférieur (Write Elem. with Lower digit)

WEU: Ecrire dans les éléments le digit supérieur (Write Elem. with Upper digit)

Code mnémo.	Code chiffré	Opérand	Description
WEL	09	En de	Ecrire dans les éléments } R4(10 ⁰)
WEU	10	3...999(i)	En...En-3 les digits de } R4(10 ¹)

Cette instruction permet le transfert digital de nombres (4 bits) directement sur des indicateurs ou des sorties, comme par ex. pour une sortie multiplexée sur un affichage.

Les 4 digits peuvent avoir aussi bien le format BCD qu'une autre combinaison quelconque de bits.

(i) = indexable

RRE, WRE Transfert de registre Rn dans un groupe d'élément En

RRE: Lire le registre et écrire dans des éléments
(Read Registers and Write in Elements)

WRE: Ecrire le registre avec le contenu des éléments
(Write Registers with Elements)

Code mnémotechnique	Code chiffré	Opérand	Description
RRE	Ø3	Rn de Ø...999(i)	Lire le mot Rn (8 bits) et le mémoriser dans les éléments En...En-7. En sera adressé par A1.
WRE	Ø4	Rn de Ø...999(i)	Ecrire en Rn le contenu des éléments En...En-7 (8 bits). En sera adressé par A1.

Avant l'utilisation de cette instruction, l'adresse de l'élément En doit être disponible en A1. Cette adresse résulte souvent d'une opération arithmétique ou est prélevée dans le registre d'index.

I 5 Instructions de comptage et de saut

INR, DER Variation de la valeur du registre de 1

INR: Augmenter d'un le registre (Increment Register by 1)

DER: Diminuer d'un le registre (Decrement Register by 1)

Code mnémotechnique	Code chiffré	Opérand	Description
INR	11	Rn de Ø...999(i)	Augmenter la valeur BCD Rn de 1 et positionner "carry" si le résultat > 99
DER	12		Diminuer la valeur BCD Rn de 1 et positionner "carry" si le résultat < Ø

Ces instructions, en liaison avec SNC, permettent de constituer dans Rn un compteur BCD de grandeur selon besoin.

(i) = indexable

SNC Saut conditionnel de d'instruction suivanteSNC: Saute si carry = 0 (Skip if no carry)

Code mnémonique	Code chiffré	Opérand	Description
SNC	13	0	Saute l'instruction suivante, si "carry" = 0
		En (i)	Saute l'instruction suivante, si l'élément En = L

Ensemble avec INR et DER des enchaînements de comptage se laissent construire.

SEW Saute toutes les instructions à EWP ou NOP 1248SEW: Saute à EWP ou NOP 1248 si aucun CARRY (Skip to EWP or to NOP 1248)

Code mnémonique	Code chiffré	Opérand	Description
SEW	24	0	Saute à EWP ou NOP 1248, si "carry" = 0
		En (i)	Saute à EWP ou NOP 1248, si En = L

Si "carry" = 1 ou En = H, l'instruction SEW n'a aucune influence sur le déroulement du programme. Si "carry" = 0 ou En = L, toutes les instructions jusqu'à EWP resp. NOP 1248 seront ignorées. EWP commute en processeur de bit, NOP 1248 laisse le programme en processeur de mot.

(i) = indexable

I 6 Diverses instructions

CLK Assignment d'une source d'impulsions cycliques (base de temps externe)

CLK: Source d'horloge (clock source)

Code mnémo- nique	Code chiffré	Opérand	Description
CLK	21	En	Assignment d'une source d'impulsions cycliques au registre d'horloge
		0...999	Agit sur le registre des sec.
		1000...1999*	Agit sur le registre des min.
		2000	Renvoi à la base de temps interne

*) Si l'on ajoute 1000 à l'adresse d'élément (ne pas à confondre avec indexation) on l'adresse au registre de minutes.

Avec CLK, l'horloge est pilotée par une source d'impulsions externes (p.ex. une horloge mère) via une entrée PLC. Suivant l'opérand, ces impulsions agissent soit sur le registre de secondes soit sur celui de minutes. Les impulsions doivent avoir une durée ou un intervalle d'au moins 125ms (max. 4Hz) chacun.

En cas d'absence de CLK, l'horloge procède automatiquement à partir de l'"horloge" interne ou du module R27. L'opérand 2000 rétablit cet état.

SHI Registre à décalage

SHI: Registre à décalage (Shift register)

Code mnémo- nique	Code chiffré	Opérand	Description
SHI	22	Rn 21...999(i)	Décaler les registres "mots" (8 bits) depuis R20 jusqu'à Rn d'une adresse vers le haut

Le sommaire au début du chapitre I 4 montre le positionnement. Il est ainsi possible de réaliser un registre à décalage pour plusieurs bits.

Après l'opération de décalage l'information de R20 reste inchangée tandis que l'ancienne information de Rn est perdue.

(i) = indexable

TXT Sortie de texteTXT: Texte (Text)

Code mnémonique	Code chiffré	Opérand	Description
TXT	23	TXn de Ø...818 (i)	Démarrage de la sortie de texte

Le texte avec le numéro TXn est sorti par l'interface série jusqu'à ce que le caractère NULL (dans le texte) soit trouvé.
Pour des détails voir K 5.

NOP Pas d'opérationNOP: Pas d'opération (No Operation)

Code mnémonique	Code chiffré	Opérand	Description
NOP	ØØ	Ø	Pas d'opération

De même que dans le jeu d'instructions du processeur de bits, NOP Ø introduit des lignes libres dans le programme de mots; celles-ci sont traitées, mais n'ont pas d'action sur le programme.

EWP Fin du processeur de motsEWP: Fin processeur de mots (End Word Processor)

Code mnémonique	Code chiffré	Opérand	Description
EWP	31	Ø	Fin processeur de mots

Toute partie de programme du processeur de mots commence par l'ordre NOP 1248 et se termine par EWP Ø. Ces ordres provoquent chaque fois le passage dans le jeu d'instructions correspondant.

(i) = indexable

Notes:

PARTIE K SORTIE DE TEXTE ET COMMUNICATION

- K 1 L'interface série**
- K 1.1 Qu'est-ce qu'une interface série?**
- K 1.1.1 Qu'est-ce que le code ASCII?**
- K 1.1.2 Transmission série de données, unité BAUD, parité**

- K 2 La boucle de courant 20mA (également nommée interface TTY, interface à courant de ligne ou boucle de courant)**
- K 2.1 PCA2.M22 et M32**
- K 2.2 PCA14 et PCA02**

- K 3 Fonction de la lampe ERROR**

- K 4 Assignment de l'interface**

- K 5 Entrée/sortie de textes**
- K 5.1 Organisation de la mémoire de textes**
- K 5.2 Sortie d'un texte**
- K 5.3 Entrée de textes**
- K 5.4 Entrée/sortie de texte à 8 bits**

- K 6 Transmission de données par l'interface série**
- K 6.1 Introduction, modes de communication**
- K 6.2 Définition du mode C**
- K 6.2.1 Assignment pour le mode C**
- K 6.3 Définition du mode N**
- K 6.3.1 Assignment pour le mode N**
- K 6.3.2 Les télégrammes du mode N**
- K 6.4 Définition du mode P**
- K 6.4.1 Assignment pour le mode P**
- K 6.4.2 Les télégrammes du mode P**
- K 6.4.3 Ecriture de données dans le PCA**
- K 6.4.4 Lecture de données du PCA**
- K 6.5 Assignment pour les modes combinés**
- K 6.6 Vue générale du PAS 100 et des variantes de fonctionnement**

PARTIE K SORTIE DE TEXTE ET COMMUNICATION

K 1 L'interface série

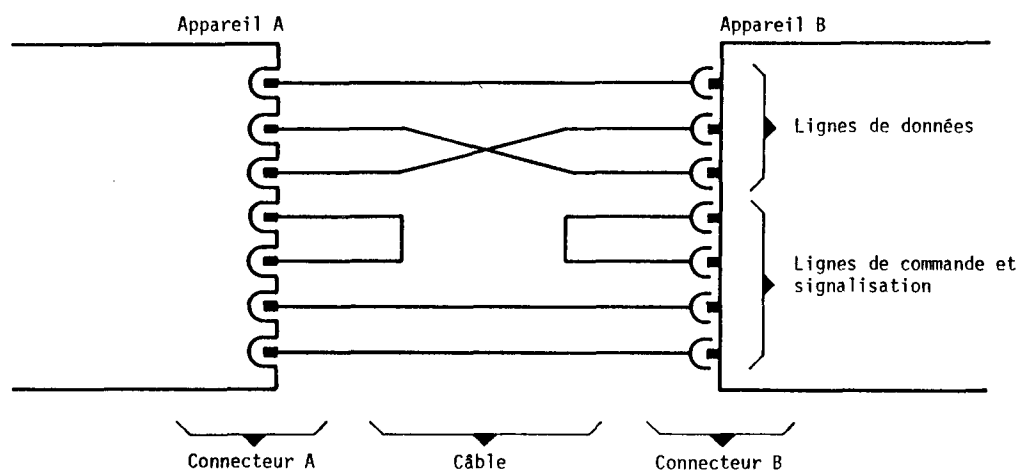
L'interface série du PCA prend en charge les tâches suivantes:

- Sortie de textes à partir de la mémoire de texte (mode TEXT)
- Entrée de textes dans la mémoire de textes par un appareil périphérique (mode EDITOR)
- Echange de caractères ASCII individuels avec un autre système (mode C)
- Echange de données numériques via des indicateurs selon DIN 66019 (mode N)
- Echange de données internes, concernant des éléments, compteurs, lignes de programme selon DIN 66019 (mode P)

K 1.1 Qu'est-ce qu'une interface série?

Si la signification des termes code ASCII, unité BAUD, bit de parité, RS232c et boucle de courant 20mA vous est familière, vous pouvez passer directement au chapitre K2. Sinon, il vaut la peine de lire attentivement ce paragraphe.

Bien qu'aujourd'hui, la plupart des appareils périphériques et des ordinateurs soient équipés d'interfaces répondant aux normes reconnues telles que par ex. EIA RS232c ou CCITT V.24, ou encore DIN 66020, resp. 66259, il surgit toujours et encore des problèmes lors du raccordement de ces appareils. La cause en est que toutes ces "normes" laissent encore un large degré de liberté, qu'il faut de cas en cas éclaircir dans le détail.



Il se pose, concernant le matériel, les questions suivantes:

- Quel genre de connecteur utiliser (nombre de pôles, mâle/femelle)?
- Comment équiper le connecteur?
- Faut-il prévoir, outre les lignes de données, des lignes de commande et de signalisation?
- Comment placer le câble?

Concernant les caractéristiques des 2 appareils, les questions suivantes, entre autres, sont à éclaircir:

- Vitesse de transmission?
- Nombre de bits d'arrêt?
- Parité?
- Nature du duplex?
- Quel appareil est actif (fournit le courant de 20mA)?
- etc.

K 1.1.1 Qu'est-ce que le code ASCII?

L'information utilise des signaux électriques qui ne peuvent prendre que les valeurs "0" et "1". Si l'on veut représenter électriquement des chiffres ou d'autres signes, la combinaison de plusieurs bits est nécessaire. 7 bits donnent $2^7 = 128$ combinaisons et permettent de représenter un nombre équivalent de signes. Le code ASCII range ces 128 signes dans un tableau divisé en "cases". Les cases "control" regroupent 32 ordres de commande, tels que par ex. "CR" = Carriage Return (retour de chariot) ou "LF" = Line Feed (avance d'une ligne) etc.

Les cases "Symbols" sont attribuées à 32 signes graphiques et aux chiffres décimaux. L'alphabet est consigné sous "Upper Case" pour les majuscules et "Lower Case" pour les minuscules.

Dans le tableau ASCII (page suivante) les signes sont numérotés en décimale de 0 à 127, notation qui nous sera utile pour l'introduction de texte avec le clavier de programmation P05. D'autres représentations utilisent également le code hexadécimal de 0 à 7F.

La désignation anglaise des ordres de commande est listée ci-dessous:

(0) NUL : Null (vide)	(8) BS : Back Space
(1) SOH : Start of Heading	(9) HT : Horizontal Tabulation
(2) STX : Start of Text	(10) LF : Line Feed
(3) ETX : End of Text	(11) VT : Vertical Tabulation
(4) EOT : End of Transmission	(12) FF : Form Feed
(5) ENQ : Enquiry	(13) CR : Carriage Return
(6) ACK : Acknowledgement	(14) SO : Shift-Out
(7) BEL : Bell	(15) SI : Shift-In
<hr/>	
(16) DLE : Data Link Escape	(24) CAN : Cancel
(17) DC1 : Device Control 1	(25) EM : End of Medium
(18) DC2 : Device Control 2	(26) SUB : Substitute
(19) DC3 : Device Control 3	(27) ESC : Escape
(20) DC4 : Device Control 4	(28) FS : File Separator
(21) NAK : Negative Acknowledgement	(29) GS : Group Separator
(22) SYN : Synchronous Idle	(30) RS : Record Separator
(23) ETB : End of Transmission Block	(31) US : Unit Separator

ASCII CODE CHART

BITS				0 0 0				0 0 1				0 1 0				0 1 1				1 0 0				1 0 1				1 1 0				1 1 1			
84	83	82	81	CONTROL				SYMBOLS				UPPERCASE				LOWERCASE																			
0	0	0	0	NUL	0	DLE	16	SP	32	0	48	@	64	P	80	\	96	p	112																
0	0	0	1	SOH	1	DC1	17	!	33	1	49	A	65	Q	81	a	97	q	113																
0	0	1	0	STX	2	DC2	18	"	34	2	50	B	66	R	82	b	98	r	114																
0	0	1	1	ETX	3	DC3	19	#	35	3	51	C	67	S	83	c	99	s	115																
0	1	0	0	EOT	4	DC4	20	\$	36	4	52	D	68	T	84	d	100	t	116																
0	1	0	1	ENQ	5	NAK	21	%	37	5	53	E	69	U	85	e	101	u	117																
0	1	1	0	ACK	6	SYN	22	&	38	6	54	F	70	V	86	f	102	v	118																
0	1	1	1	BEL	7	ETB	23	/	39	7	55	G	71	W	87	g	103	w	119																
1	0	0	0	BS	8	CAN	24	(40	8	56	H	72	X	88	h	104	x	120																
1	0	0	1	HT	9	EM	25)	41	9	57	I	73	Y	89	i	105	y	121																
1	0	1	0	LF	10	SUB	26	*	42	:	58	J	74	Z	90	j	106	z	122																
1	0	1	1	VT	11	ESC	27	+	43	;	59	K	75	[91	k	107	{	123																
1	1	0	0	FF	12	FS	28	,	44	<	60	L	76	\	92	l	108	!	124																
1	1	0	1	CR	13	GS	29	-	45	=	61	M	77]	93	m	109	}	125																
1	1	1	0	SO	14	RS	30	.	46	>	62	N	78	^	94	n	110	~	126																
1	1	1	1	SI	15	US	31	/	47	?	63	O	79	-	95	o	111	RUBOUT (DEL)	127																



Dans les cases "caractères", les 2 signes signifient:

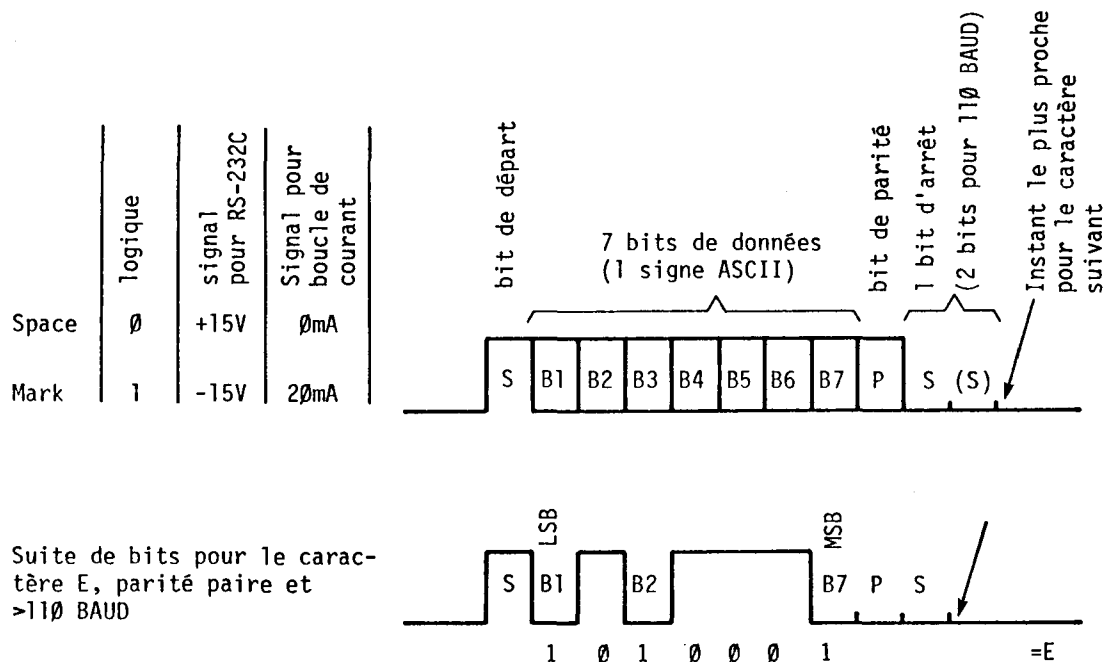
SP (no 32) = Space (espace)
 DEL (no 127) = Delete resp. Rubout (effacer)

- *) Dans la plupart des appareils périphériques, les caractères des cases "Control" peuvent être obtenus par pression simultanée sur la touche **CTRL** et celle du caractère correspondant, situé dans "Upper Case".
 Par ex. "FF" = Form Feed (no 12) est obtenu en actionnant simultanément **CTRL** und **L** ----> CTRL/L.

K 1.1.2 Transmission série de données, unité BAUD, parité

Comme l'indique le tableau ASCII le caractère E correspond à la valeur binaire 1000101 (valeur décimale 69). Cette valeur binaire peut être transmise par une ligne électrique sous forme de signaux série, dont la norme RS232c fixe le format et le niveau électrique. Un caractère est délimité par un bit de départ en son début (space) et un bit d'arrêt à sa fin (mark); seule la vitesse de 110 baud nécessite 2 bits d'arrêt.

Pour la boucle de courant 20mA, la "norme" aussi bien que l'usage reprennent les règles RS232c.



La vitesse de transmission, soit le nombre de bits qui peuvent être écoulés en 1 seconde, est exprimée en unité BAUD. Les valeurs suivantes sont standardisées:

110, 150, 300, 600, 1200, 2400*, 4800*, 9600* BAUD

Avec un format de 11 bits et une vitesse de 110 baud, il est possible de transmettre 10 caractères ASCII par seconde au maximum.

Pour contrôler la transmission, on peut ajouter aux 7 bits de caractère, un bit de parité. Ce dernier est normalement choisi pour le nombre de "1" dans la suite de 8 bits, et pour une transmission asynchrone, soit pair (voir figure). Si l'on constate un nombre impair à la réception, c'est qu'une erreur de transmission s'est produite.

*) Afin d'atteindre des vitesses de transmission élevées, le programme utilisateur doit être structuré correctement (vérification avec le programme "RTA" du PCA-ASSEMBLER conseillée).

K 2 La boucle de courant 20mA

(également nommée interface TTY, interface à courant de ligne ou boucle de courant)

Le signal électrique représentant les états logiques "0" ou "1" peut être une tension ou un courant. La norme RS232c ou V.24 prescrit une tension $\pm 15V$. Bien que cette interface soit la plus employée, elle présente en milieu industriel certains désavantages:

- Pas de séparation galvanique possible entre émetteur et récepteur.
- Vu la grande sensibilité aux parasites, les liaisons ne doivent pas dépasser 15m.

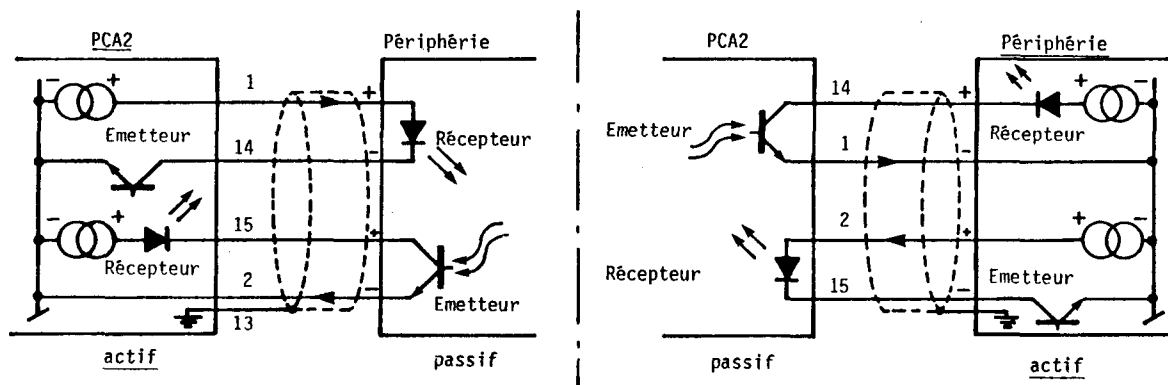
Par ailleurs, la boucle de courant offre:

- Transmission directe des 20mA par opto-coupleur possible, permettant la séparation galvanique émetteur/récepteur.
- Bonne tenue aux parasites pour des longueurs de ligne jusqu'à 1000m.
- Vitesse de transmission limitée à 9600 baud (ce qui n'a pas d'importance pour le SPS).

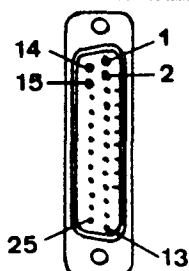
Avec l'interface boucle de courant, il faut chaque fois définir de quel côté se fait l'alimentation. Celui qui est appelé actif, est bien sûr lié galvaniquement à la source, alors que le côté passif en est séparé par opto-coupleur.

K 2.1 PCA2.M22 et M32

Par déplacement d'un connecteur multibroche (voir PCA2 Hardware), la liaison "active" ou "passive" de l'interface, pour l'adaptation de l'appareil périphérique. Le câble de liaison reliant le PCA2 avec le périphérique est à construire comme suit.



Prise sériele:



Désignation:
DATA LINES, prise mâle
à 25 pôles de type D-Sub avec
verrouillage mécanique par vis.

K 2.2 PCA14 et PCA02

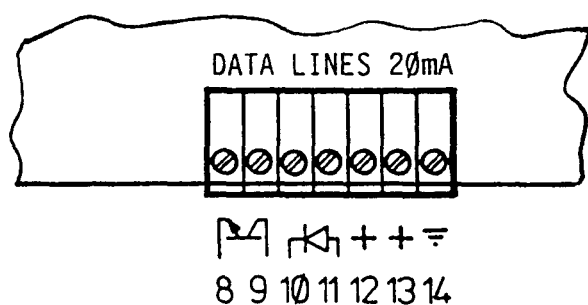
La communication série s'effectue pour ces deux types par une prise enfichable, comportant un bornier à vis. Par un câblage approprié le PCA est au choix actif ou inactif et il est également possible d'avoir l'émetteur actif et le récepteur passif.

Les PCA14 et PCA02 sont complètement compatibles tant au point de vue Hardware que Software à l'exception du câblage des bornes.

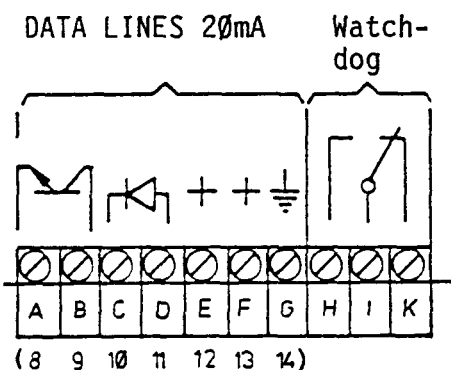
Désignation des bornes

Schéma de connexion

PCA14



PCA02



Bornes correspondante sur PCA14

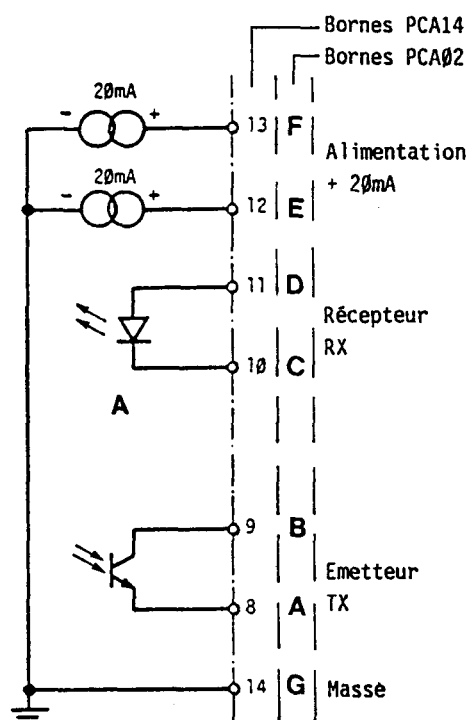
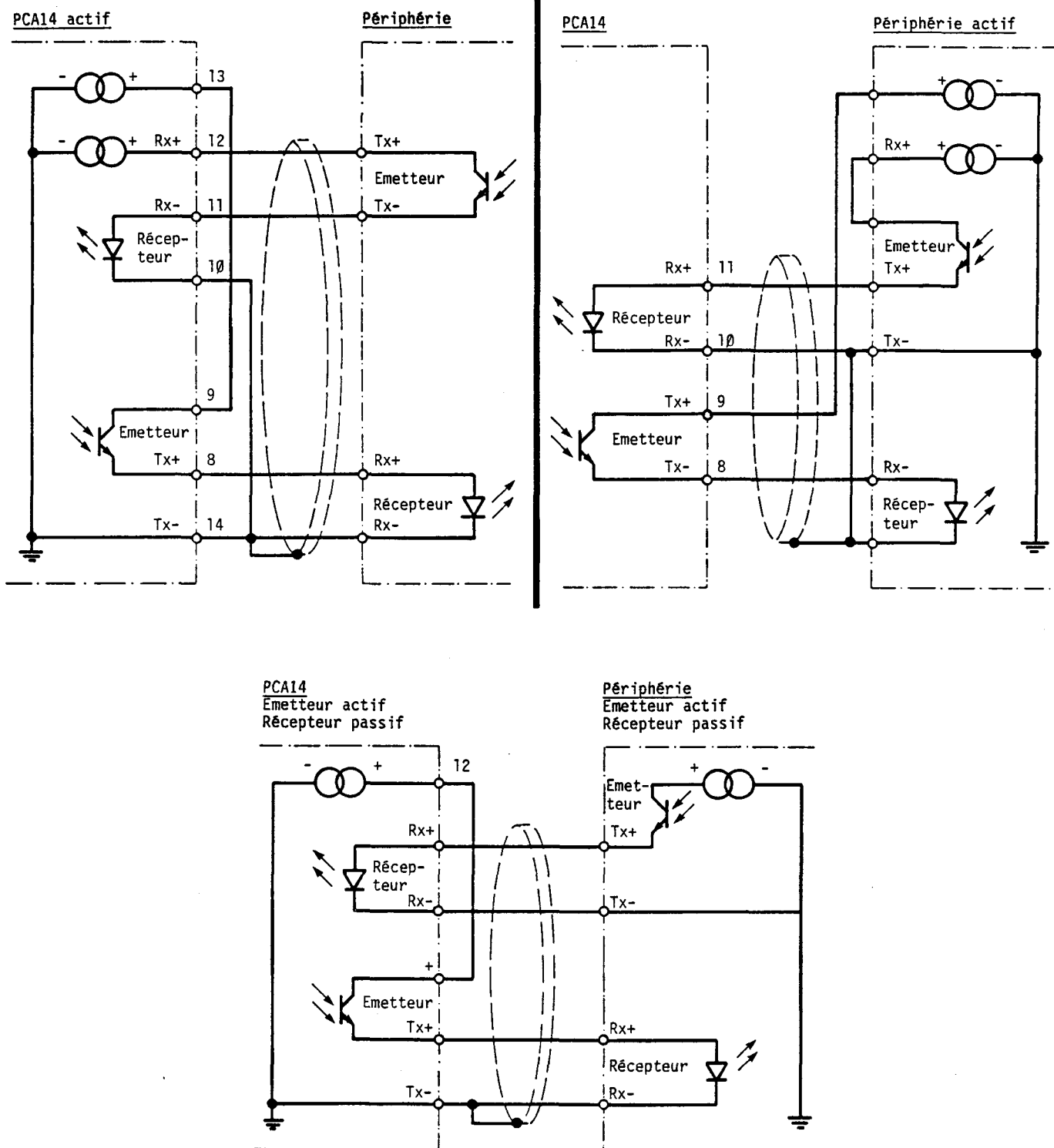


Schéma de connexion (PCA14 et PCA02)

Ci-dessous, voici les schémas du PCA14 uniquement. A partir des diagrammes et dessins de bornier du PCA02 (page précédente) il vous sera facile de relier par analogie votre PCA02 dans les 3 variantes ci-dessous.



K 3 Fonction de la lampe ERROR placée sur la face frontale

La lampe ERROR sera allumée dans les cas suivants:

- a) Le caractère reçu est faut, c'est à dire erreur de parité, framing ou overrun.
- b) Le télégramme reçu est faut, c'est à dire
 - que STC n'est pas le premier caractère
 - que le télégramme reçu ne contient pas assez ou trop de caractères
 - que le BCC est faux.
- c) Lorsque la ligne est coupée.

Pour le cas a) et b), la lampe ERROR sera éteinte à chaque dixième base de temps soit à chaque seconde pour la base de temps 1/10 ou à chaque 1/10 de seconde pour la base de temps 1/100.

K 4 Assignment de l'interface sur le CPU de PCA

Comme il ressort de l'introduction, l'échange de données série nécessite non seulement le choix correct du câble de liaison et des connecteurs, mais en outre, les divers paramètres de transmission des 2 appareils en correspondance doivent être déterminés en vue de leur ajustage réciproque. Les paramètres de transmission sont définis dans le processeur de bits par l'instruction PAS 100.

PAS 100 est une instruction à 10 lignes où chaque ligne est définie comme suit:

Ligne 1:	PAS	(29)	100	:	Assignment de l'interface
2:		00	xxxx	:	Vitesse en BAUD, parité et bits d'arrêt
3:		0x	xxx	:	Text Busy Flag (TXB)
4:		n	xxx	:	Receive Buffer Busy Flag (RBY)
5:		n	xxx	:	Transmit Buffer Busy Flag (TBY)
6:		0x	d	:	Block Check Code (somme de contrôle)
7:		00	0	}	Vide
8:		00	0		
9:		00	0		
10:		00	0		

PAS 100 sera placé en général au début d'un programme, dans la partie d'assignation.

Le mode assignié (C,N,P) peut être changé, si nécessaire, partout dans le programme utilisateur.

Ligne 1: PAS 100

Ligne 2: vitesse en BAUD, parité, bits d'arrêt

Dans l'opérand de la 2ème ligne figure la somme des valeurs des paramètres ci-après; le code est toujours 00.

Paramètres			Constantes à additionner
Vitesse en BAUD:	110 BAUD		0
	150 BAUD		1
	300 BAUD		2
	600 BAUD		3
	1200 BAUD		4
	2400 BAUD*		5
	4800 BAUD*		6
	9600 BAUD*		7
Nombre de bits de donnée par caractère:	7		0
	8		32
Génération et contrôle de parité:	inactif		0
	actif		128
Si parité active:	impair		0
	pair		256
Nombre de bits d'arrêt:	1 bit		512
	1 1/2 bits		1024
	2 bits		1536

*) Afin d'atteindre des vitesses de transmission élevées, le programme utilisateur doit être structuré correctement. Une vérification à l'aide du programme "RTA" du PCA-ASSEMBLER est recommandée.

Exemple:	1200 BAUD	4	
	7 bits de donnée	0	
	parité active	128	
	parité paire	256	
	1 bit d'arrêt	512	
		—	
	somme	900	≡ opérand de la 2ème ligne
		===	

Ligne 3: Text Busy Flag (TXB)

00 dans le code, indique une utilisation générale de l'interface.

01 dans le code, active l'éditeur pour l'entrée de texte par l'interface série (voir K 5.3).

L'opérand recevra l'adresse d'un indicateur ou d'une sortie (BUSY) qui endra l'état "H" tant qu'un texte est transmis, et reviendra à l'état "L" dès la fin de cette transmission, permettant ainsi de transférer entièrement un texte avant le traitement d'un autre texte venu d'ailleurs.

Si cependant, un texte doit être prioritaire (par ex. l'annonce d'une alarme), il est possible à tout moment, sans interroger l'élément BUSY, d'interrompre le transfert en cours pour laisser la place au texte prioritaire.

Lignes 4 et 5: mémoire-tampon de réception/émission

Le code définit la capacité de cette mémoire et l'opérand, l'adresse de l'indicateur BUSY. Pour plus de détails, voir chapitre K 6.

Ligne 6: somme de contrôle

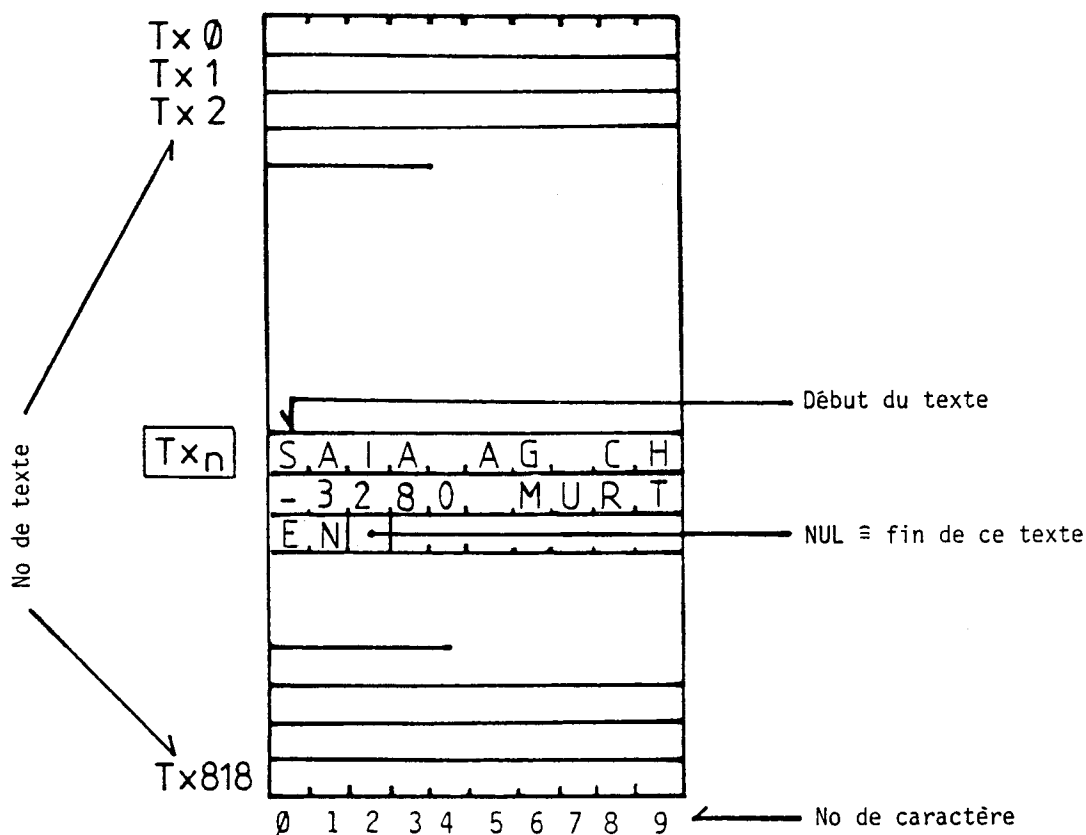
Dans le cas de transmission selon DIN 66019, le calcul d'une somme de contrôle permet d'augmenter la sécurité de transmission. Pour plus de détails, voir chapitre K 6.3.

K 5 Entrée/sortie de textes

K 5.1 Organisation de la mémoire de textes

Une partie de la mémoire texte est utilisée comme mémoire de donnée (voir page 20A manuel PCA1 Hardware ou page 24A manuel PCA2 Hardware). 4K de mémoire de donnée correspond à 818 lignes de texte à 10 caractères chacune, c'est-à-dire 8K caractères.

Chaque début de texte est adressable de Tx0...Tx818. Le premier caractère est positionné au No de caractère 0. La fin du texte est toujours désignée par NUL (zéro du code ASCII). Un texte plus long peut se composer d'autant de No de texte que l'on veut.



K 5.2 Sortie d'un texte

Tous les processeurs du niveau 2 utilisent le PAS 23. Dans le niveau 3 (PCA231 et 232) l'instruction TXT est à disposition dans le processeur de mots. L'effet est toujours le même est les instructions sont exécutées indépendamment de l'ACCU et ne l'influence pas.

Aussi bien l'instruction du processeur de bits

que l'instruction du processeur de mots (seulement PCA231/232)

PAS	(29)	23	(i)
	00	375	

TXT	(23)	375	(i)
-----	------	-----	-----

du jeu d'instruction du processeur de bits, transmet le texte à partir du No 375 de texte jusqu'au signe "NUL". Pour assurer la transmission, les paramètres correspondants doivent être tout d'abord définis et programmés:

- La ligne série peut être choisie active (recommandé) ou passive selon le raccordement effectué sur le bornier (voir manuel hardware).
- Les paramètres de transmission, tels que vitesse en BAUD, parité, etc. sont à déterminer en logiciel dans le programme-utilisateur par la 2ème ligne de l'instruction PAS 100 (voir paragraphe précédent).
- La 3ème ligne de l'instruction PAS 100 désigne un élément BUSY (indicateur du sortie) prenant l'état "H" pendant toute la durée de transmission du texte et l'état "L" dès la fin de cette transmission.
- Les lignes 4 et 5 de l'instruction PAS 100 adressent des éléments qui restent normalement à l'état "L".

Exemple:

Le texte "SAIA AG CH-3280 MORAT" débute au No de texte 375; à la fermeture du contact E7, il doit apparaître sur un terminal.

Le programme correspondant se présente comme suit:

1	PAS	(29)	100	Assignment de l'interface	
2		00	902	Paramètres pour terminal*	
3		00	32	Indication Busy (TXB)	
4		00	254	} L'adresse la plus haute de 2 éléments qui restent toujours au niveau "L"	
5		00	254		
6		00	0	} 00	
7		00	0		
8		00	0		
9		00	0		
10		00	0		
11	STH	(01)	7	Attendre fermeture contact 7	
12	DYN	(09)	400		
13	JIZ	(22)	11		
14	WIH	(25)	32	Attendre jusqu'à la fin de la transmission du texte précédent	
15	PAS	(29)	23	Transmettre le texte à partir du texte No 375	15 NOP (00) W 1248
16		00	375		16 TXT (23) W 375
17	JMP	(20)	11		17 EWP (31) W 0
					18 JMP 20 11

Sortie de texte dans le processeur de bits

Sortie de texte dans le processeur de mots

*) 4800 BAUD, 7 bits, parité active et paire, 1 bit d'arrêt

(i) = indexable

K 5.3 Entrée de textes

Les textes parviennent à la mémoire de textes RAM d'un programme utilisateur.

Pour la copie d'un texte, par ex. sur EPROM, le processus est identique à celui de la copie d'un programme utilisateur.

Lorsqu'une partie d'un texte doit être copiée, les adresses limites doivent être données comme suit:

Adresse de début $\hat{=}$ No texte \times 5
 Adresse de fin $\hat{=}$ No texte \times 5 (+4)

Exemple:

Copier un texte ayant son début au No 150 et sa fin au No 200

Adresse de début de la copie $\hat{=}$ 750 (150 \times 5)

Adresse de fin de la copie $\hat{=}$ 1004 (200 \times 5 + 4)

Signes spéciaux

Les mémoires RAM et EPROM peuvent contenir, outre des textes fixes, des signes spéciaux, à l'aide desquels des registres peuvent être lus et, en combinaison avec des textes en clair, des protocoles ou représentations graphiques peuvent être commentés.

A l'exception du signe "NUL" (\emptyset)* marquant la fin du texte, tous les signes du tableau ASCII, de 1 à 127 (1 à 7F hex) peuvent être utilisés. En général, ils seront transmis sans modification, excepté le signe \$, suivi de 1 à 4 signes.

Puisque l'on travail avec des caractères à 8 bits, il est aussi possible d'utiliser les caractères ASCII 128...255. Leurs signification et fonction est dépendante du terminal branché.

*) Le signe ASCII "NUL" peut être produit, par la majorité des terminals de la manière suivante: CTRL/@.

Signification des signes spéciaux:

Suite de signes dans la mémoire de textes		Suite de signes dans la forme où ils sont transmis
Introduc- tion alpha- numérique	Introduction avec le P05 (chiffres déc.)	
\$\$	36, 36	\$
\$C Cn *	36, 67,...	Contenu du registre de comptage Cn, 5 chiffres décimaux. Les zéros non-significatifs du nombre sont <u>affichés</u> .
\$c Cn *	36, 99,...	Contenu du registre de comptage Cn, 5 chiffres décimaux. Les zéros non-significatifs du nombre sont <u>remplacés</u> par des espaces.
\$A Rn *	36, 65,...	Contenu du bloc de registre Rn, 10 chiffres décimaux (avec signe si négatif). Les zéros du début du nombre sont <u>affichés</u> . **
\$a Rn *	36, 97,...	Contenu du bloc de registre Rn, 9 chiffres décimaux (avec signe si négatif). Les zéros non-significatifs du nombre sont supprimés et remplacés par des espaces. **

\$R Rn *	36, 82,...	Contenu de la valeur de registre Rn, 2 chiffres décimaux. **
\$r Rny	36, 114,...	Contenu de la partie LOWER ou UPPER du mot de registre, 1 chiffre décimal, y=0 --> LOWER, y=1 --> UPPER. **
\$E En *	36, 69,...	Etat des 8 éléments En...En-7, 8 caractères 1 ou 0
\$e En *	36, 101,...	Sortie du carac. ASCII formé sur les 8 En...En-7
\$S En *	36, 83,...	Si l'élément En = "H" alors le caractère sorti est "-" Si l'élément En = "L" alors le caractère sorti est "+" (PCA14) resp. "espaces" pour PCA232
\$T	36, 84	Contenu de l'horloge: : 22 : 06 / 13 : 30 : 42 semaine jour heure minute seconde (22ème) (SAM)
\$H	36, 72	83 - 06 - 04 / 13 : 30 : 42 année mois jour heure minute seconde
\$D	36, 68	83 - 06 - 04 année mois jour

*) voir à la page suivante

**) voir à la page suivante

***) seulement possible avec PCA231/232

\$ xxx ¹⁾	36, ,...	<p>Répétition du caractère suivant immédiatement le signe \$, aussi souvent que xxx l'indique:</p> <ul style="list-style-type: none"> - xxx ≤ 127 Nombre de répétitions correspondant à xxx - xxx = 256...511 Nombre de répétition correspondant au contenu du compteur adressé (max. 127)
		<p>Caractère à répéter. Tous les caractères ASCII (NUL également) peuvent être utilisés, excepté les suivants, qui sont réservés pour le déclenchement des fonctions: \$, C, c, D, E, e, H, L, T, U, A, a, R, r</p>
\$L xxx ²⁾	36, 76,...	Saut au texte No xxx avec mémorisation de l'adresse de départ (sous-routine texte semblable à JMS)
\$U	36, 85	Retour au texte principal (semblable à RET, mais pour un plan ³⁾ seulement)



Partie L
Exemples
1,2,3,4 et 11

- ¹⁾ Pour Cn, Rn, En ou xxx, 3 caractères sont nécessaires, ex. 027.
- ²⁾ Pour les valeurs décimales entre 10 et 15 de mots de 4 bits formatés binaires, les caractères suivants sont émis:
10 ---> : 11 ---> ; 12 ---> < 13 ---> = 14 ---> > 15 ---> ?
- ³⁾ Dès certaine version firmware, 3 niveaux de sous-programme sont à disposition: PCA02: V6.132; PCA14: V6.034; PCA222: V6.230.

Introduction de texte:

Pour l'introduction dans la mémoire de texte proprement dite, 2 possibilités sont offertes:

- avec le clavier de programmation P05/P10 ou P21 enfiché sur la prise 25 pôles prévue à cet effet.
- avec un appareil périphérique, disposant d'une interface boucle de courant, raccordé au bornier de connexion de la ligne série ("DATA LINES").

a) Introduction de texte avec clavier de programmation PCA2.P05/P10

Le clavier de programmation est enfiché comme décrit ci-dessus. Le sélecteur de mode de fonctionnement doit occuper la position TEXT. L'introduction s'effectue dans la même manière que sous PROG.

Par ex.: Introduire le texte "SAIA AG CH-3280 MORAT" en partant du No de texte 375:

No de texte		No du code ASCII sous forme décimale		Texte
	No du caractère			
[A]	3750 *	[E]	83	S
	(3751)**		65	A
	(3752)		73	I
	(3753)		65	A
	(3754)		32	
	(3755)		65	A
	(3756)		71	G
	(3757)		32	
	.		.	.
	.		.	.
	(3776)		13	"CR"
	(3777)		10	"LF"
	(3778)		0	"NUL" = fin de texte

*) A noter, que le No du caractère est à introduire; p.ex.: pour le début du texte 375 ----> 3750.

**) Les adresses consécutives de caractères ne doivent pas être introduites; elles sont automatiquement affichées.

b) Introduction de texte avec un appareil périphérique disposant d'une interface boucle de courant

Après la mise en place du câble et définition des paramètres de transmission, le périphérique est raccordé au connecteur DATA LINES. Le sélecteur de modes de fonctionnement P05 peut occuper toutes les positions sauf RUN et BREAK.

Conditions de logiciel:

L'utilisateur doit s'assurer que le CPU a bien traité les 10 lignes de l'instruction PAS 100, la 3ème de ces lignes contenant le code-opération 01. Ainsi, le CPU traitera tous les caractères par son éditeur de texte (voir l'exemple d'application suivant).

Signification des caractères de l'éditeur de texte

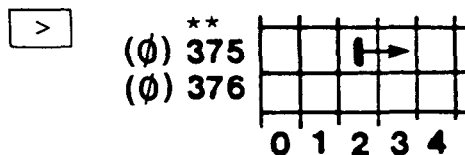
L'éditeur de texte dispose de 2 modes de fonctionnement:

- mode affichage de texte
- mode introduction de texte
- Mode affichage de texte

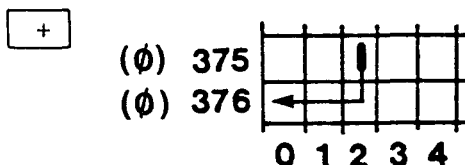
A l'aide de quelques instructions spéciales, ce mode permet d'afficher sur un écran le texte introduit ou bien, à l'aide d'un "pointeur" apparaissant sur l'écran, de sauter dans diverses parties du texte.

Txn CR **

Introduction d'un No de texte xxx (seuls les 3 derniers chiffres sont significatifs), suivi de "CR". Le "pointeur" est positionné au début du texte No xxx et le No de texte correspondant avec son contenu (10 caractères) est affiché.



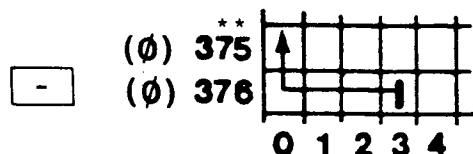
Le pointeur est déplacé d'une position à droite dans la suite des No de caractère. Une pression prolongée de la touche provoque plusieurs répétitions du processus* et permet de faire apparaître le texte mémorisé d'une manière continue sans interlignes.



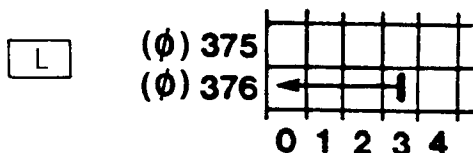
Le pointeur est déplacé vers le bas, au début du No de texte suivant. Une pression prolongée de la touche provoque plusieurs répétitions du processus* et permet de faire apparaître le texte mémorisé en progression par le haut avec chaque fois un interligne.

*) La fonction de répétition dépendante du périphérique.

**) Une entrée à 4 positions est nécessaire pour le PCA232, ainsi le texte 375 doit être donné comme ceci 0375.



Le pointeur est déplacé vers le haut, au début du No de texte précédent. Une pression prolongée de la touche provoque plusieurs répétitions du processus* et permet de faire apparaître le texte mémorisé en progression par le bas, avec chaque fois un interligne.



Le pointeur est ramené au début du No de texte actuel, dont le contenu est affiché.

CTRL/T

Cette commande permet d'accéder au mode introduction de texte.

- Mode introduction de texte

Un choix préalable d'un No de début de texte permet, dans ce mode, d'introduire en continu le texte avec le clavier du terminal.

^

Si ce signe précède "CR", ce dernier sera interprété dans le texte comme indication de retour du chariot et non comme ordre de retour au mode affichage de texte.

BS

Le PCA232 permet l'usage de la touche "BS" (Back Space) pour modifier le dernier caractère introduit. C'est-à-dire, le caractère se trouvant sous le curseur est effacé.

CR ou **CTRL/M**

Ces deux signes ont la même action de retour au mode d'affichage de texte.

! Remarque: Avant d'entreprendre d'autres activités, il faut absolument sortir de l'éditeur par l'introduction d'un "CR". **!**

Exemple: Introduction d'un texte avec un appareil périphérique à partir de No de texte 30.

Normalement, le programme commence par l'assignation des programmes parallèles suivie de PAS 100, introduisant l'assignation de l'interface, conformément aux prescriptions de service.

Pour l'introduction proprement dite du texte, un 2ème bloc PAS 100, suivi par l'instruction à deux lignes PAS (29) 23
00 30

sera placé à la fin de la mémoire-programme, par ex. à partir du pas 4001. Dans le mode STEP, l'on aura alors:

- saut à l'adresse 4001 pour activer l'éditeur de texte.
- transfert du texte introduit pour contrôle.

*) La fonction de répétition dépendante du périphérique.

**) Une entrée à 4 positions est nécessaire pour le PCA232, ainsi le texte 375 doit être donné comme ceci 0375.

Activation de l'interface pour entrée de texte au moyen de l'éditeur:

4001	PAS	(29)	100	; Paramètre pour 4800 Baud, 7 bits, parité paire,
4002		00	902	1 stop bit
4003		01	63	; 01 pour l'éditeur de texte; 63 pour Text Busy (TXB)
4004		00	254	} L'adresse la plus haute de 2 éléments qui restent
4005		00	254	
4006		00	0	toujours au niveau "L" (254 et 253)
.	.	.	.	} Zéro
.	.	.	.	
4010		00	0	
4011	PAS	(29)	23	} Sortie du texte No 30
4012		00	30	
4013	JMP	(20)	0	
4014		00	4011	

A partir du No de texte 30, nous voulons entrer le texte suivant:

SAIA-PLC
COURONNE PAR LE PCA232

Procédé:

- 1) Enficher clavier de programmation P05 dans la prise PROG-UNIT et positionner le sélecteur des modes de fonctionnement sur "STEP".
- 2) Introduire A 4001 + sur le clavier de programmation P05 pour sauter jusqu'au début de l'assignation.
- 3) Avec + procéder jusqu'à l'adresse de pas 4011.
- 4) Introduire par le clavier du VDU 030 "CR". **

Affichage sur le VDU
(chaque introduction par les
touches est soulignée)

Description

030 "CR" **

030:.A.B. . . . ^@^@^@^@
030:

Le texte ci-contre provient accidentellement d'une partie non effacée de la mémoire. Un point précédant un caractère signifie que celui-ci provient de la partie visible du tableau ASCII (32...127). Les ordres de cases sont précédés d'une flèche ou d'un ^.

**) Une entrée à 4 positions est nécessaire pour le PCA232, donc les 4 derniers caractères sont significatifs.

Affichage sur le VDU
(chaque introduction par les
touches est soulignée)

CTRL/T

- 5) Le mode "entrée de texte" est obtenu par pression simultanée sur les touches **CTRL** et **T** (CTRL/T).
- 6) Ensuite, prévoir 8 espaces libres à l'aide du signe **\$** puis introduire le nom SAIA°PLC.
- 7) On observe qu'après l'introduction de 10 caractères, le pointeur passe automatiquement au No de texte suivant.
- 8) Pour pouvoir entrer "CR" ou en alternative "CTRL/M", sans changer le mode de traitement du texte, ces signes doivent être précédés de **^**. Dans les 2 cas, il apparaît **^ M** (CTRL/M).
- 9) Pour introduire "LF = Line Feed", utiliser **CTRL/J**, correspondant à "LF", il apparaît **^ J**.
- 10) Le PCA232 permet l'usage de la touche "BS" (Back Space) pour modifier le dernier caractère introduit. C'est-à-dire le caractère se trouvant sous le curseur est effacé.
- 11) Pour terminer le texte, introduire "NUL", ce qui est possible par **CTRL/@** (éventuellement une autre commande pour former "NUL" selon l'appareil périphérique utilisé).
- 12) Pour sortir du mode de l'éditeur, entrer **"CR"**.
- 13) Pour afficher les caractères sous forme de texte lisible, actionner plusieurs fois la touche **+** du clavier de programmation P05. La boucle de programme 4011...4014 est ainsi parcourue et l'instruction PAS 23 active la sortie de texte.

```

30 .$. .0.0.8.S.A.I.A.-
31 .P.L.C^M^J.C.O.U.R.O
32 .N.N.E. .P.A.R. .L.E
33 . .P.C.A.2.3.2^M^J^@

```

SAIA-PLC
COURONNE PAR LE PCA232

Resumé de l'éditeur de textes

a) Activation de l'interface pour entrée de textes au moyen de l'éditeur

4001	PAS	(29)	100	
4002		00	902	Paramètres de transfert
4003		01	63	TXB (01) ---> activer l'éditeur
4004		00	254	L'adresse la plus haute de 2 éléments qui restent toujours au niveau "L" (254 et 253)
4005		00	254	
4006		00	0	Zéro
.	.	.	.	
4010		00	0	Sortie de texte en mode STEP
4011	PAS	(29)	23	
4012		00	Txn	
4013	JMP	(20)	0	
4014		00	4011	

b) A l'aide du clavier de programmation P05 et en mode STEP

Sauter à l'adresse 4001 et travailler jusqu'à 4011 (activation de l'éditeur).

c) Mode affichage de texte (Affichage du contenu de la mémoire de texte)

Txn Le contenu de la ligne de texte No Txn sera affiché.

<input type="text" value=">"/>	(0) 375	
	(0) 376	
<input type="text" value="L"/>	(0) 375	
	(0) 376	
<input type="text" value="+"/>	(0) 375	
	(0) 376	
<input type="text" value="-"/>	(0) 375	
	(0) 376	
		0 1 2 3 4 5 6 7 8 9

d) Mode introduction de texte

Cette commande permet d'introduire un nouveau caractère ou texte à partir du No de caractère Txn, sélectionné en mode affichage.

Effacer un caractère (seulement PCA232).

Sera compris comme "CR" dans la mémoire de texte.

Changement en mode affichage.

e) Sortie de texte

En mode STEP actionner deux fois la touche sur le clavier de programmation P05, chaque fois le texte affiché pour contrôle.

K 5.4 Entrée/sortie de texte à 8 bits

Le nombre de bits de données par caractère peut être de 7 (correspondant aux codes ASCII 0...127) ou 8 (0...255), c'est-à-dire que l'on peut travailler au choix avec 7 ou 8 bits de données par caractère. La présélection a lieu dans la deuxième ligne de l'instruction PAS 100. La signification des caractères supplémentaires 128...255 est dépendante du périphérique utilisé (ordinateur, terminal graphique) et est différente d'un appareil à l'autre.

Entrée de texte

L'introduction du texte peut se faire soit avec le clavier de programmation P05/P10/P21 ou avec un appareil périphérique équipé d'une interface boucle de courant.

a) Introduction du texte avec le clavier de programmation P05/P10 par le connecteur PGU

L'introduction se fait de la même que précédemment sous forme décimale, les signes ASCII, correspondant à 0...127, ainsi que les valeurs 128...255 sont possibles.

b) Introduction du texte avec un appareil périphérique par l'éditeur du PCA

Pour effectuer l'introduction du texte de cette manière, il faut utiliser un appareil périphérique possédant une interface série et pouvant générer le code à 8 bits de donnée par caractère. Du côté PCA, il faut définir dans la 2ème ligne du PAS 100, le numero de bits de donnée par caractère (7 ou 8 bits).

Sortie de texte

a) Affichage du texte contenu dans la mémoire de texte par l'éditeur de texte du PCA

Indépendamment du fait que l'appareil périphérique ait 7 ou 8 bits de donnée par caractère, les 256 signes seront sortis par l'éditeur de texte de la manière suivante:

No du caractère	Sortie par l'éditeur	Sortie de texte par le prog. utilis.
0...31	^ <u> </u>	Code ASCII (comme actuellement)
32...127	. <u> </u>	
128...159	! <u> </u> ¹⁾	Pas défini, différent par périphérique
160...255	: <u> </u> ²⁾	

 Caractère

¹⁾ Le caractère qui apparaît est un signe ASCII compris entre 64 et 95

²⁾ Le caractère qui apparaît est un signe ASCII compris entre 32 et 127

b) Sortie de texte par le programme utilisateur

Il est possible de sortir des caractères avec 7 ou 8 bits de données, selon les besoins pour la communication. En outre il est possible d'asservir par ex. un terminal graphique en utilisant les 128 caractères supplémentaires.

K 6 Transmission de données par l'interface série

K 6.1 Introduction, mode de communication

Le chapitre précédent traitait des entrées/sorties de textes. Alors que l'entrée de texte est une fonction "off-line" pure (comme la programmation de la mémoire-utilisateur), la sortie de texte peut se dérouler "on-line". Les textes sont appelés par un signal d'entrée par ex., générés via le programme-utilisateur et transmis par l'interface série. Il s'agit donc d'un transfert d'information à sens unique du SPS vers un appareil périphérique (application typique: protocole). Un dialogue entre périphérique et SPS n'est pas encore possible avec la transmission de texte seule.

Ce chapitre traitera les modes d'opération de l'interface série qui permettent un véritable dialogue (échange d'informations dans les 2 sens) entre les appareils suivants:

- PCA14 *) avec appareils périphériques "non intelligents" tels que TTY ou VDU
- PCA14 *) avec PCA0, PCA14, PCA23
- PCA14 *) avec un autre système "intelligent" (par ex. ordinateur)

Cet échange d'informations est divisé en modes selon les définitions suivantes:

Mode C Echange de caractères ASCII isolés

L'interface série permet l'échange de 128 signes différents qui peuvent être logés dans les indicateurs du PCA et être attribués à diverses fonctions-utilisateur. Le mode C convient particulièrement au dialogue entre le PCA et l'utilisateur de l'appareil périphérique qui est raccordé (TTY ou VDU). Dans le PCA les caractères sont lus et enregistrés dans un registre bien défini.

Mode N Echange de données numériques sous la forme de télégramme

Le mode N respecte totalement la norme DIN 66019 correspondant à l'échange d'information. Pour la transmission d'un télégramme, jusqu'à 31 caractères numériques au max. peuvent être utilisés, ce qui élargit considérablement le domaine d'application. D'une part, il est possible par ex. de transférer d'un poste de lecture des informations codées et après réception, de les mémoriser et de les exploiter. D'autre part, il est également possible de correspondre avec un autre système "intelligent".

Pour des installations plus simples, une variante est disponible, qui permet également l'échange avec l'opérateur de données numériques par l'intermédiaire d'un périphérique "sans intelligence". Cette variante sera surtout utile dans les phases de programmation et mise en service. Dans le mode N, il y a 3 variantes: N1, N2 et N3. Voir explications dans les pages suivantes.

*) ou d'autres PCA avec sortie série, comme les PCA02, PCA222 et PCA232.

Mode P Echange de données, orientées PCA, par télégramme selon DIN 66019

Dans les modes d'utilisation C et N, il est déjà possible d'accéder indirectement aux entrées, sorties, indicateurs, compteurs et temporisateurs. Dans l'échange de données selon le mode P, cet accès est direct et s'étend également aux pas d'adresse et aux opérandes. Il est ainsi possible d'ouvrir l'accès au PCA un système hiérarchiquement supérieur, et ceci à un niveau de fiabilité plus élevé. Le PCA est toujours passif. Il y a deux variantes distinctes P1 et P2.

Combinaison des divers modes d'opération

La combinaison des divers modes d'utilisation entre eux et avec la transmission de textes présente plusieurs aspects intéressants:

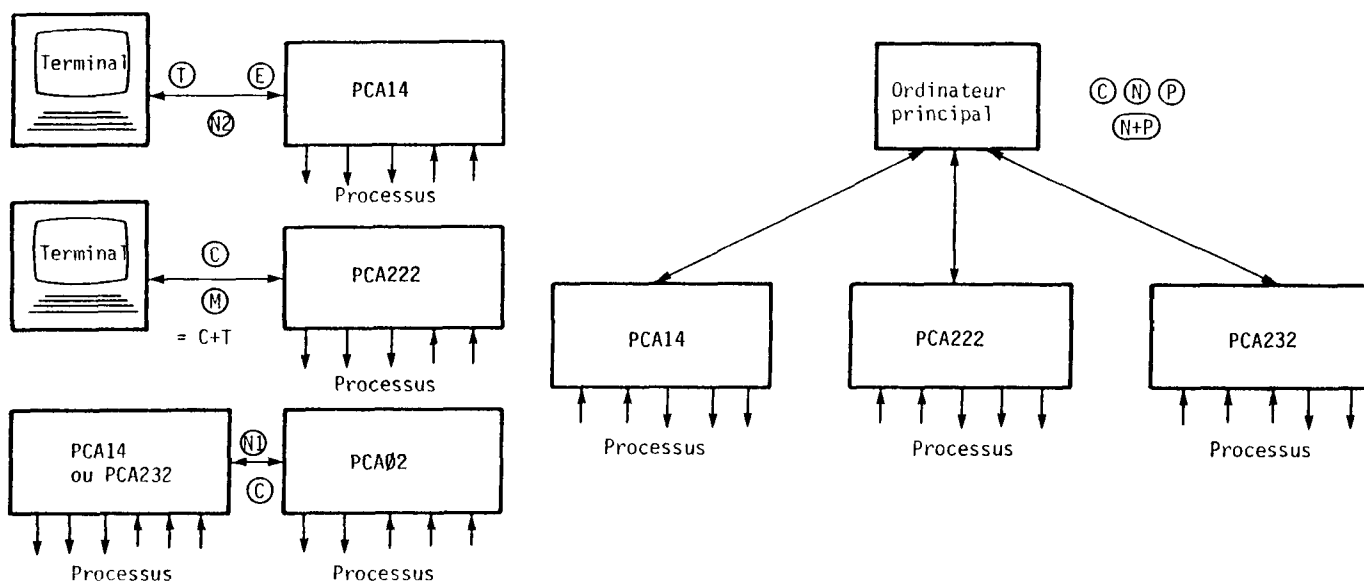
Mode "menu" = combinaison d'une transmission de texte avec le mode C (caractères isolés)

Cette combinaison permet un dialogue vivant, avec sortie de textes en clair, entre le PCA et l'utilisateur du périphérique (guidé par menu). Un texte est sorti, formant un menu. Par l'envoi de caractères uniques, le choix d'un menu peut être fait et la fonction en rapport est exécutée.

Mode "ordinateur" = combinaison des modes N et P

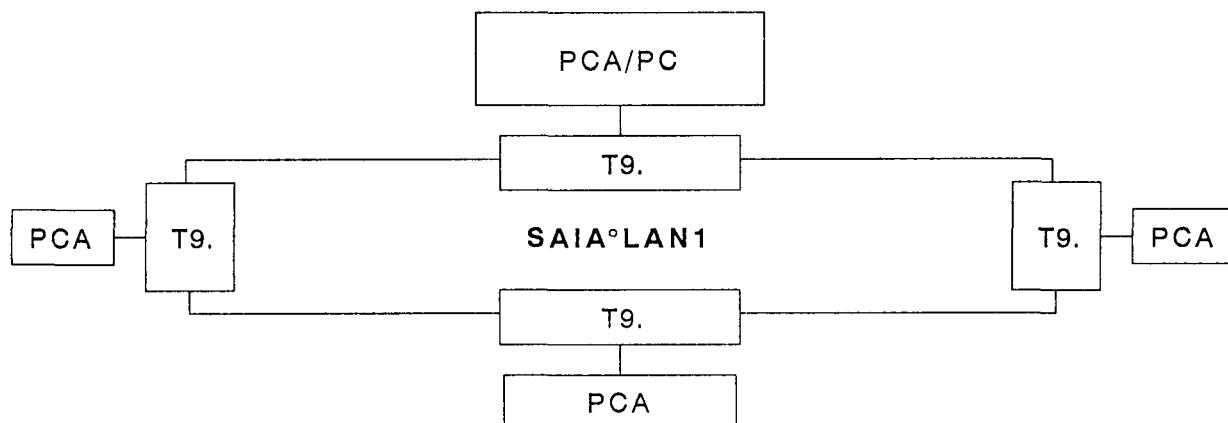
Par échange de données numériques et spécifiques au PCA et par combinaison entre elles, il est possible de faire intervenir dans le PCA un système qui lui soit hiérarchiquement supérieur, ce dernier recevant en retour des informations du PCA. La structure de commande de cette "intelligence répartie" est ainsi facilement réalisable.

Combinaisons d'appareils pour une liaison point à point



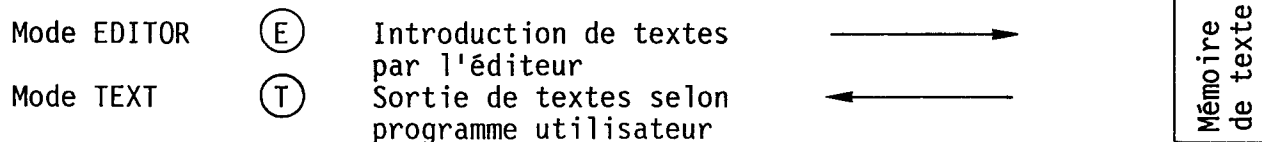
SAIA°LAN 1

Tous les modes de communication inclus la sortie de texte peuvent être employés dans le réseau local SAIA°LAN 1. Dès que le choix d'une station réceptrice est faite, la communication devient transparente, comme dans une liaison point à point. A ce propos, veuillez demander la documentation complète concernant le SAIA°LAN 1.

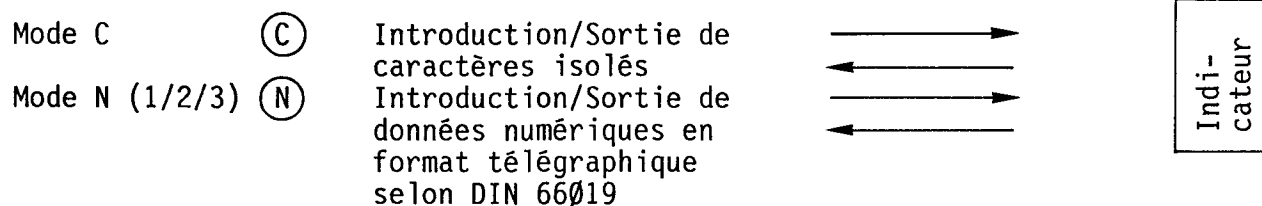


Résumé des modes de fonctionnement de l'interface série

Introduction/sortie de textes au/de mémoire de texte

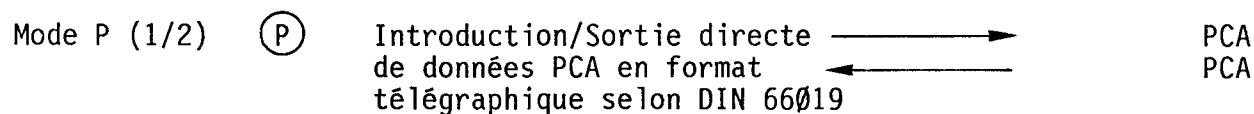


Echange d'information par registres d'indicateurs (Flag Buffer)



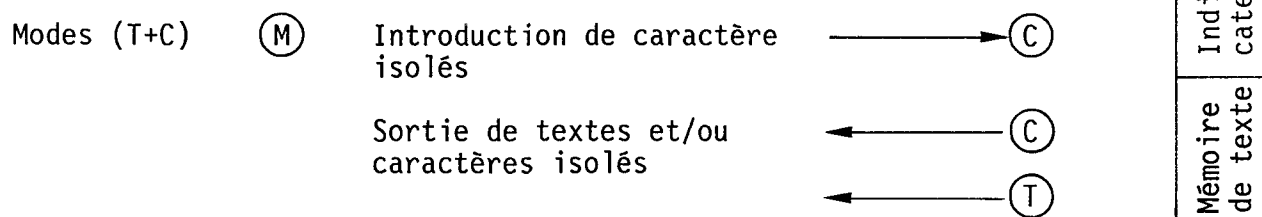
C et N dépendant du programme utilisateur (Flag-Buffer)

Transfert de données en direct avec le PCA

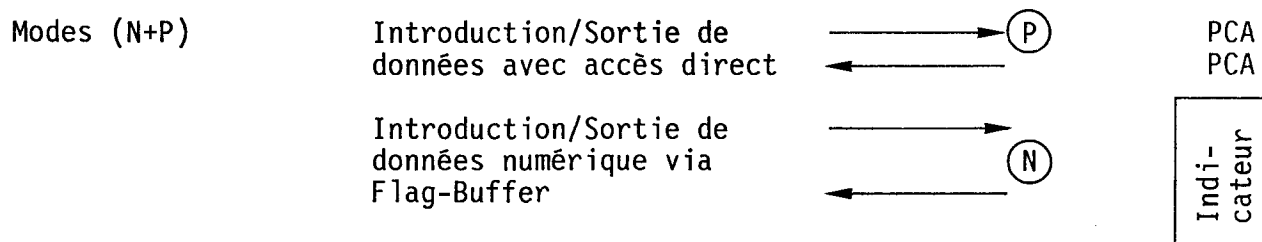


P est indépendant du programme utilisateur (Master-Slave)

Conduite par menu



Communication avec un système d'ordre supérieur

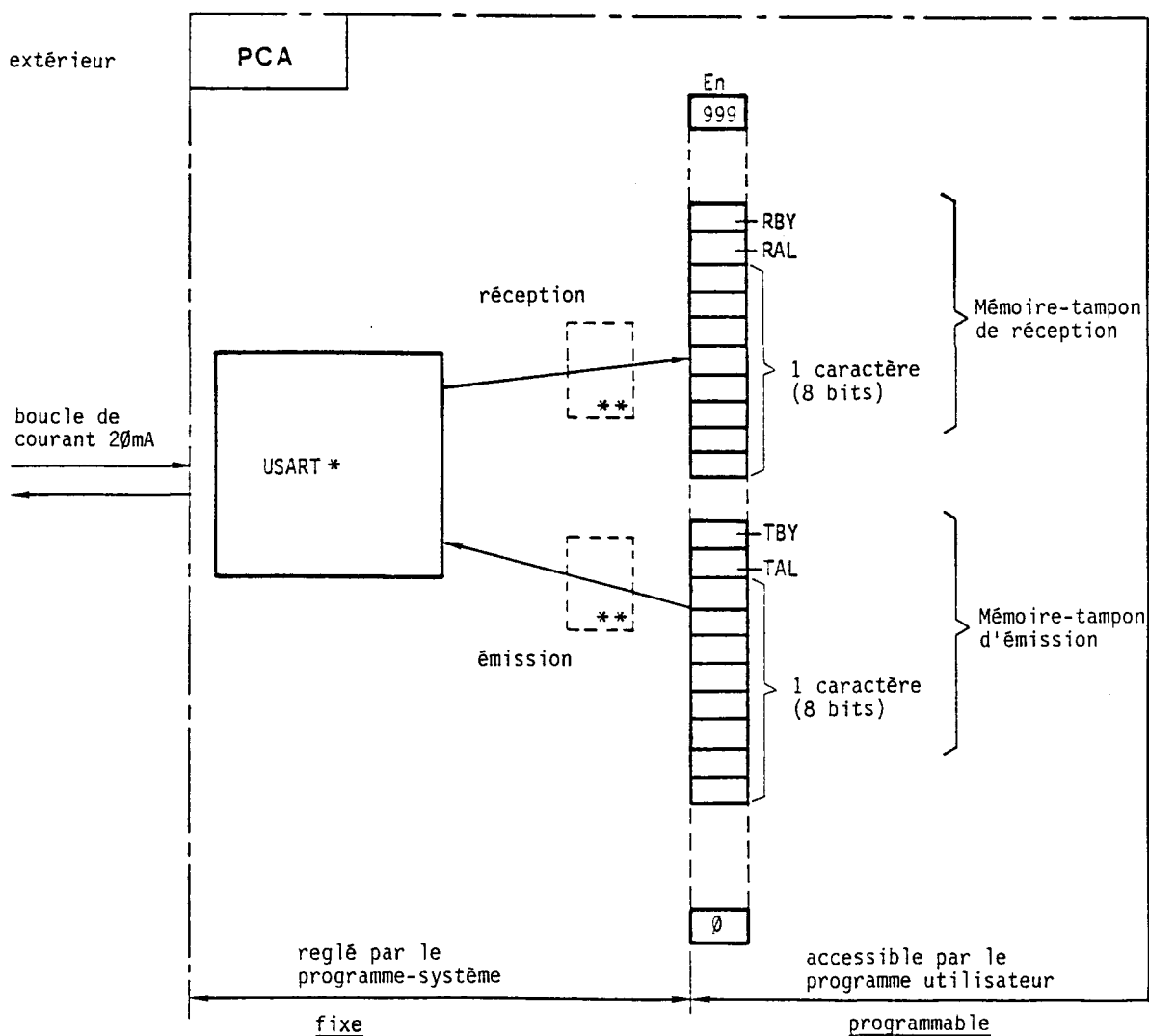


K 6.2 Définition du mode C

Il s'agit ici d'une réception et d'envoi de caractères ASCII uniques. L'activation de la ligne série est faite par le programme système, ainsi que l'interprétation des caractères reçus et le formatage des caractères émis. Le programme système utilise des registres d'indicateurs comme intermédiaire entre émission/réceptions et le programme utilisateur.

Comme le montre la figure ci-dessous, l'utilisateur doit définir 2 domaines d'indicateurs, l'un pour la mémoire-tampon de réception (emplacement de lecture des caractères), l'autre pour celle d'émission (emplacement de sortie des caractères).

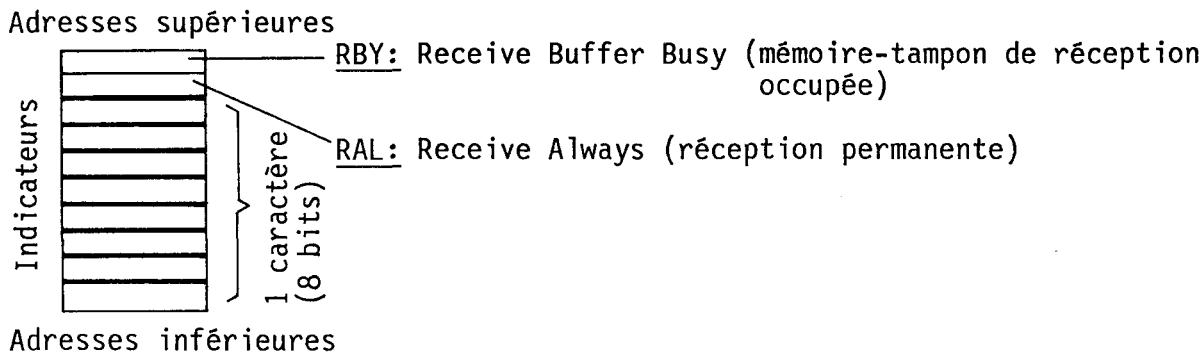
Le schéma ci-dessous en explique le fonctionnement:



*) USART = Universal Synchronous/Asynchronous Receiver Transmitter.

**) Le télégramme est tout d'abord déposé dans une mémoire-tampon où il sera examiné. Si le contrôle est positif, le programme système procédera au transfert dans la plage d'indicateurs correspondante.

Mémoire-tampon de réception pour le mode C



Si aucune erreur de transmission (telle que parité, cadrage etc.), n'est constatée à la réception, le dépôt s'effectue dans la mémoire-tampon de réception en fonction de l'état logique de RBY et RAL:

	RAL	RBY	1ère conséquence	2ème conséquence
Réception permanente	H	L	mémorisation interméd. dans la mémoire-tampon	RBY prend l'état "H"
	H	H		
Réception individuelle	L	L	mémorisation interméd. dans la mémoire-tampon	RBY prend l'état "H"
	L	H	sans effet	--

Réception permanente:

Si le programme-utilisateur définit l'état RAL = "H" et pour une réception sans erreur des caractères, ces dernières seront toujours déposées dans la mémoire-tampon de réception.

Réception individuelle:

Si par contre c'est l'état RAL = "L" qui est défini, et toujours pour une réception sans erreur des caractères, ces dernières ne seront déposées dans la mémoire-tampon de réception que pour RBY = "L". Après quoi, RBY prend automatiquement l'état "H", évitant que de nouvelles caractères ne viennent se superposer dans la mémoire. Le programme-utilisateur traite tout d'abord les caractères de la mémoire-tampon, puis positionne l'état RBY = "L" pour permettre l'entrée de caractère suivante.

N.B.: Le programme-système lui-même ne produit aucun écho; si un tel écho est nécessaire, il devra être déclenché par le programme-utilisateur.

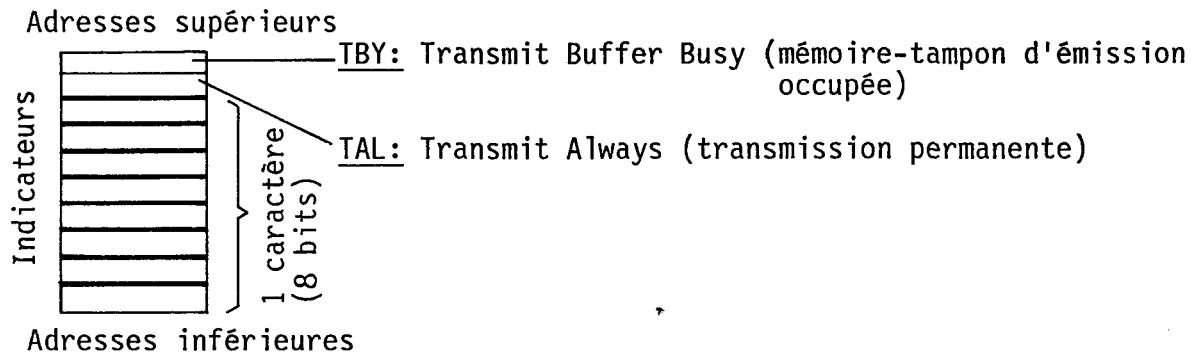
Structure de programme pour réception de données (RAL = L)

WIL RBY ; Attendre si la réception d'un nouveau caractère n'a pas encore eu lieu.

Lecture de la mémoire-tampon de réception ; Le caractère reçu est lu par le programme utilisateur et éventuellement transféré aux compteurs.

REO RBY ; Mémoire-tampon de réception libérée pour recevoir de nouveaux caractères.

Mémoire-tampon d'émission pour le mode C



Chaque 1/10 ou 1/100 s, selon la base de temps choisie, l'USART peut donner l'ordre d'émission de données au dépend de l'état logique des indicateurs TBY et TAL:

	TAL	TBY	1ère conséquence	2ème conséquence
Transmiss. permanente	H	L	transmission du caractère depuis la mémoire-tampon	→ TBY prend l'état "L"
	H	H		
Transmiss. individuelle	L	L	sans effet	--
	L	(H)	transmission du caractère	→ TBY prend l'état "L"

Transmission permanente:

Si le programme-utilisateur définit l'état $TAL = "H"$ le caractère fixé par le contenu de la mémoire-tampon est émise à chaque 1/10 ou 1/100 s.

Transmission individuelle:

Si par contre c'est $TAL = "L"$ qui est défini, le caractère stocké dans la mémoire-tampon d'émission ne sera émis que si le programme-utilisateur fixe l'état $TBY = "H"$. Il s'en suivra automatiquement, sous l'effet du programme-système, le passage à "L" de TBY, pour annoncer au programme-utilisateur la fin de la transmission.

Le programme-système n'attend aucun signal en retour à l'émission de caractère.

Structure de programme pour l'émission de données ($TAL = L$)

```

      WIH   TBY           ; Emission de données précédentes terminée?

      [ Introduire
        les données
        dans la mémoire-
        tampon d'émission ] ; Le programme utilisateur transmet le caractère
                             reçu (8 bits) dans le mémoire-tampon d'émission.

      SEO   TBY           ; Emission du contenu de la mémoire-tampon
  
```

K 6.2.1 Assignment pour le mode C

1:	PAS (29)	100 ;	Assignment de l'interface	
2:	00	xxxx ;	BAUD, parité et nombre de bits d'arrêt	
3:	00	xxx ;	TXB (Text-Busy Flag)	} Caractéristique du mode C
4:	00	xxx ;	RBV (mémoire-tampon de réception)	
5:	00	xxx ;	TBY (mémoire-tampon d'émission)	
6:	00	0 ;	aucun BCC	
7:	00	0	} Zéro	
8:	00	0		
9:	00	0		
10:	00	0		

- Ligne 3: TXB (Text-Busy Flag)

Pour xxx introduire l'adresse d'un indicateur (ou d'une sortie) libre, qui prend l'état "L" lors de la mise en service du PCA.

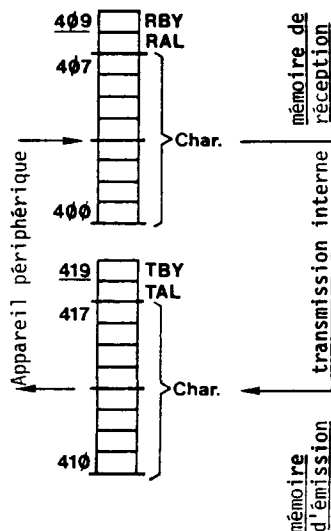
- Ligne 4: RBV (mémoire-tampon de réception)

Pour la mémoire-tampon de réception réserver 10 indicateurs (ou sorties) consécutives et libres, dont l'adresse la plus haute sera xxx.

- Ligne 5: TBY (mémoire-tampon d'émission) voir RBV, ligne 4

La mémoire-tampon de réception ainsi que celle d'émission sont toujours à définir, même si une seule des deux est nécessaire.

Exemple: Génération d'un "écho" par le programme utilisateur. Le caractère reçu par la mémoire-tampon de réception est transmis à celle d'émission et apparaît sur le VDU.



Dialogue "menu" avec écran VC 4404

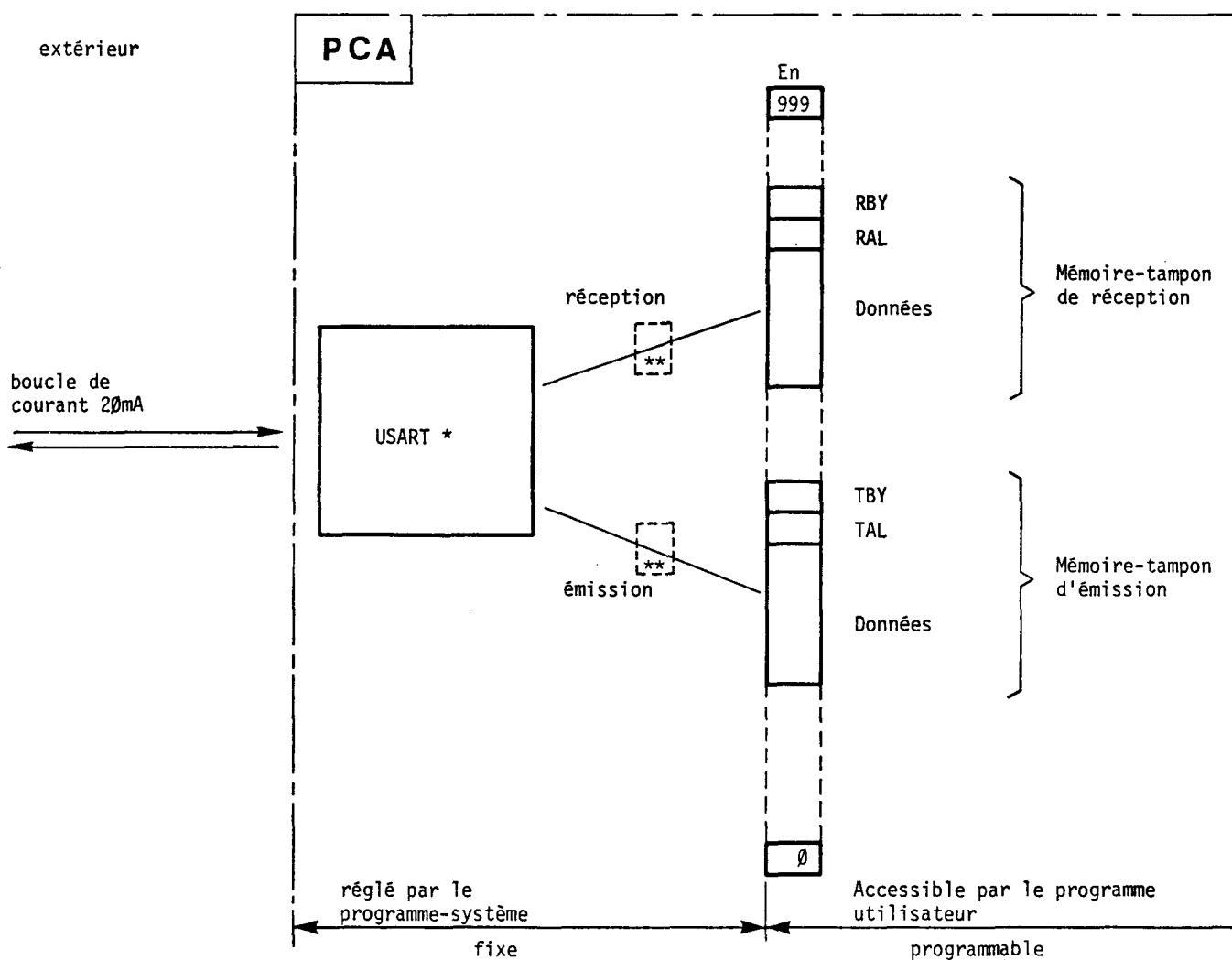
0	NOP	0 ;	
1	PAS	100 ;	} Assignment Initialisation à l'enclenchement
2	00	903 ;	
3	00	399 ;	
4	00	409 ;	
5	00	419 ;	
6	00	0 ;	
.	.	.	
10	00	0 ;	
11	WIL	409 ;	} Programme
12	SCR	256 ;	
13	24	407 ;	
14	REO	409 ;	
15	WIH	419 ;	
16	SCR	256 ;	
17	21	417 ;	
18	SEO	419 ;	
19	JMP	11 ;	

K 6.3 Définition du mode N

Il s'agit ici, dans le mode N d'un échange de données numériques sous forme de télégrammes selon la norme DIN 66019. De façon à activer la ligne série pour ce type de transfert, il y a dans le programme système un programme destiné à cette tâche. Les caractères reçus sont interprétés et chargés dans des éléments (en général indicateurs), de même pour les caractères émis, qui doivent tout d'abord être lu dans les registres, formatés et finalement envoyés. Ainsi le travail de l'utilisateur est facilité par l'emploi de registres d'indicateurs.

Comme le montre la figure ci-dessous, l'utilisateur doit définir 2 domaines d'indicateurs, l'un pour la mémoire-tampon de réception (emplacement de lecture des données), l'autre pour celle d'émission (emplacement de sortie des données).

La grosseur de la mémoire tampon d'entrée et de sortie est variable et redéfinissable pour chaque nouvelle application.

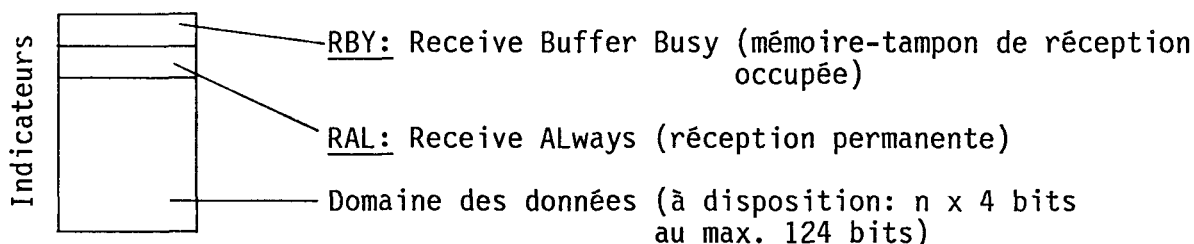


*) USART = Universal Synchronous/Asynchronous Receiver Transmitter.

**) Le télégramme est tout d'abord déposé dans une mémoire-tampon où il sera examiné. Si le contrôle est positif, le programme système procédera au transfert dans la plage d'indicateurs correspondante.

Mémoire-tampon de réception pour le mode N

Adresses supérieures



Adresses inférieures

Si aucune erreur de transmission (telle que parité, cadrage, etc.), n'est constatée à la réception, le dépôt s'effectue dans la mémoire-tampon de réception en fonction de l'état logique de RBY et RAL:

	RAL	RBY	1ère conséquence	2ème conséquence
Réception permanente	H	L	mémorisation interméd. dans la mémoire-tampon	RBY prend l'état "H"
	H	H		
Réception individuelle	L	L	mémorisation interméd. dans la mémoire-tampon	RBY prend l'état "H"
	L	H	sans effet	--

Réception permanente:

Si le programme-utilisateur définit l'état RAL = "H" et pour une réception sans erreur des données, ces dernières seront toujours déposées dans la mémoire-tampon de réception.

Réception individuelle:

Si par contre c'est l'état RAL = "L" qui est défini, et toujours pour une réception sans erreur des données, ces dernières ne seront déposées dans la mémoire-tampon de réception que pour RBY = "L". Après quoi, RBY prend automatiquement l'état "H", évitant que de nouvelles données ne viennent se superposer dans la mémoire. Le programme-utilisateur traite tout d'abord les données de la mémoire-tampon, puis positionne l'état RBY = "L" pour permettre l'entrée des données suivantes.

Structure de programme pour réception de données (RAL = L)

```

WIL  RBY          ; Attendre la réception de nouvelles données.

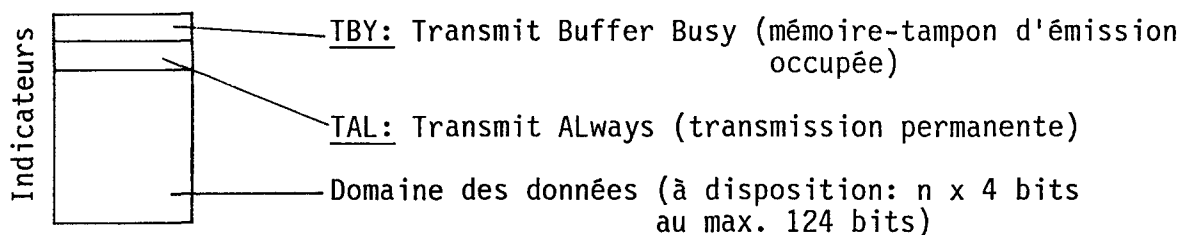
Lecture de la mémoire-tampon de réception ; Les données sont reçues par le programme utilisateur et transmises aux registres de comptage

REO  RBY          ; Mémoire-tampon de réception libérée pour recevoir de nouvelles données.
  
```

N.B.: RBY devient H à la réception du premier caractère; ainsi après le WIL, il faut être certain que les caractères suivants seront reçus et mémorisés. Le temps d'attente est dépendant de la vitesse de transmission.

Mémoire-tampon d'émission pour le mode N

Adresses supérieures



Adresses inférieures

Chaque 1/10 ou 1/100 s, selon la base de temps choisie, l'USART peut donner l'ordre d'émission de données au dépend de l'état logique des indicateurs TBY et TAL:

	TAL	TBY	1ère conséquence	2ème conséquence
Transmiss. permanente	H	L	transmission du caractère depuis la mémoire-tampon	TBY prend l'état "L"
	H	H		
Transmiss. individuelle	L	L	sans effet	--
	L	(H)	transmission du caractère depuis la mémoire-tampon	TBY prend l'état "L"

Transmission permanente:

Si le programme-utilisateur définit l'état $TAL = "H"$ les données fixées par le contenu de la mémoire-tampon sont émises à chaque 1/10 ou 1/100 s.

Transmission individuelle:

Si par contre c'est $TAL = "L"$ qui est défini, les données stockées dans la mémoire-tampon d'émission ne seront émises que si le programme-utilisateur fixe l'état $TBY = "H"$. Il s'en suivra automatiquement, sous l'effet du programme-système, le passage à "L" de TBY, pour annoncer au programme-utilisateur la fin de la transmission.

Structure de programme pour l'émission de données ($TAL = L$)

WIH TBY ; Emission de données précédentes terminée?

Introduire les données dans la mémoire-tampon d'émission ; Le programme utilisateur transmet les données à envoyer, à la mémoire tampon d'émission.

SEO TBY ; Emission du contenu de la mémoire-tampon

N.B.: TBY devient "L" à l'émission du premier caractère; ainsi après le WIH, il faut être certain que les caractères suivants seront émis. Le temps d'attente est dépendant de la vitesse de transmission.

K 6.3.1 Assignment pour le mode N

1: PAS (29) 100 ; Assignment de l'interface

2: 00 xxxx ; BAUD, parité et nombre de bits d'arrêt

3: 00 xxx ; TXB (Text-Busy Flag)

4: n* xxx ; RBY (mémoire-tampon de réception)

5: n* xxx ; TBY (mémoire-tampon d'émission)

6: Code	{ 00	d ; sans BCC (var. N2)	Deux variantes
	{ 01	d ; avec BCC (var. N1)	à choix

} Caractéristiques du mode N

7: 00 0

8: 00 0

9: 00 0

10: 00 0

} Zéro

- Ligne 3: TXB (Text-Busy Flag)

Pour xxx introduire l'adresse d'un indicateur (ou d'une sortie) libre, qui prend l'état "L" lors de la mise en service du PCA.

- Ligne 4: RBY (mémoire-tampon de réception)*

n = 01...31 doit correspondre à une mémoire-tampon de données (de 4 x n) bits (c.-à-d. 4...124 bits).

Au total, réserver 2 + 4 x n indicateurs (ou sorties) consécutives et libres, dont l'adresse d'élément la plus haute sera xxx (RBY).

- Ligne 5: TBY (mémoire-tampon d'émission)* voir RBY, ligne 4

Les mémoires-tampon de réception et d'émission peuvent être de dimensions différentes.

- Ligne 6:

La ligne 6 a plusieurs significations: Le CODE définit le fonctionnement avec ou sans BCC:

Code 01: avec BCC (le périphérique doit savoir travailler avec BCC)

Mode **Variante N1**

Code 00: sans BCC (le périphérique est un terminal non-intelligent)

Mode **Variante N2**

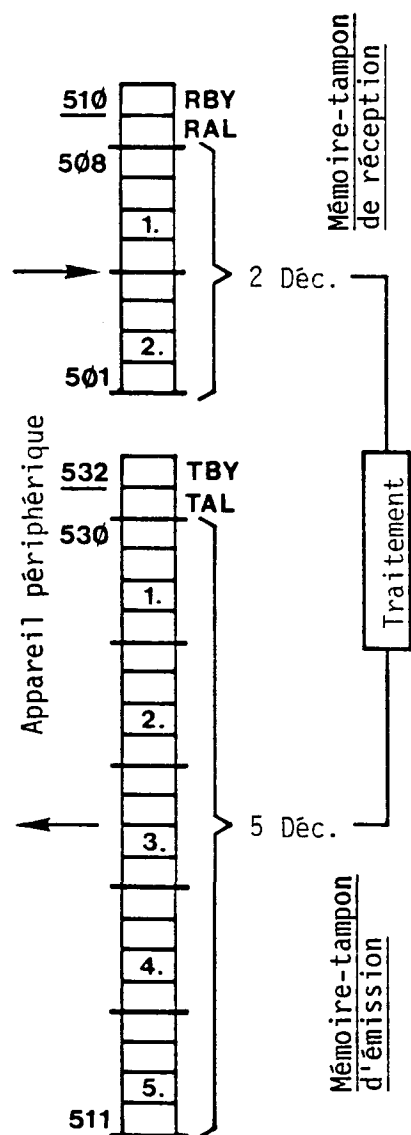
Si l'opérand d = 0, un télégramme n'ayant pas été quittancé sera répété toutes, les 30 bases de temps (3s pour bas de temps 1/10) jusqu'à ce qu'il soit quittancé.

Si l'opérand d = 1, le télégramme ne sera émis qu'une seule fois indépendamment du fait qu'il ait été quittancé ou non. Si la quittance manque, le TBY restera à l'état logique 1. Dans ce cas, il faudra prévoir une temporisation de surveillance.

Si l'opérand d = 3, le télégramme sera envoyé une seule fois, mais le PCA n'attend aucune réponse. **Variante N3**

*) Le mémoire-tampon de réception ainsi que celle d'émission sont toujours à définir, même si une seule des deux est nécessaire, p. ex. 01399, c'est-à-dire le minimum de 2 + 4 bits (394...399).

Exemple: Réception d'une valeur 2 décades et émission du résultat à 5 décades (Mode N2) après traitement. Comme l'échange d'informations est exécuté avec un "non-intelligente" appareil périphérique et son opérateur, conformément au mode N2 on se prive du BCC. Pour pouvoir émettre le télégramme, il faut précéder chaque introduction de données d'un CTRL/B (pour STX) et terminer chaque introduction d'un CTRL/C (pour ETX).



Programme (pour SILENT 743)

0	NOP	0		
1	PAS	100		
2	00	898	;	(300 BAUD)
3	00	32	;	TXB
4	02	510	;	RBY
5	05	532	;	TBY
6	00	0	;	aucun BCC
7	00	0		
8	00	0		
9	00	0		
10	00	0		
11	WIL	510	;	Réception possible?
12	SCR	256		Lecture mémoire de
13	16	508		réception
14	REO	510	;	Réception libérée!
15	WIH	532	;	Emission possible?
16	SCR	256		x 64
17	29	64		
18	SCR	256		Déposer le résultat
19	20	530		dans la mémoire
20	SCR	256		Résultat sur
21	20	63		les sorties
22	SEO	532	;	Emettre!
23	JMP	11		

Assignment

Programme

K 6.3.2 Les télégrammes du mode N

a) Caractères de commande

Dans un télégramme plusieurs caractères spéciaux sont employés, ils proviennent de la table ASCII, colonne "Control Case". En voici un aperçu:

Abré- viation	Désignation anglaise	Désignation française	Case ou No décimal	CTRL/..
STX	Start of text	Début du texte	2	B
ETX	End of text	Fin du texte	3	C
EOT ¹⁾	End of trans- mission	Fin de la transmission	4	D
ENQ ²⁾	Enquiry	Demande de la station	5	E
ACK	Acknowledge	Signal de retour positif	6	F
NAK	Negative acknowledge	Signal de retour négatif	21	U
#	Hash	Signe nu- mérique	35	

¹⁾ EOT fait que si le PCA émet, il interrompt la transmission.

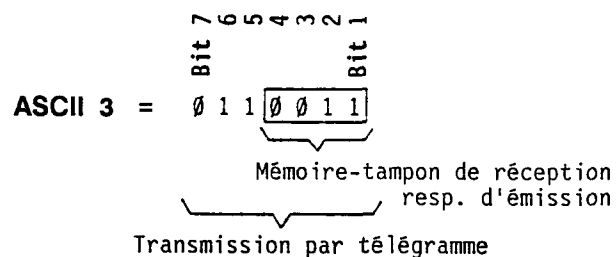
²⁾ ENQ demande une réponse au PCA (un télégramme ou NAK) ou a pour effet d'annuler un télégramme en cours de transmission vers le PCA.

b) Le télégramme envoyé

Le Mode N permet l'échange de données numériques, ce sont les caractères de la 3ème colonne de la table ASCII code décimal 48 à 63. La décade binaire 5, 6 et 7 de ces caractères est toujours la même (voir table ASCII). La mémoire tampon d'entrée et de sortie est formée de 4 bits, ainsi un groupe de 4 bits correspond à 1 caractère numérique.

A l'émission le système envoie automatiquement les 5, 6 et 7ème bit (car ils sont constants). Le programme système vérifie aussi la sommation de contrôle BCC sur tous les bits y inclus ETX. Si cette somme est correcte les données sont alors transmises à la mémoire de réception sans les bits 5, 6, 7.

Par exemple:

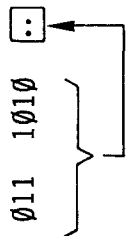


c) Que veut dire BCC?

BCC = Block Check Character (somme de contrôle)

D'un télégramme délimité par un STX au début et un ETX à la fin, le BCC est la somme transversales des 1 (binaires) des données et de l'ETX comme le montre l'exemple ci-dessous. La somme ainsi obtenue est comparée à la valeur émise, ceci afin d'augmenter considérablement la sécurité de transmission.

Télégramme:

STX	données	ETX	BCC
CTRL/B	3 6 9 = 8	CTRL/C	Somation de contrôle avec données et ETX 
B1 0010	0011 0110 1001 1101 1000	0011	
B7 0000	011 011 011 011 011	000	

d) Structure d'un télégramme

Un télégramme se compose des parties suivantes:

STX = départ du télégramme

DATA = contenu du télégramme

ETX = fin du télégramme

BCC = somme de contrôle du télégramme

Télégramme: $\xrightarrow{\text{STX DATA ETX BCC}}$

Normalement chaque télégramme doit être quittancé. Réponses:

ACK = télégramme correctement reçu

NAK = télégramme incorrectement reçu

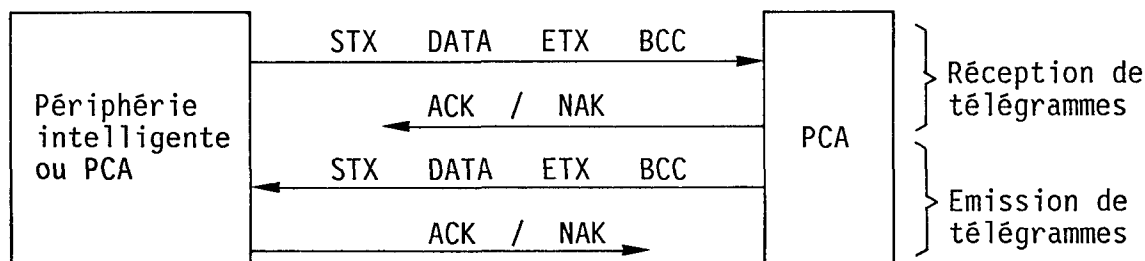
Réponse: $\xleftarrow{\text{ACK ou NAK}}$

e) Echange de télégrammes avec la PCA dans le mode N

Afin de respecter les normes DIN 66019, diverses autres variantes sont aussi possible afin de faciliter le travail.

e1) Mode N1 - selon DIN 66019 (avec BCC)

La norme fixe l'échange de données entre 2 "points". Elle suppose que les 2 systèmes en correspondance peuvent interpréter les caractères STX et ETX, de même que la somme de contrôle BCC, ainsi que les caractères de renseignement ACK, NAK, ENQ et EOT.

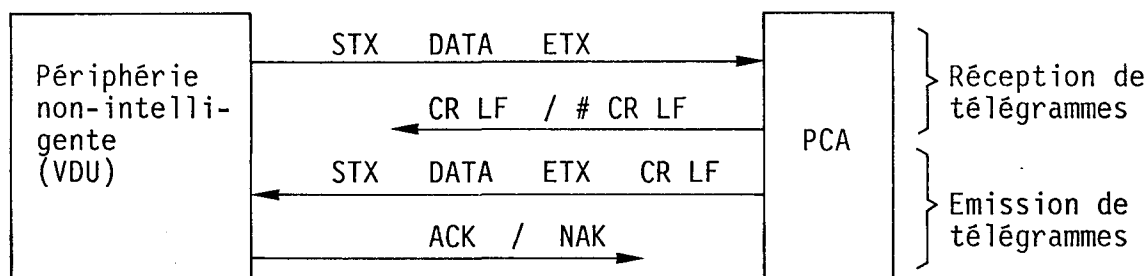


e2) Mode N2 (sans BCC)

Avec une console de correspondance et son opérateur, la formation du BCC n'est guère possible. Pour permettre cependant un échange de données avec un périphérique simple (par ex. lors de l'établissement du programme ou à la mise en service), une variante a été créée, pour laquelle l'on a renoncé au contrôle BCC et où les caractères non visibles ACK et NAK sont remplacés par:

ACK ----> par "CR LF" (carriage return avec line feed)

NAK ----> par # "CR LF".



Définition du mode de fonctionnement dans l'instruction PAS 100

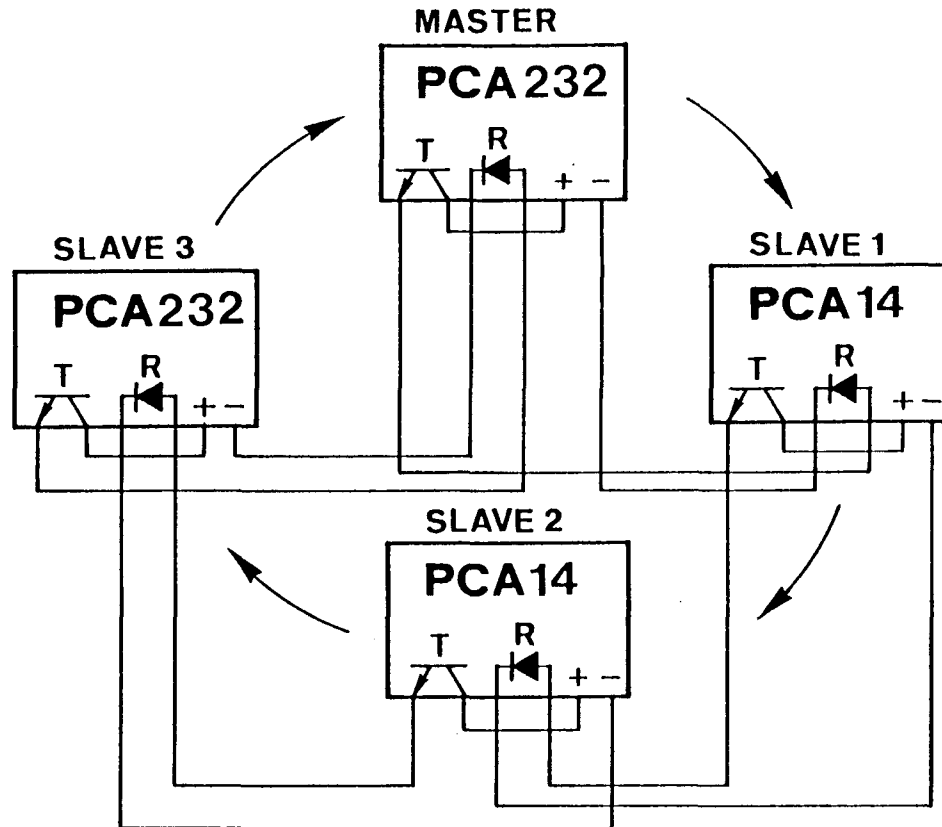
Elle est donnée par la ligne 6:

CODE = 01 ----> avec BCC, et signal en retour (non visible selon DIN 66019)
 ----> mode N1

CODE = 00 ----> pas de BCC, opération entre périphérique et PCA231
 ----> mode N2 (signal en retour visible)

e3) Mode N3

Le mode N3 permet l'échange de données entre plusieurs PCA couplés en boucle. Il faut tout de même définir un des PCA comme maître.



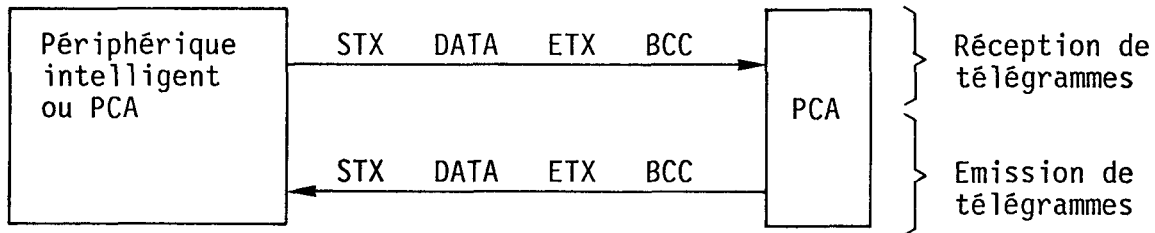
Avec les modes de transmission N1 et N2, tous les télégrammes doivent être quittancés (par ACK ou NAK). Avec le mode N3 la quittance n'est plus nécessaire. Ce qui signifie que le PCA qui a émis un télégramme, n'attend aucune réponse du système qui a reçu ce télégramme.

Le maître aura la confirmation que les informations ont été transmises à tous les PCA, lorsque dans le cas d'une boucle, le télégramme lui est retourné. Il est bien entendu que d'autres configurations, que la boucle, sont possibles.

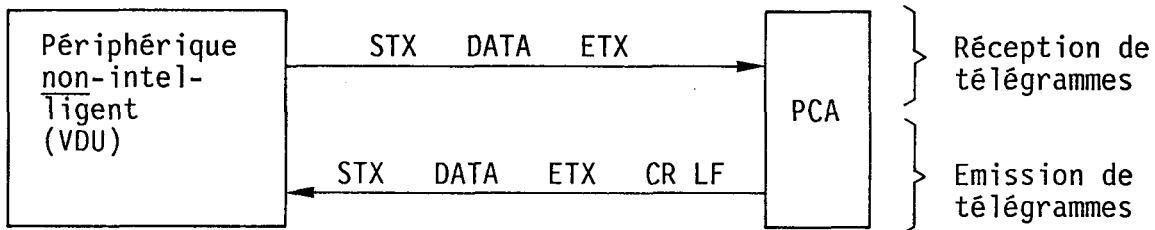
Avec la somme de contrôle BCC, une grande sécurité de transmission peut être garantie. Lorsqu'une erreur de transmission est remarquée, le télégramme entrant n'est pas mémorisé dans la mémoire tampon de réception sur indicateurs.

Le télégramme du mode N3

- Avec BCC:



- Sans BCC:



La variante N3 est définie à la ligne 6 de l'instruction PAS 100 (voir chapitre K 6.3.1)

K 6.4 Définition du Mode P

Les données du PCA sont accessibles d'une part par transfert direct de données comme vu précédemment avec les modes C et N (par l'intermédiaire de registres d'indicateur), et d'autre part avec le mode P, par un système évolué, intelligent, sans passer par un quelconque intermédiaire. Ainsi les données du PCA sont émises et reçues sans une seule mémoire tampon. Les données sont reçues et exécutées par le PCA et la quittance ACK est envoyée (NAK, si la commande ne peut pas être exécutée).

Il s'agit des données PCA suivantes:

Elements	(Adr. 0.... 999)	----> Signe E
8 éléments	(Adr. 7.... 999)	----> Signe e
Temporisateurs	(Adr. 256.... 287)	----> Signe T
5 registres de temporisation	(Adr. 260.... 287)	----> Signe t ¹⁾
Registre de comptage	(Adr. 256.... 511)	----> Signe C ²⁾
5 registres de comptage	(Adr. 260.... 511)	----> Signe c ³⁾
Registre d'opérand		----> Signe O
Adresse de pas	(Adr. 0... 8191)	----> Signe S
2 byte de mémoire texte	(Adr. 0...15999)	----> Signe M ⁴⁾
5 x 2 byte de mémoire texte	(Adr. 4...15999)	----> Signe m ⁴⁾
1 registre mot (8 bits)	(Adr. 0.... 999)	----> Signe R ⁵⁾
1 bloc de reg. (5 x 8 bits)	(Adr. 4.... 999)	----> Signe A ⁵⁾
3 blocs de reg. (15 x 8 bits)	(Adr. 14.... 999)	----> Signe B ⁵⁾
10 bits mémoire de données	(Adr. 4...15999)	----> Signe N ⁶⁾
30 bits mémoire de données	(Adr. 14...15999)	----> Signe n ⁶⁾

Pour l'écriture -----> Signe (W) (write)
 Pour la lecture -----> Signe (D) (display)

- 1) PCA02 dès la version V6.130
 PCA14 dès la version V6.031
 PCA222 dès la version V6.230
 PCA232 dès la version V7.131
- 2) PCA02 : Adr. 256...319 (dès la version V6.132 Adr. 256...479)
 PCA14 : Adr. 256...319 (dès la version V6.034 Adr. 256...479)
 PCA222: Adr. 256...319 (dès la version V6.230 Adr. 256...479)
 PCA232: Adr. 256...511
- 3) PCA02 dès la vers. V6.130 (Adr. 260...319, dès vers. V6.132 Adr. 260...479)
 PCA14 dès la vers. V6.031 (Adr. 260...319, dès vers. V6.034 Adr. 260...479)
 PCA222 dès la version V6.230 (Adr. 260...479)
 PCA232 dès la version V7.131 (Adr. 260...511)
- 4) PCA02 dès la version V6.130 (la plus haute adresse 4095)
 PCA14 dès la version V6.031 (la plus haute adresse 8191)
 PCA222 dès la version V6.230 (la plus haute adresse 8191)
 PCA232 dès la version V7.131 (la plus haute adresse 15999)
- 5) Seulement pour PCA231, 232.
- 6) Les télégrammes N et n existent que sur le PCA232 à partir de la version V7.131. Ces 2 télégrammes sont utilisables tout spécialement quand une valeur vient d'être transférée avec l'instruction PAS 57 du registre de mot vers la mémoire de donnée.

K 6.4.1 Assignment pour le mode P

a) L'instruction PAS 100

L'instruction PAS 100 dans le mode P est identique à l'instruction dans le mode N (voir chapitre K 6.3.1). Le télégramme est cependant différent.

1:	PAS (29)	100	; Assignment de l'interface
2:	00	xxxx	; Baud, parité et bits d'arrêt
3:	00	xxx	; Elément, qui prend "L"
4:	01	xxx	; Adresse la plus haute des 6 indicateurs
5:	01	xxx	; Adresse la plus haute des 6 indicateurs supplémentaires
6:**	{	00	0 ; Sans BCC (Var. P2)
		01	0 ; Avec BCC (Var. P1)
7:	00	0	} Zéro
8:	00	0	
9:	00	0	
10:	00	0	

- Ligne 3: TxB (Text-Busy Flag)
xxx est l'adresse d'un indicateur libre (ou E/S) cet élément doit être "L" à l'enclenchement du PCA.
- Ligne 4: RBY (Mémoire-tampon de réception)
RBY est uniquement utilisé dans le mode P, ainsi il faut réserver 2x6 registres d'indicateurs. Dans le mode computer (mode N + P), les lignes 4 et 5 sont assignées de la même façon que dans le mode N.
- Ligne 5: TBY (Mémoire-tampon de réception)
Voir ligne 4 (idem)
- Ligne 6: BCC (Code définissant le BCC)
Code = 00 ---> pas de BCC, utilisation avec un périphérique à clavier (terminal non-intelligent et réponse visible sur l'écran)
---> mode P2
Code = 01 ---> avec BCC, utilisation avec un terminal intelligent (réponse invisible) ---> mode P1

K 6.4.2 Les télégrammes du mode P

b) Caractères de commande

Les paragraphes suivants utilisent divers caractères de commande des cases "Control" du tableau ASCII, dont le résumé ci-dessous donne un aperçu:

Abré- viation	Désignation anglaise	Désignation française	Case ou No décimal	CTRL/..
STX	Start of Text	Début du texte	2	B
ETX	End of Text	Fin du texte	3	C
EOT ¹⁾	End of Trans- mission	Fin de la transmission	4	D
ENQ ²⁾	Enquiry	Demande de la station	5	E
ACK	Acknowledge	Signal de retour positif	6	F
NAK	Negative Acknowledge	Signal de retour négatif	21	U
#	Hash	Signe numérique	35	-

¹⁾ EOT fait que si le PCA emmet, il interrompt la transmission.

²⁾ ENQ demande une réponse au PCA (un télégramme ou NAK) ou a pour effet d'annuler un télégramme en cours de transmission vers le PCA.

b) Que veut dire BCC?

BCC = Block Check Character (somme de contrôle)

D'un télégramme délimité par un STX au début et un ETX à la fin, le BCC est la somme transversales des 1 (binaires) des données et de l'ETX comme le montre l'exemple ci-dessous. La somme ainsi obtenue est comparée à la valeur émise, ceci afin d'augmenter considérablement la sécurité de transmission.

Télégramme:

STX	Données						ETX	BCC
CTRL/B	Ⓜ	ⓔ	0	3	2	1	CTRL/C	CTRL/Q
B1 0010	0111	0101	0000	0011	0010	0001	0011	Somme de contrôle avec données et ETX 001 0001
B7 0000	101	100	011	011	011	011	000	

c) Structure d'un télégramme

Un télégramme se compose des parties suivantes:

STX = Départ du télégramme

DATA = Contenu du télégramme

ETX = Fin du télégramme

BCC = Somme de contrôle du télégramme

Télégramme: $\xrightarrow{\text{STX DATA ETX BCC}}$

Normalement chaque télégramme doit être quittancé. Réponses:

ACK = Télégramme correctement reçu

NAK = Télégramme incorrectement reçu

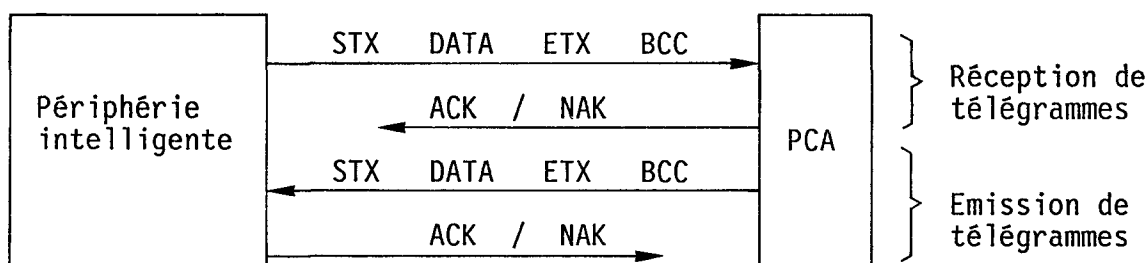
Réponse: $\xleftarrow{\text{ACK bzw. NAK}}$

d) Echange de télégrammes avec le PCA dans le mode P

Afin de respecter les normes DIN 66019, diverses autres variantes sont aussi possible afin de faciliter le travail.

d1) Mode P1 - selon DIN 66019

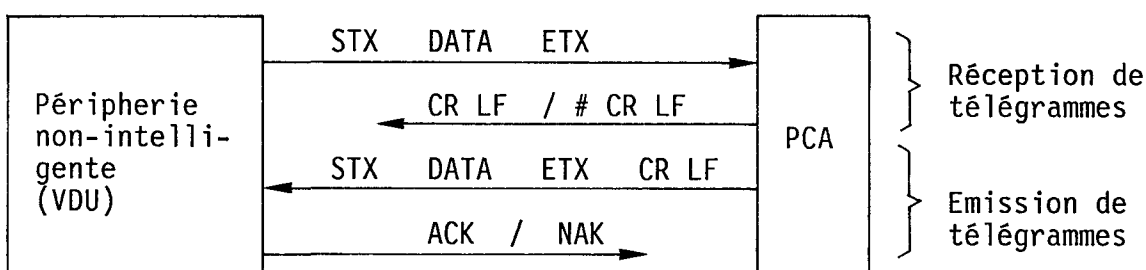
La norme fixe l'échange de données entre 2 "points". Elle suppose que les 2 systèmes en correspondance peuvent interpréter les caractères STX et ETX, de même que la somme de contrôle BCC, ainsi que les caractères de renseignement ACK, NAK, ENQ et EOT.

d2) Mode P2

Avec une console de correspondance et son opérateur, la formation du BCC n'est guère possible. Pour permettre cependant un échange de données avec un périphérique simple (par ex. lors de l'établissement du programme ou à la mise en service), une variante a été créée, pour laquelle l'on a renoncé au contrôle BCC et où les caractères non visibles ACK et NAK sont remplacés par:

ACK ----> par "CR LF" (carriage return avec line feed)

NAK ----> par # "CR LF".

Définition du mode de fonctionnement dans l'instruction PAS 100

Elle est donnée par la ligne 6:

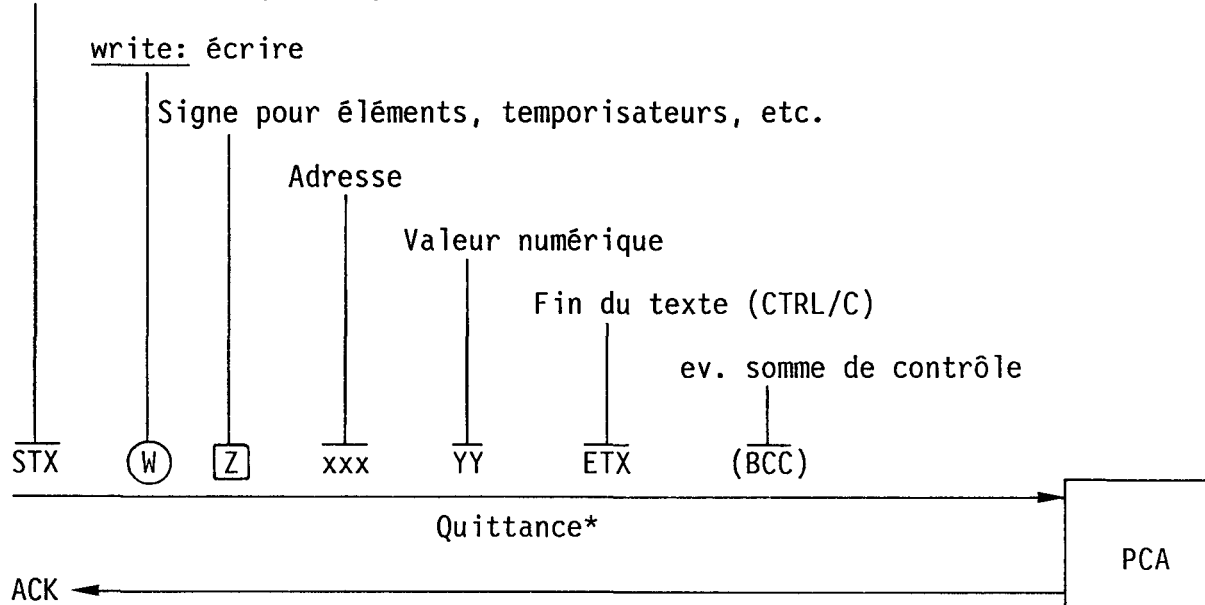
CODE = 01 ----> avec BCC, et signal en retour
 ----> Mode P1 (réponse non-visible selon DIN 66019)

CODE = 00 ----> pas de BCC, opération entre périphérique et PCA231
 ----> Mode P2 (signal en retour visible)

K 6.4.3 Ecriture de données dans le PCA

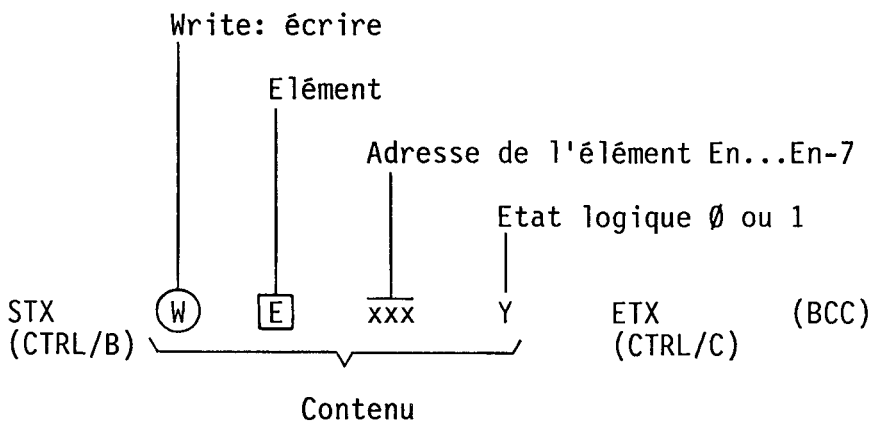
Le télégramme à envoyer au CPU se présente généralement sous la forme suivante:

Début du texte (CTRL/B)



Réponse positive du CPU, indiquant que l'ordre d'entrée a pu être exécuté (également CR LF). NAK (ou # CR LF), dans le cas où l'ordre n'a pas pu être exécuté.

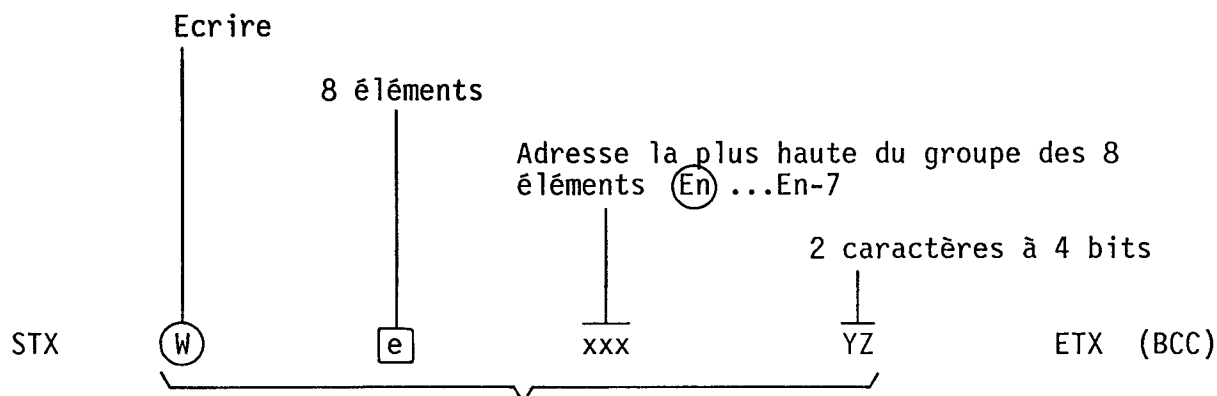
- Positionner les éléments En (E, S, T, C, I) dans l'état logique (0 ou 1)



---> mettre l'élément xxx (3 chiffres) dans l'état logique Y (1 chiffre 0 ou 1)

*) Exécuté immédiatement (sans attendre la prochaine base de temps).

- Positionnement de 8 éléments consécutifs (S, M) avec leurs états logiques correspondant (1 ou 0) par l'intermédiaire de deux caractères à 4 bits



---> Positionnement du groupe d'éléments dont xxx est l'adresse la plus haute, avec les états logiques définis dans les caractères YZ.

(Y positionne les éléments En-4...En-7; Z positionne les éléments En...En-3)

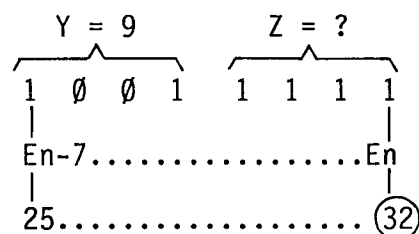
Les caractères pour Y et Z peuvent être donnés comme suit:

Caractères		Etat logique des éléments			
		Y	En-7	En-6	En-5
Z		En-3	En-2	En-1	En
Ø		Ø	Ø	Ø	Ø
1		Ø	Ø	Ø	1
2		Ø	Ø	1	Ø
3		Ø	Ø	1	1
4		Ø	1	Ø	Ø
5		Ø	1	Ø	1
6		Ø	1	1	Ø
7		Ø	1	1	1
8		1	Ø	Ø	Ø
9		1	Ø	Ø	1
:		1	Ø	1	Ø
;		1	Ø	1	1
<		1	1	Ø	Ø
=		1	1	Ø	1
>		1	1	1	Ø
?		1	1	1	1

Exemple:

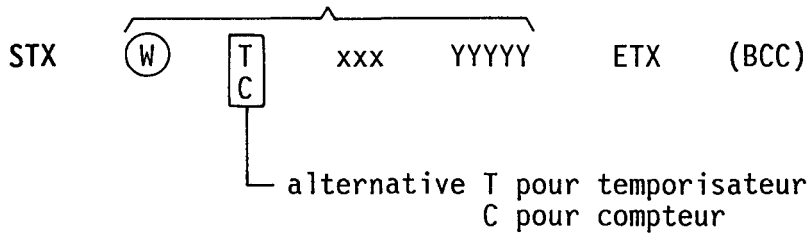
STX W e (32) 9? ETX

correspond au positionnement des sorties 25... (32) comme suit:



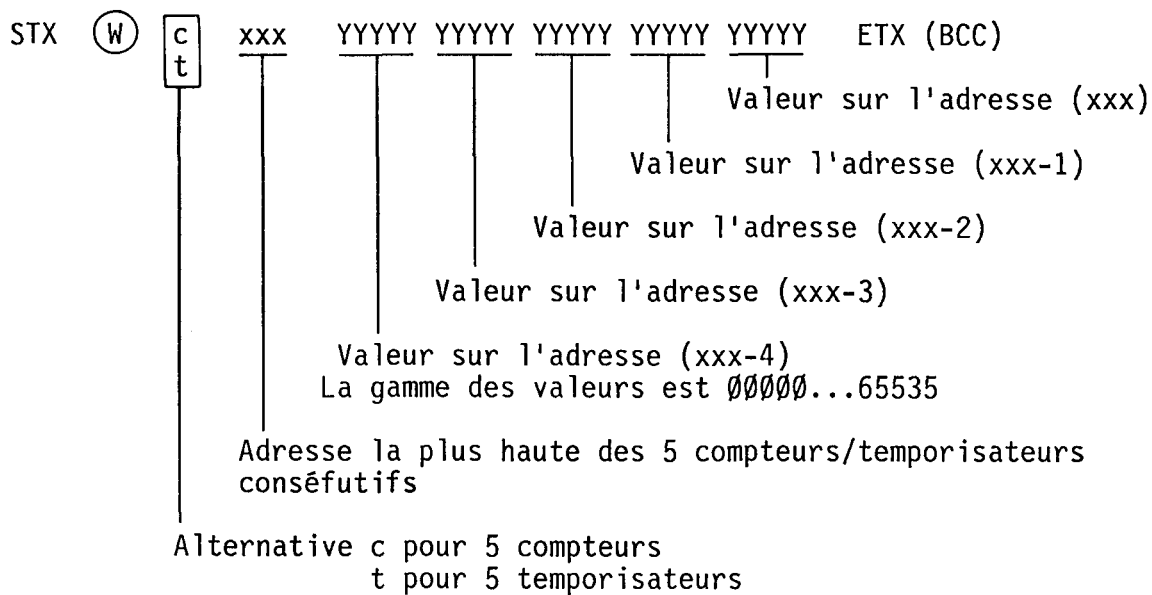
Par rapport au positionnement des états logiques de 8 éléments séparés, avec cette variante, les temps de positionnement est raccourci d'un facteur 7.

- Ecrire une valeur dans le registre de temporisateurs ou compteurs (Cn)

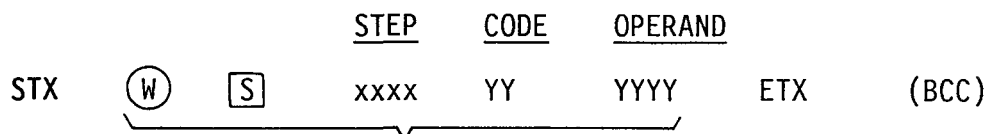


---> Ecrire la valeur YYYYY (5 chiffres) dans le registre de temporisateurs ou compteurs à l'adresse xxx (3 chiffres)

- Ecrire 5 valeurs dans une suite de 5 compteurs ou temporisateurs



- Ecrire une ligne de programme (STEP) dans la mémoire utilisateur RAM



---> Ecrire à l'adresse (STEP) xxxx de la mémoire RAM utilisateur une valeur YYYYYY (6 chiffres, 2 pour le code 0...31 et 4 pour l'opérand 0...2047)

- Ecrire une valeur dans le registre op rand pour afficher

Les instructions du processeur de bits DOP et DTC permettent d'inscrire dans le registre d'op rand des nombres de 4 chiffres qui peuvent  tre affich s dans le champ op rand du clavier de programmation. Le contenu du registre est affich  dans la seconde qui suit pour une dur e d'une seconde.

STX (W) (O) YYYY ETX (BCC)

└──────────┘

---> Ecrire la valeur YYYY (4 chiffres) dans le registre op rand.

- Ecrire une valeur   2 chiffres (1 mot de registre) dans le registre de mot Rn *

STX (W) (R) xxx YY ETX (BCC)

└──────────┘

---> Ecrire la valeur YY (2 chiffres) dans le registre de mot avec l'adresse xxx (3 chiffres).

- Ecrire une valeur de 1  chiffres (1 bloc de registre) dans le registre de mot (5 mots de registre) *

STX (W) (A) xxx YYYYYYYYYY ETX (BCC)

└──────────┘

---> Ecrire la valeur Y...Y (1  chiffres) dans un bloc de registres, xxx correspondant   la plus haute adress du bloc de registre.

- Ecrire une valeur 3 x 1  chiffres (3 blocs de registres) dans le registre de mot (15 mots de registre) *

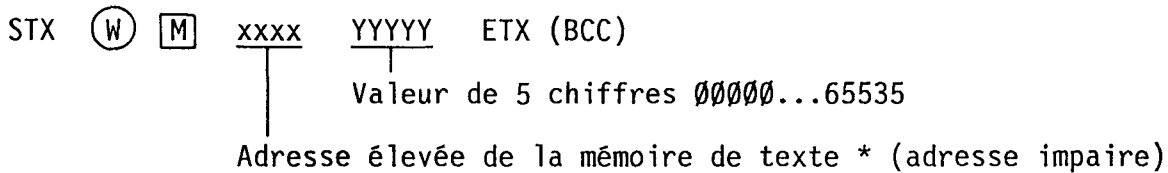
STX (W) (B) xxx Y...(3 x)...Y ETX (BCC)

└──────────┘

---> Ecrire la valeur Y...Y (3  chiffres) dans le registre de mot, xxx correspondant   la plus haute adresse de 3 blocs de registres.

*) Seulement PCA231/232.

- Ecrire une valeur de 2 octets sur la mémoire de texte

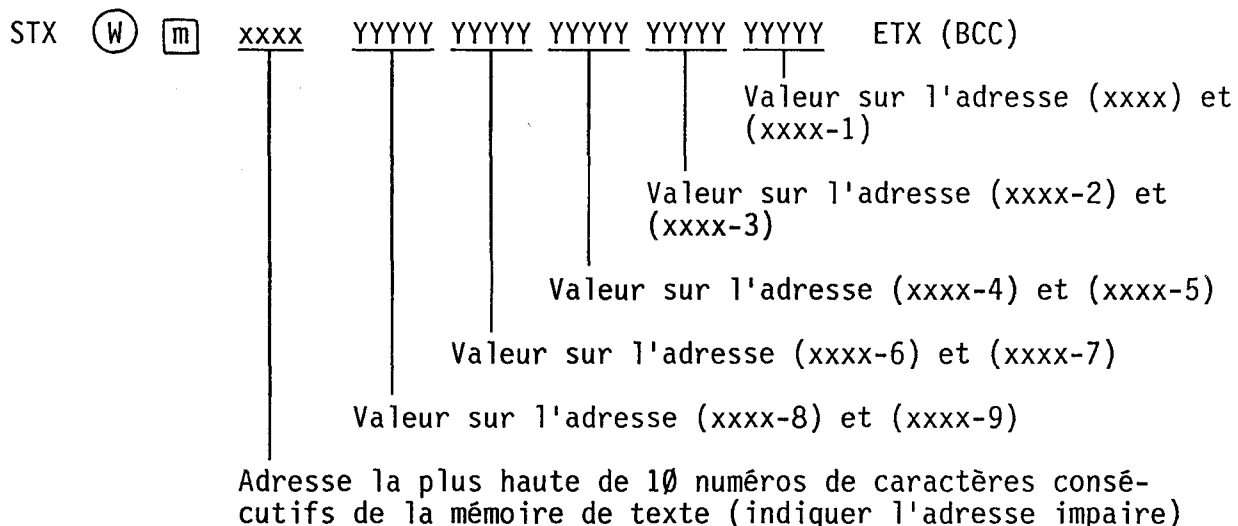


La valeur YYYYY est mémorisée dans la mémoire de texte en forme binaire. A l'adresse élevée (xxxx) se trouve l'octet supérieur, a l'adresse basse (xxxx-1) se trouve l'octet inférieur.

*) Le PCA2.M32 dispose d'une mémoire de texte de 8K caractères et d'une mémoire de données de 8K octets qui peut aussi être utilisée comme mémoire de texte. L'adresse indiquée se réfère à la mémoire de texte, par conséquent la gamme des adresses est de 0 à 15999, tenant compte que pour les 10 - 15 milliers il faut les signes suivants:

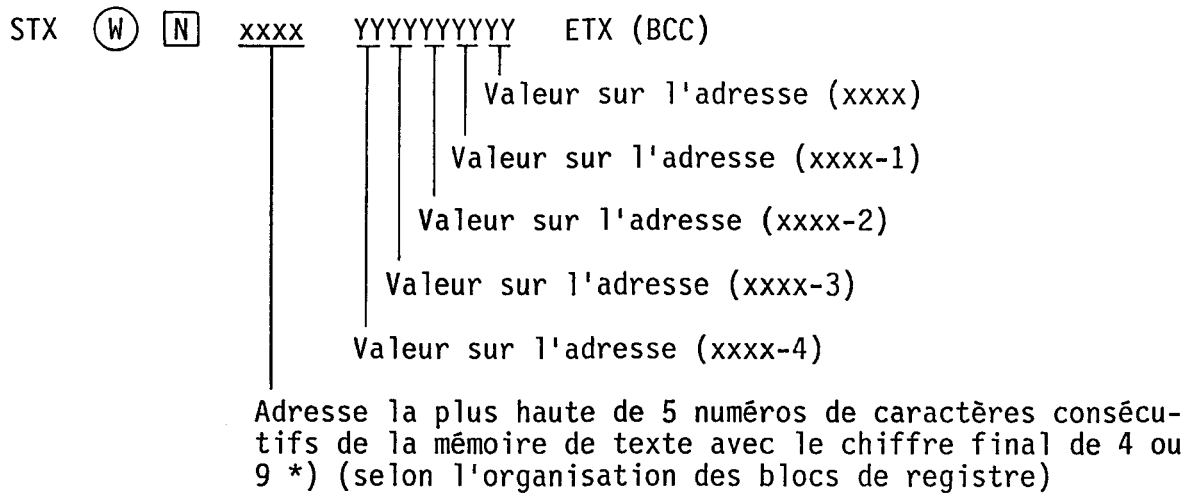
10 : / 11 ; / 12 < / 13 = / 14 > / 15 ?

- Ecriture des valeurs de 5 x 2 octets sur la mémoire de texte



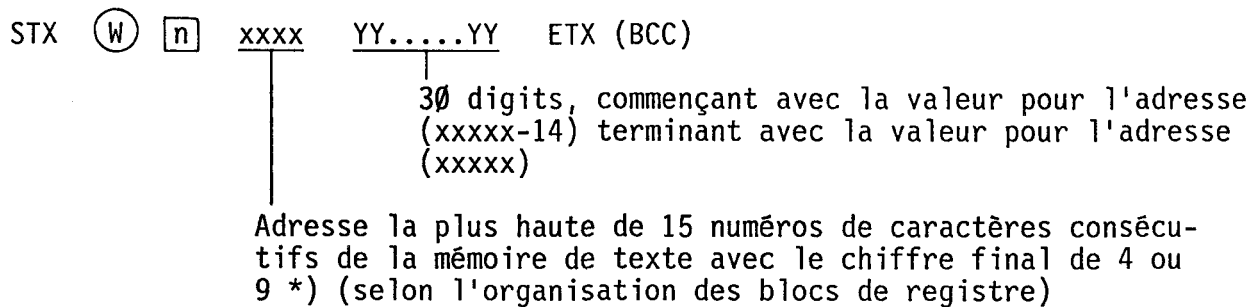
Les valeurs YYYYY sont mémorisées de la même manière que le télégramme (M). L'information mentionnée sous la note *) a toujours sa validité.

- Ecriture de 10 digits (1 bloc de registre) sur la mémoire de texte



*) voir note *) sous télégramme (M)

- Ecriture de 30 digits (3 blocs de registres) sur la mémoire de texte



*) Voir note *) sous télégramme (M)

Seulement certains télégrammes fonctionnent sur tous les CPU de la famille PCA. A ce propos, consultez la vue d'ensemble "Définition du mode P", au chapitre K 6.4.

P.S.: Avant que le dernier caractère d'un télégramme soit transmis, le télégramme peut être désigné comme non-valable par le code ENQ (CTRL/E). Le PCA donne ensuite une quittance avec NAK (# CR LF). Ensuite un nouveau télégramme sera envoyé.

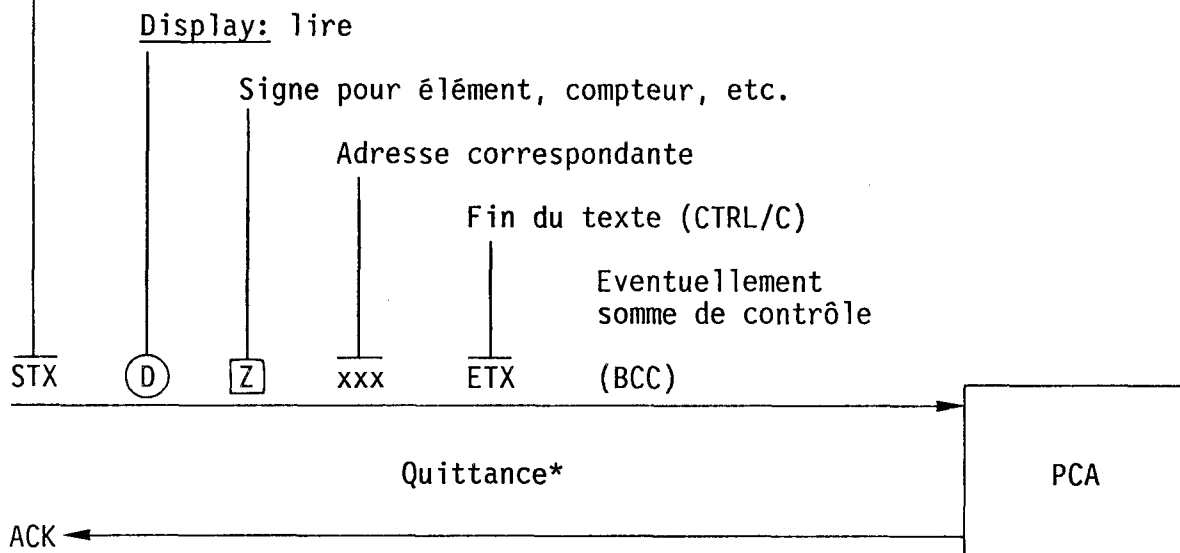
K 6.4.4 Lecture de données du PCA

Ce genre d'interrogation permet de lire du PCA des valeurs telles que: état logique d'éléments En, contenu de compteurs ou temporisateurs, opérandes à afficher, ou encore contenu de pas du programme utilisateur.

L'interrogation se déroule en 2 pas:

① Ordre de mémorisation dans le CPU M32

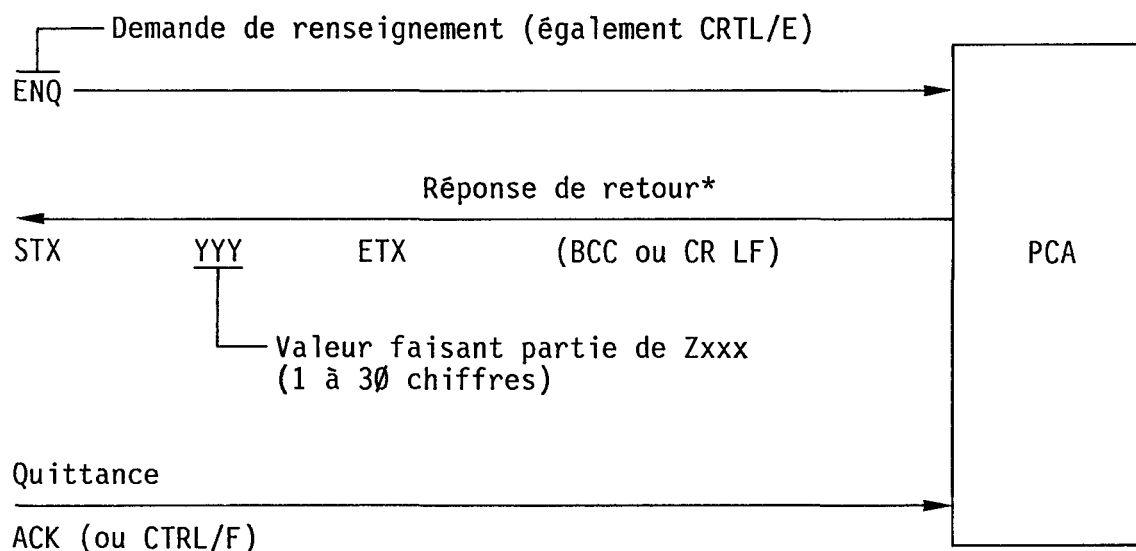
Début de texte (CTRL/B)



Réponse positive du CPU, attestant l'acceptation de l'ordre (ou CR LF).
NAK (ou # CR LF) signifie que la mémorisation n'a pas pu se faire.

*) Exécuté immédiatement (sans attendre la prochaine base de temps).

- ② Ordre d'émission de la valeur (YY..) appartenant à l'élément précédemment introduit (Zxxx).



Ce qui termine l'échange de données. Si le signal ACK manque (que l'opérand de la ligne 6 du PAS 100 contient 0000), le télégramme sera répété toutes les 30 bases de temps. Dans le cas où le CPU reçoit NAK (ou CTRL/U), le télégramme sera répété immédiatement.

P.S.: Si aucune adresse n'a été préalablement mémorisée (pas 1), le PCA répond à ENQ par NAK (ou # CR LF).

*) Exécuté immédiatement (sans attendre la prochaine base de temps).

Les diverses données du PC sont consultées de la manière suivante:

- Lire l'état logique d'éléments En avec adresse xxx

En = (E, A, T, C, M)

STX (D) (E) xxx ETX (BCC) , xxx = Adresse d'éléments
à 3 chiffres

- Lecture de 8 éléments consécutifs (E, S, M) sous la forme de
2 caractères à 4 bits

STX (D) (e) xxx ETX (BCC) , xxx = Adresse d'éléments
la plus haute (3
chiffres) d'un groupe
d'éléments

Le PCA répond avec

STX YZ ETX (BCC ou CR LF)

Exemple pour YZ

Caractère	Y	Z
Configuration binaire avec En = 32	0 0 1 1 25	1 1 1 0 32
Caractère général	3	>

- Lire le contenu de registre de temporisation ou comptage (Cn)

STX (D) (T
C) xxx ETX (BCC) , xxx = Adresse temporisateur/
compteur à 3 chiffres

- Lecture de 5 compteurs/temporisateurs consécutifs

STX (D) (t
c) xxx ETX (BCC) ; xxx = Adresse la plus haute de 5 temporisa-
teurs/compteurs consécutifs

- Lire le contenu d'une ligne de programme (STEP)

STX (D) (S) xxxx ETX (BCC) , xxxx = Adresse de pas
4 chiffres (max.
8191)

- Lire le contenu du registre d'opérand à afficher (pour DOP et DTC)

STX (D) (O) ETX (BCC)

- Lire le contenu du mot de registre (Rn) de 2 digits (1 mot de registre)

STX (D) [R] xxx ETX (BCC) , xxx = Adresse de registre à 3 chiffres

- Lire le bloc de registre (Rn) de 10 digits (5 mots de registre)

STX (D) [A] xxx ETX (BCC) , xxx = Adresse de registre la plus haute du bloc de registre

- Lire les 3 blocs de registres de 30 digits (15 mots de registre)

STX (D) [B] xxx ETX (BCC) ; xxx = Adresse de registre la plus haute de 3 blocs de registre

- Lecture d'une valeur de 2 octets de la mémoire de texte

STX (D) [M] xxxx ETX (BCC) ; xxxx = l'adresse élevée de la mémoire de texte (indiquer l'adresse impaire)

- Lecture des valeurs de 5 x 2 octets de la mémoire de texte

STX (D) [m] xxxx ETX (BCC) ; xxxx = l'adresse la plus haute de 10 numéros de caractères consécutifs (indiquer l'adresse impaire)

- Lecture de 10 digits (1 bloc de registres) de la mémoire de texte

STX (D) [N] xxxx ETX (BCC) ; xxxx = l'adresse la plus haute de 5 numéros de caractères de la mémoire de texte avec le chiffre final 4 ou 9 (selon l'organisation des blocs de registres)

- Lecture de 30 digits (3 blocs de registres) de la mémoire de texte

STX (D) [n] xxxx ETX (BCC) ; xxxx = l'adresse la plus haute de 15 numéros de caractères de la mémoire de texte avec le chiffre final 4 ou 9 (selon l'organisation des blocs de registres)

Remarque

Les données demandées par le télégramme de lecture seront transmises dans le télégramme de réponse selon le même format que les données dans le télégramme d'écriture.

Seulement certains télégrammes fonctionnent sur tous les CPU de la famille PCA. A ce propos, consultez la vue d'ensemble au chapitre K 6.4 "Définition du mode P".

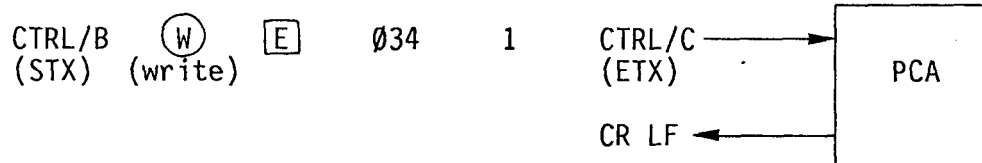
P.S.: Avant la transmission du dernier caractère d'un télégramme, celui-ci peut être déclaré non valable par ENQ (CTRL/E). Le PCA quitte alors par NAK (# CR LF). Ensuite, il est possible de transmettre un nouveau télégramme.

Exemples:

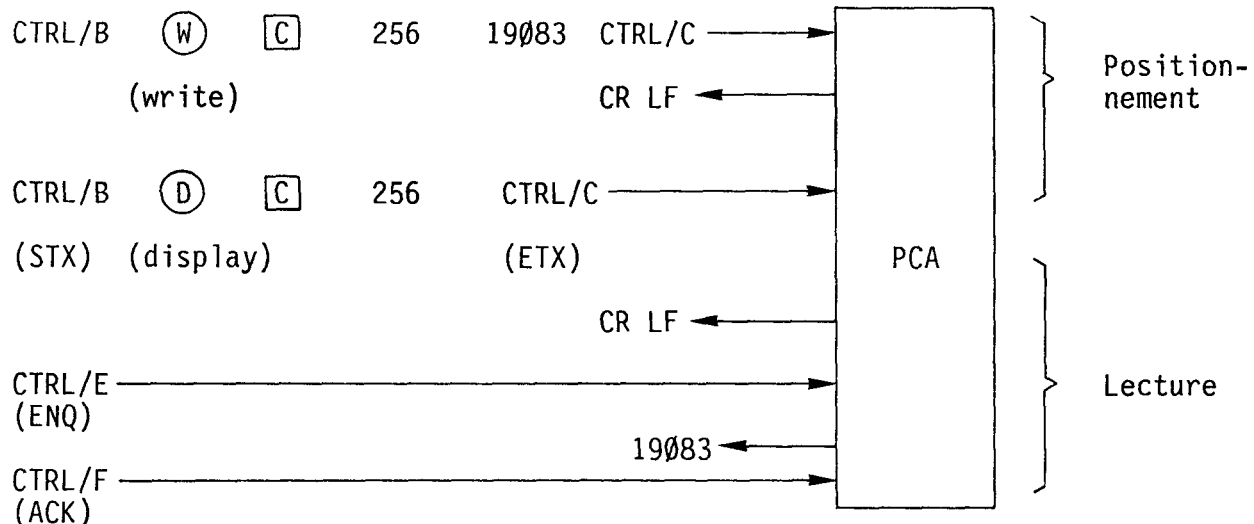
Les deux exemples ci-dessous ne fonctionnent que si l'assignation de la sortie série a été faite correctement c'est-à-dire à la ligne 6, le code 00 ainsi le mode P2 est sélectionné. Les exemples peuvent alors être exécutés avec un terminal simple.

a) Positionnement d'une sortie en mode P

Adresse de sortie: 34



b) Positionnement d'un compteur et lecture de ce compteur en mode P



K 6.5 Assignment pour les modes combinés

Une combinaison de plusieurs mode de communication pour l'émission et la réception est possible dans les cas suivants avec une seule assignation:

Mode Menu (Réception de caractères uniques et envoi de texte):

Assignation pour Mode C

Mode Computer (Echange de données numériques et de données orientées PCA):

Assignation pour Mode N

Assignation les choix possibles:

Assignation	Opération possible dans
pour sortie de texte	Mode T
pour caractères uniques	Modes C + T
pour données numériques	Modes N + T + P
pour données orientées PCA	Modes P + T

K 6.6 Vue générale du PAS 100 et des variantes de fonctionnement

Commentaires concernant le <u>CODE</u> :	Step No	Code	Opérand	Commentaire concernant les <u>opérands</u> :
	1:	PAS	100	Assignment de la ligne série
	2:	00	xxxx	BAUD, bits/caractère, parité, bit d'arrêt
E — Mode éditeur de texte Autres modes de service	3:	01 00	TXB	Text busy flag
N — Réception de télégramme avec 1...31 x 4 bits de données C — Réception de caractères uniques	4:	01...31 00	RBY	Receive buffer busy flag
— Emission de télégramme avec 1...31 x 4 bits de données — Emission de caractères uniques	5:	01...31 00	TBY	Transmit buffer busy flag
N1, P1, N3 — Avec BCC pour "périphérie intelligente"	6:	01	0	Sans quittance, le télégramme sera répété
N2, P2, N3 — Sans BCC pour périphérie conventionnelle		00	1 3	Le télégramme ne sera émis qu'une seule fois Le télégramme ne sera émis qu'une seule fois Aucune réponse attendue
	7...10:	00	0	Vide

Les modes N et P se différencient seulement par le contenu du télégramme:

N S | Données numériques E (B
 T | T (C
 X | X (C

N1 A N
 C A
 K K

N2 C | L # C | L
 R | F R | F

N3 pas de réponse

————— Télégramme —————→

←————— Réponse —————

P S | E,e,T,t,C,c,S,O, E (B
 T | R,A,B,M,m,N,n T (C
 X | X (C

P1 A N
 C A
 K K

P2 C | L # C | L
 R | F R | F

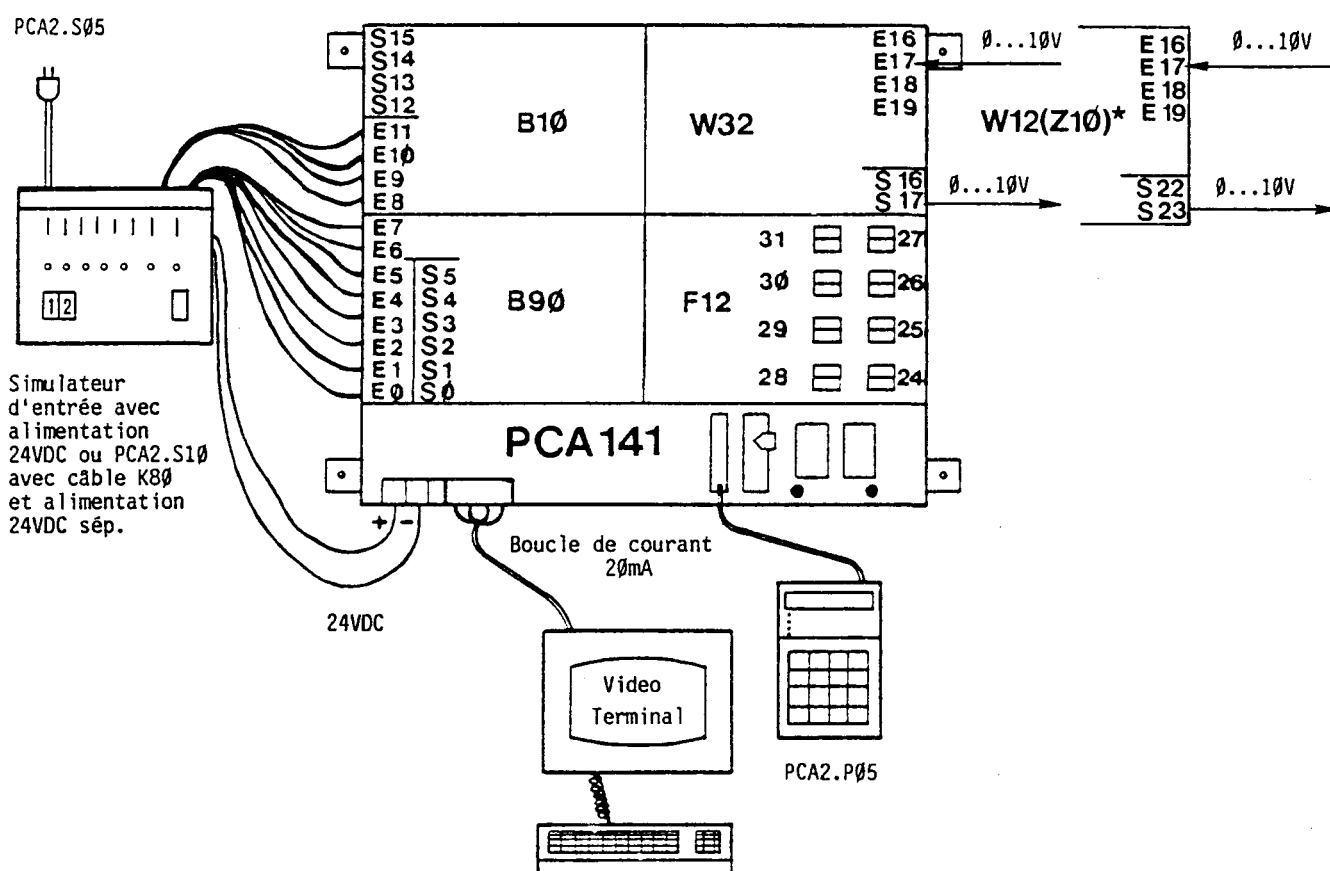
PARTIE L EXEMPLES DE PROGRAMMATION

- Exemple 1 Fonction d'enclenchement et sortie de texte à une heure prédéterminée (PAS 50)**
- Exemple 2 "Check-Sum" du programme-système et programme utilisateur avec activation du Watchdog (PAS 30...38)**
- Exemple 3 Signes spéciaux pour textes**
- Exemple 4 Jeu de balle**
- Exemple 5 Echange de données entre deux PCA14 dans le mode N par l'interface série**
- Exemple 6 Boucle de réglage PID rapide avec PAS 202 et PCA14 (Moteur-tachymètre)**
- Exemple 7 Réglage manuel/PID avec PAS 211 et PCA14 (Module analogique 12 bits PCA1.W32)**
- Exemple 8 Registre à décalage rotatif avant/arrière avec remise à zéro (Utilisation du PAS 250)**
- Exemple 9 Registre à décalage BCD avant et arrière (Utilisation du PAS 250)**
- Exemple 10 Registre FIFO (Utilisation du PAS 251)**
- Exemple 11 Signalisations de dérangements (Mémorisation dans le registre FIFO avec PAS 251)**

PARTIE L EXEMPLES DE PROGRAMMATION

Dans les exemples suivants, 1 à 11 nous aborderons les multiples aspects de la communication série à l'aide d'exemples concrets dans les domaines suivants: Sortie de texte, régulation PID, registre à décalage, registres First-in First-out. Quelques autres exemples expliquent aussi l'utilisation de l'horloge interne et de la sommation de contrôle.

Pour faciliter la compréhension des exemples nous avons choisi une configuration de base identiques pour tous les exemples suivants. C'est le plus répandu de nos automates programmables, le PCA14 dans la configuration suivante:



Tous les exemples peuvent aussi fonctionner avec un PCA2 pour autant que l'on dispose d'un module digital d'entrée et un module de sortie digital positionnés à l'adresse 0. Nous recommandons le module analogique PCA2.W14 placé à l'adresse 176 (par exemple).

*) Le module PCA1.W12 Z10 permet une entrée de tension de 0...10V (au lieu de 0...5V).

Exemple 1 Fonction d'enclenchement et sortie de texte à une heure prédéterminée (Utilisation de l'instruction PAS 50)

Problème 1a

En mode de fonctionnement MAN, l'horodateur doit être positionné. Par les 6 décades des adresses S24...S26, l'heure à laquelle les sorties S0...S5 doivent briller pendant 4s*, doit être fixée. Les heures et minutes de l'horodateur doivent être affichées sur l'afficheur d'opérand.

Affichage d'opérand

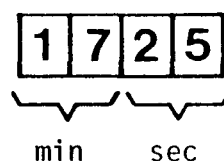
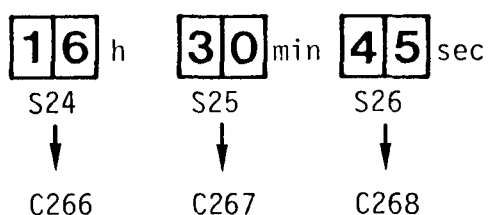
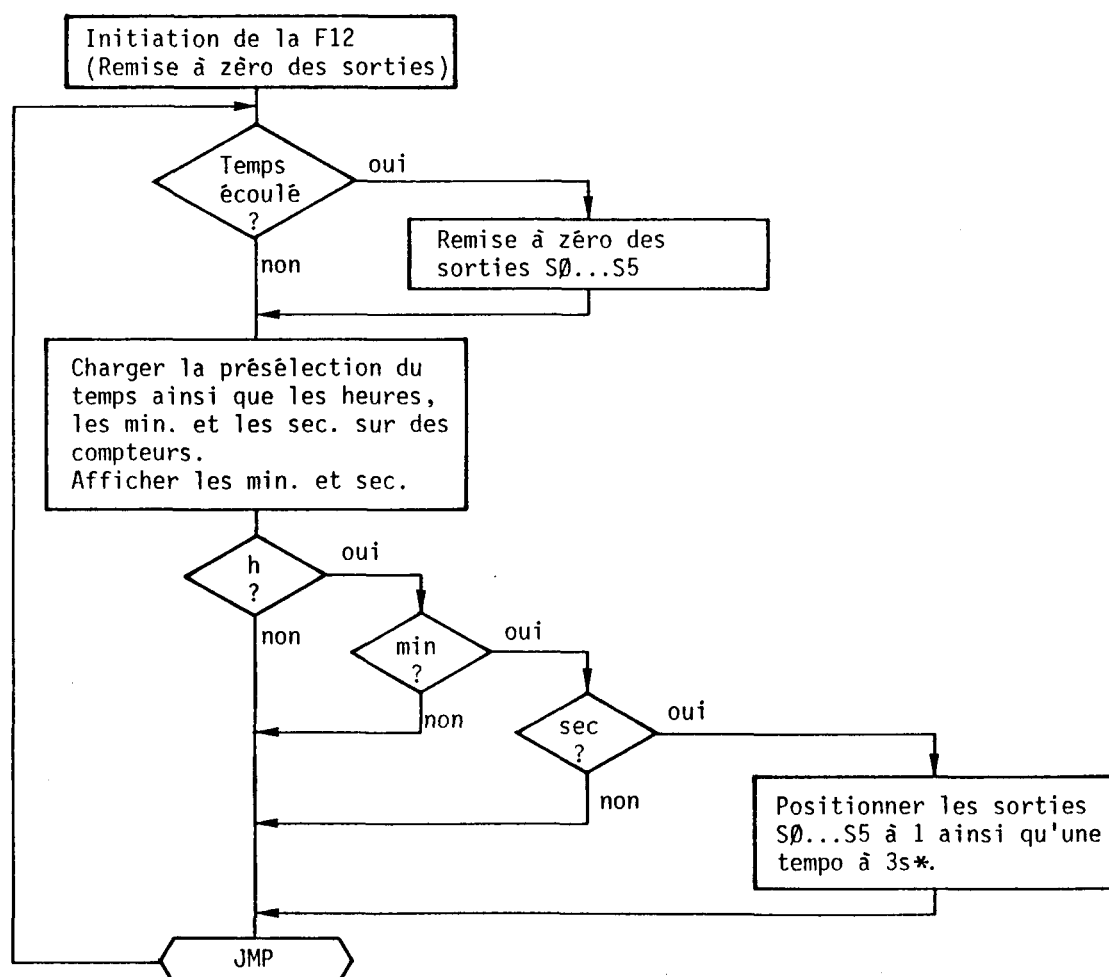
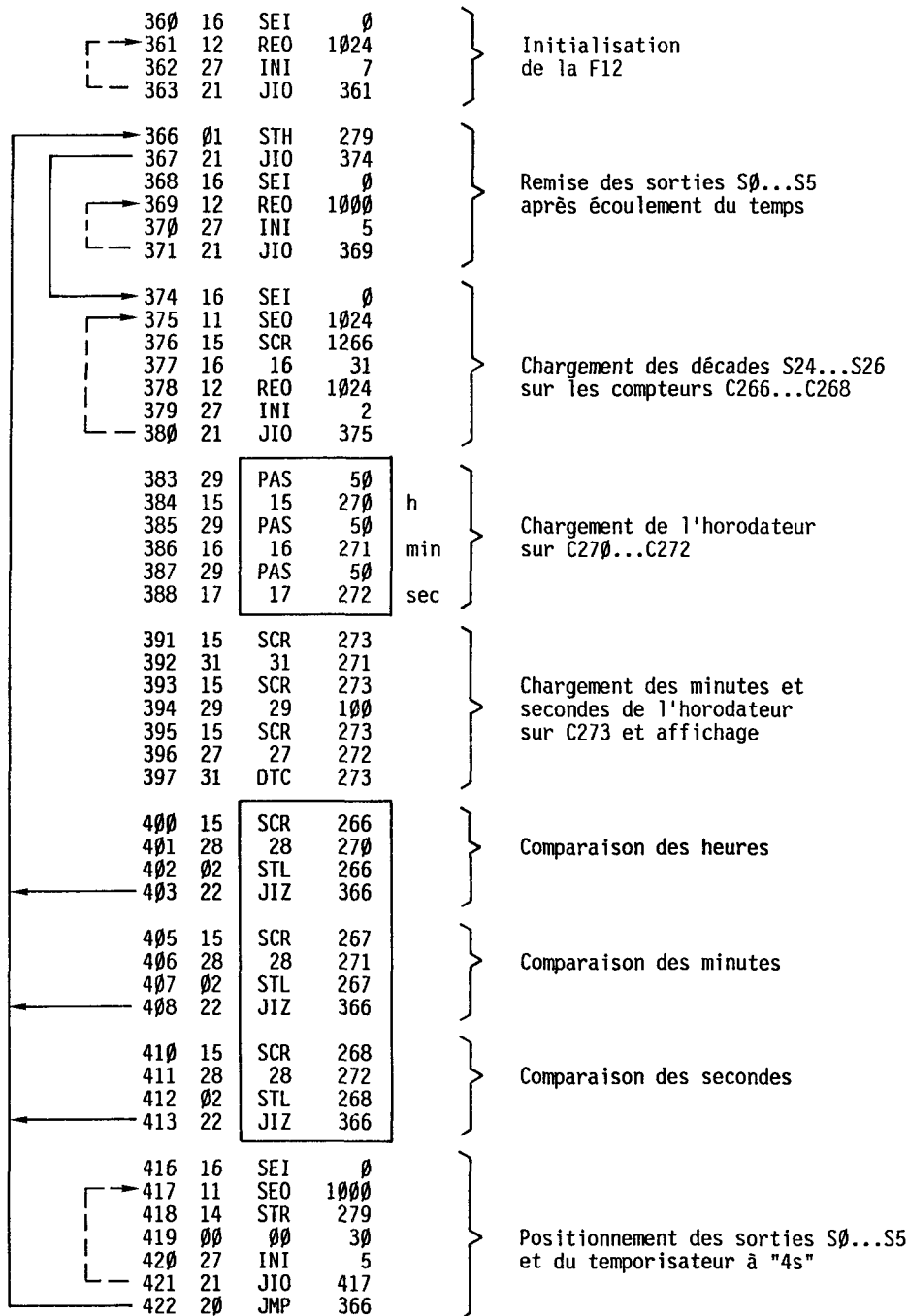


Schéma fonctionnel



*) Pendant une seconde le temporisateur est constamment positionné à nouveau. Pour activer les sorties durant un temps de 4s, il faut que le temporisateur soit positionné sur 3s.

Solution 1a



Problème supplémentaire 1b

En cas de coïncidence de l'horodateur avec le temps présélectionné, le texte suivant doit être imprimé:

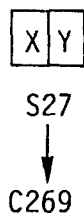
Date + heure (forme conventionnelle)

La série (SAIA-) PLC PCA14 est le produit (SAIA)
le plus récent à l'intérieur de la famille (SAIA-) PLC.

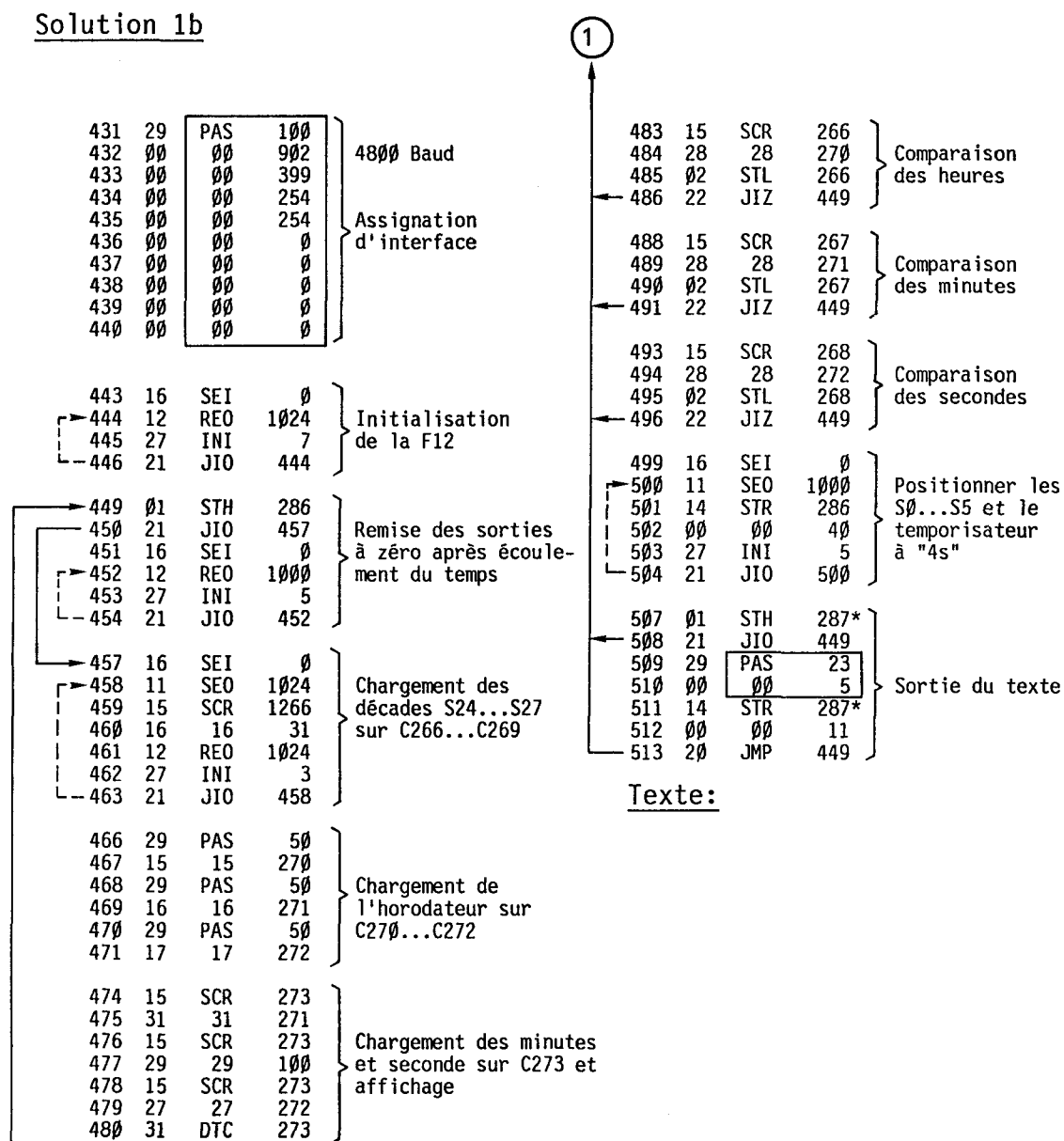
(SAIA-) PLC existe dans XY pays !

(SAIA)

- (SAIA-) doit être écrit en tant que sous-routine de texte.
- La valeur XY doit être lue sur les codeurs BCD S27 (sans zéros non significatifs).



Solution 1b



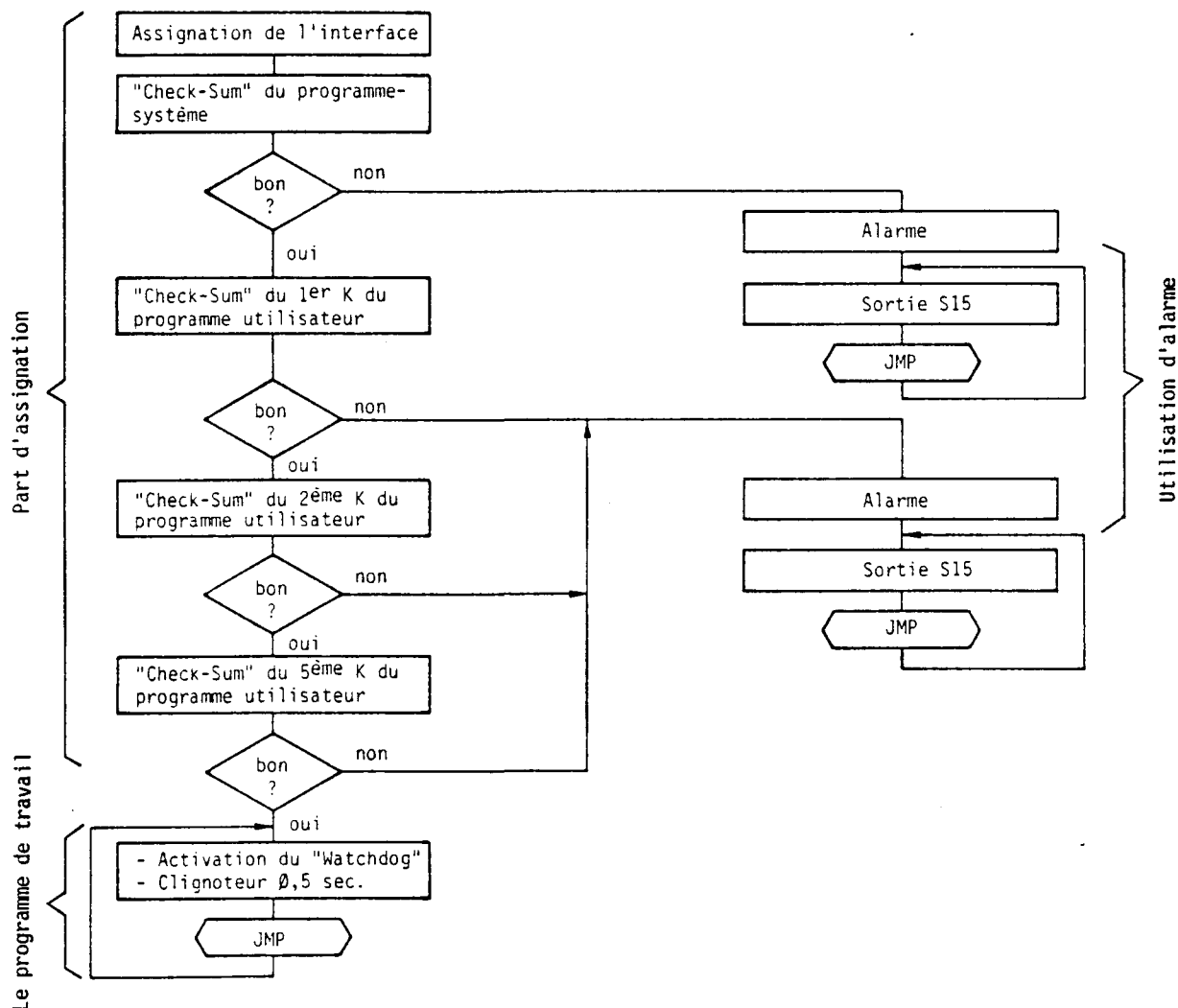
*) Pendant la seconde au cours de laquelle les conditions pour la sortie du texte sont remplies, celui-ci serait sorti plusieurs fois. Un temporisation (1,1s) assure que le texte ne soit sorti qu'une seule fois.

Exemple 2 "Check-Sum" du programme-système et programme utilisateur avec activation du "Watchdog" (Utilisation du PAS 30...38)

Problème 2

Lors de la première exécution de ce programme, un contrôle par totalisation du programme système et du programme utilisateur (1^{ère} K, 2^{ème} K et 5^{ème} K)* doit être effectué. Dans un petit programme de travail, le Watchdog doit être activé et la sortie S12 doit clignoter au rythme de 0,5s. Si un dérangement est constaté dans le programme-système, il faut que la signalisation "Contrôle par totalisation programme-système" soit donnée. Si un dérangement est constaté dans le programme utilisateur, c'est la signalisation "Contrôle par totalisation programme utilisateur" qui doit être donnée. Dans les deux cas, la sortie S15 (alarme) doit être activée.

Diagramme fonctionnel



*) Le 1^{ère} K et le 2^{ème} K contiennent le programme utilisateur, le 5^{ème} K les textes.

Des défauts dans le programme-système ne peuvent pas être simulés. Par contre, il est possible de en simuler dans le programme utilisateur, mémorisé en mémoire RAM, afin de provoquer une alarme.

Solution 2

521	29	PAS	100
522	00	00	902
523	00	00	399
524	00	00	254
525	00	00	254
526	00	00	0
527	00	00	0
528	00	00	0
529	00	00	0
530	00	00	0

4800 Baud

Attribution de
l'interface

Explication:

Le test sur la sommation de contrôle est effectué uniquement à l'enclenchement du PLC. Il appartient ainsi à la partie assignation (lignes 521 à 544).

Le programme de travail à proprement parler se trouve entre les lignes 546 et 551. Le reste des lignes de programme sert à la sortie de texte en cas de défaut.

532	29	PAS	30
533	00	00	0
534	22	JIZ	554
535	29	PAS	31
536	00	00	1929
537	22	JIZ	559
538	29	PAS	32
539	03	03	1397
540	22	JIZ	559
541	29	PAS	35
542	00	00	1815
543	22	JIZ	559
544	12	REO	15

Check-Sum du
programme-système

Check-Sum du
1ère K

Check-Sum du
2ème K

Check-Sum du 5ème K

Remise à zéro de l'alarme

Watchdog

Programme de travail

546	13	COO	255
547	02	STL	285
548	14	STR	285
549	00	00	5
550	13	COO	12
551	20	JMP	546

Sortie du texte et de l'alarme
en cas de défaut dans le
programme-système

554	29	PAS	23
555	00	00	100
556	11	SEO	15
557	20	JMP	556

Sortie du texte et de l'alarme
en cas de défaut dans le
programme-utilisateur

559	29	PAS	23
560	00	00	104
561	11	SEO	15
562	20	JMP	561

Texte 100:

```
->100  .$.H^M^J.S.O.M.M.E.
101   .D.E. .C.O.N.T.R.O.L
102   .E. .D.U. .P.R.O.G.R
103   .A.M.M.E. .S.Y.S.T.E
104   .M.E^M^J^J^@^@^@^@^@
```

Texte 104:

```
->105  .$.H^M^J.S.O.M.M.E.
106   .D.E. .C.O.N.T.R.O.L
107   .E. .D.U. .P.R.O.G.R
108   .A.M.M.E. .U.T.I.L.I
109   .S.A.T.E.U.R^M^J^J^@
```

*) Le blocage dans la boucle d'alarme est voulu. Lorsque le contrôle Check-Sum est négatif, ceci signifie que la mémoire du programm-système ou la mémoire utilisateur sont défectueuses et doivent être remplacées. Il est possible de prendre d'autres mesures de sécurité grâce au WD.

Exemple 3 Signes spéciaux pour textes

Problème 3

Tous les signes spéciaux doivent être représentés sur un écran.
Il s'agit de:

- a) Textes libres
- b) Sortie de la valeur d'un registre de compteur Cn, sans suppression des zéros non significatifs
- c) Sortie de la valeur d'un registre de compteur Cn, avec suppression des zéros non significatifs
- d) Sortie d'éléments (états logiques de 8 éléments consécutifs)
- e) Sortie du caractère ASCII* (formé par 8 éléments consécutifs)
- f) Contenu de l'horodateur, forme industrielle
- g) Contenu de l'horodateur, forme conventionnelle
- h) Contenu de l'horodateur, année, mois, jour
- i) Répétition de signes
- k) Sous-routine de textes
- l) Sortie de signe \$

*) Tous les caractères de 0...127 sont pris en compte. Les signes visibles sont imprimés, tandis que les signes de commandes (0...31) agissent directement et sont remplacés par un SPACE.

SAIA SA	a)	MORAT
\$C		\$c
b) 00016		c) 16
\$E		\$e
d) 01011001		e) Y
\$T	\$H	\$D
f) 47 04/17 01 46	g) 83-11-24/17 01 46	h) 83-11-24
i) \$-	-----	
\$L, \$U k) LE (PCA14), PC-COMPACT POUR DES OPERATIONS COMPLEXES: AVEC LE (PCA14) VOTRE COMMANDE EST PLUS PERFORMANTE		

Solution 3

(Les paramètres et les diverses instructions du curseur se réfèrent au terminale VC 4404).

```

951 29 PAS 100
952 00 00 902 ; 4800 Baud
953 00 00 333 ; TXB
954 00 00 254
955 00 00 254
956 00 00 0
957 00 00 0
958 00 00 0
959 00 00 0
960 00 00 0

```

Textes:

```

962 29 PAS 23
963 00 00 0 ; Texte No 0

966 25 WIH 333 ; TXB
967 29 PAS 23
968 00 00 50 ; Texte No 50

```

```

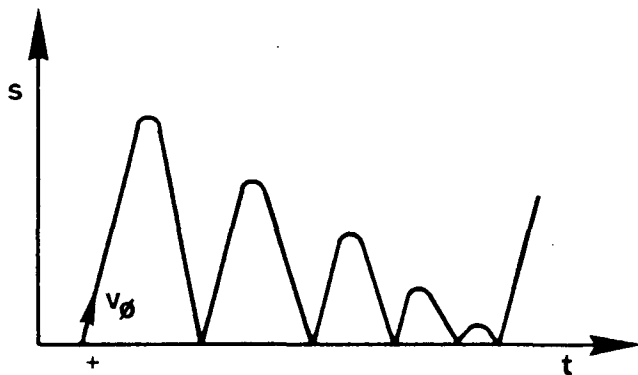
971 15 SCR 290
972 00 00 0
973 25 WIH 333 ; TXB
974 14 STR 259
975 00 00 10
976 29 PAS 23
977 00 00 30 ; Texte No 30
978 25 WIH 259
979 17 INC 290
980 20 JMP 973

```

Exemple 4 Jeu de balle

Supposons:

Une balle est jetée en l'air avec une vitesse initiale v_0 . En retombant sur une surface horizontale, elle perd chaque fois $X\%$ de sa vitesse.



$$s = v_0 \times t - \frac{g}{2} \times t^2$$

$$g = 10 \text{ m/s}^2$$

$$t = t/10 \text{ (1/10 s)}$$

$$\Rightarrow s = \frac{v_0 \times t}{10} - \frac{10 \times t^2 / 100}{2} \text{ [m]}$$

$$s = 1/10 (v_0 \times t - t^2/2) \text{ [m]}$$

Problème 4

A chaque dixième de seconde environ, il faut calculer la hauteur s qui doit être affichée sur l'écran.

A chaque rebondissement sur le sol, v_0 se réduit de la valeur en pourcent présélectionnée sur le canal S24 (de F12).

8	6
---	---

 % de v_0

S24

—
C313

Solution 4

v_0 est déposée sur C314.

La formule $v_0 \times t^2/2$ est réalisée à l'intérieur d'un registre de compteur.

Grâce à l'utilisation de signes spéciaux, le texte est d'une brièveté surprenante. Seulement 16 caractères!

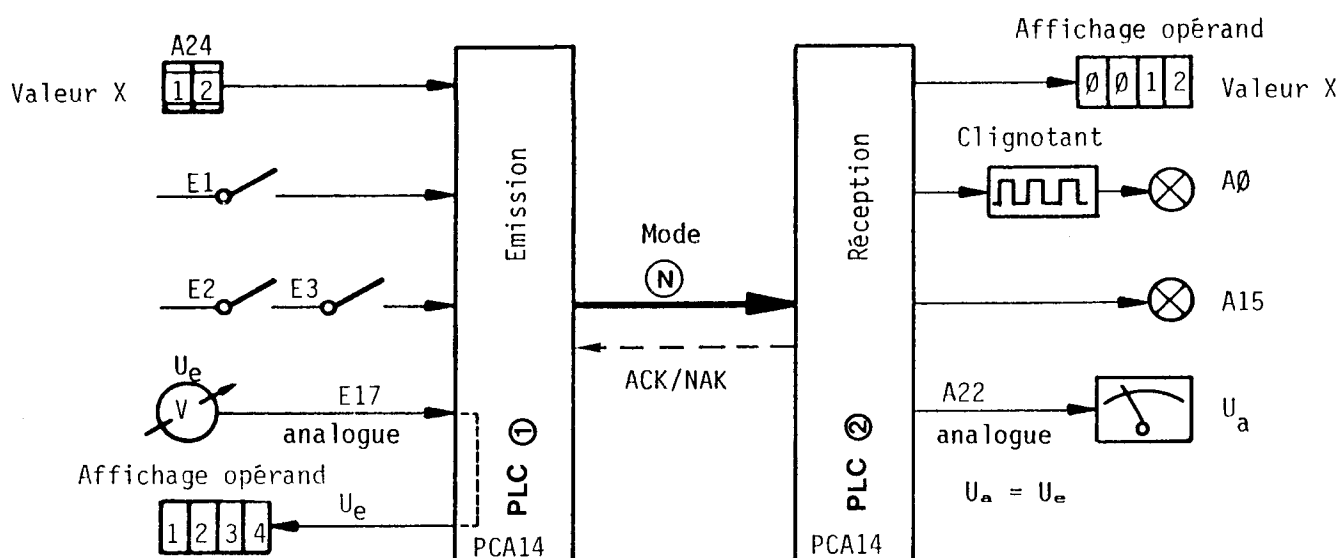
651	29	PAS	100	
652	00	00	902	; 4800 Baud
653	00	00	600	; TXB
654	00	00	254	; RBY
655	00	00	254	; TBY
656	00	00	0	
657	00	00	0	
658	00	00	0	
659	00	00	0	
660	00	00	0	
661	11	SEO	24	Entrée de l'amplitude résiduelle en % par S24
662	15	SCR	313	
663	16	16	31	
664	12	REO	24	
665	15	SCR	314	
666	00	00	11	v_0
667	15	SCR	315	
668	00	00	0	$t = 0$
669	15	SCR	317	
670	31	31	314	v
671	15	SCR	317	v
672	29	29	315	t
673	15	SCR	316	
674	31	31	315	t
675	15	SCR	316	t
676	29	29	316	t
677	15	SCR	316	t^2
678	30	30	2	2
679	15	SCR	317	vxt
680	28	28	316	$t^2/2$
681	15	SCR	318	
682	31	31	317	s
683	17	INC	315	$t + 1$
684	21	JIO	692	
685	15	SCR	314	v
686	29	29	313	%
687	15	SCR	314	$vx\%$
688	30	30	100	100
689	01	STH	314	$v = 0 ?$
690	21	JIO	667	
691	20	JMP	661	$v = 0$ --> Demarrer à nouveau
692	29	PAS	23	Texte No 3
693	00	00	3	
694	25	WIH	600	; Attendre aussi
695	00	NOP	0	longtemps que
696	20	JMP	669	TXB = H

Texte: Sortie du reg. C318

Exemple 5 Echange de données entre deux PCA14 dans le mode N par l'interface série

Problème 5

En mode N, diverses entrées du premier PLC doivent être transmises par l'interface série au deuxième PLC afin d'être traitées, pour ensuite être sorties. Il s'agit de:



Fonctions du PLC ① (Emetteur)

a) Définition PAS 100 à l'adresse 601

- TXB = 254
- RBY = 525
- TBY = 425

b) Réalisation d'un programme pour le traitement des entrées d'après la configuration susmentionnée. La valeur analogue U_e du canal E17 doit être affichée en tant que valeur binaire dans l'opérand.

c) Emission du télégramme.

d) La boucle doit être active pour le PLC ① (le PLC ① est fournisseur de courant).

Fonctions du PLC (2) (Récepteur)

a) Définition PAS 100 à l'adresse 601

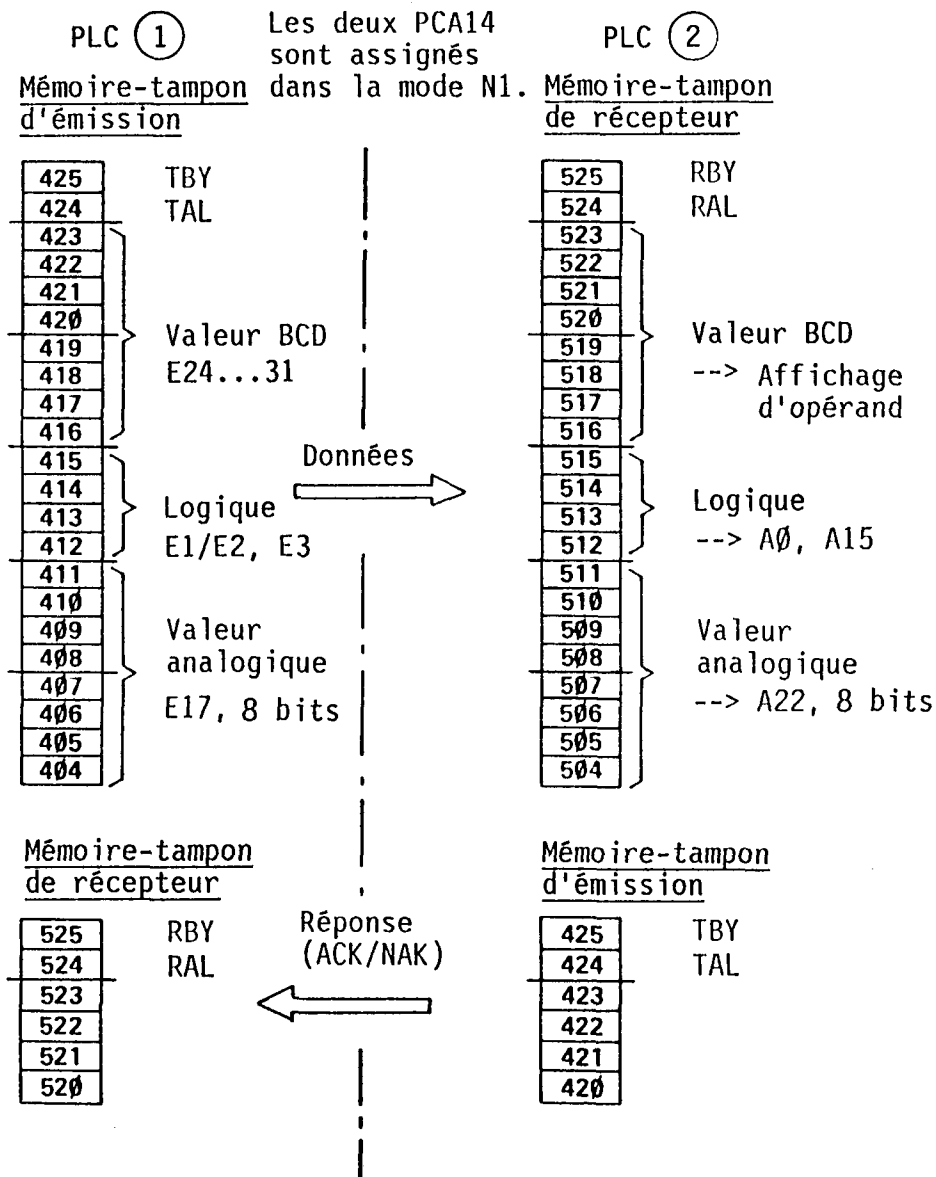
- TXB = 254
- RBY = 525
- TBY = 425

b) Réalisation d'un programme de traitement d'après la configuration mentionnée ci-dessus. La sortie S0 doit clignoter en fonction de la valeur BCD X.

c) Libération de la réception.

d) Le PLC (2) doit être passif.

Solution 5



PCA ①

Emetteur

601	29	PAS	100	
602	00	00	902	;4800 Baud
603	00	00	254	;TXB
604	01	01	525	;RBY
605	05	05	425	;TBY
606	01	01	0	;BCC
607	00	00	0	
608	00	00	0	
609	00	00	0	
610	00	00	0	

PCA ②

Récepteur

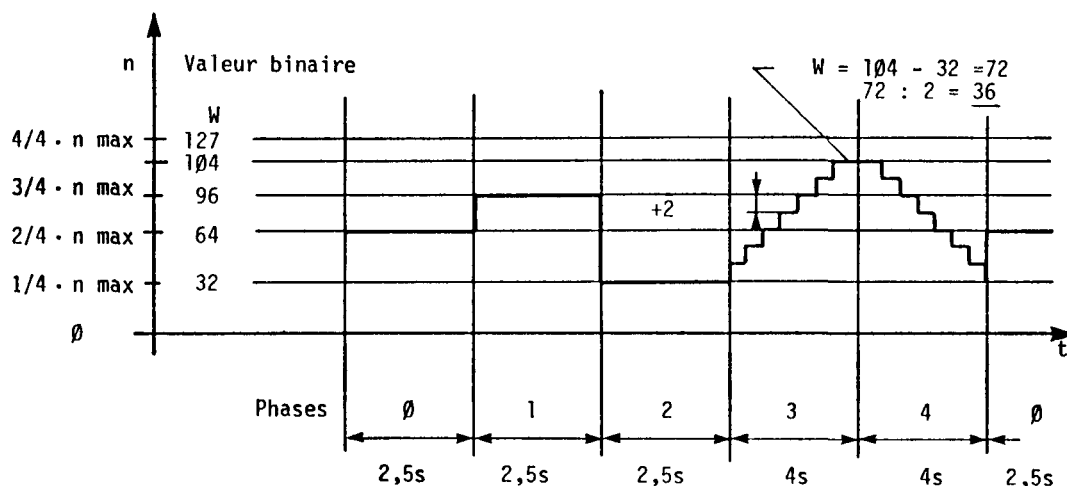
601	29	PAS	100	
602	00	00	902	;4800 Baud
603	00	00	254	;TXB
604	05	05	525	;RBY
605	01	01	425	;TBY
606	01	01	0	;BCC
607	00	00	0	
608	00	00	0	
609	00	00	0	
610	00	00	0	

613	01	STH	425	} Emetteur libre?
614	21	JIO	639	
616	11	SEO	24	} Mettre dans la mémoire tampon- émetteur la valeur BCD du S24
617	15	SCR	299	
618	24	24	31	
619	12	REO	24	
620	15	SCR	299	
621	21	21	423	
623	01	STH	1	} Mettre les ré- sultats des combi- naisons log. dans la mém. tampon
624	10	OUT	412	
625	01	STH	2	
626	03	ANH	3	
627	10	OUT	413	
629	19	SEA	0	} Mettre la valeur analogique du canal E17 dans la mémoire tampon
630	10	OUT	17	
631	15	SCR	298	
632	24	24	23	
633	15	SCR	298	
634	21	21	411	
636	19	SEA	0	} Transmettre le télégramme
637	11	SEO	425	
639	31	DTC	298	;Affichage de la valeur analogique
640	20	JMP	613	

613	01	STH	525	} Récepteur libre?
614	22	JIZ	633	
616	12	REO	525	;Récep. terminée
618	15	SCR	257	} Valeur BCD sur l'affichage d'opérand
619	16	16	523	
620	31	DTC	257	
622	15	SCR	258	} Transmettre la valeur analog. sur canal S22
623	24	24	511	
624	15	SCR	258	
625	21	21	23	
626	11	SEO	23	
627	05	ORH	22	
628	12	REO	23	
630	01	STH	513	} Sortie S15
631	10	OUT	15	
633	02	STL	512	} Clignoteur sur la sortie S0 (t = valeur BCD)
634	11	SEO	400	
635	01	STH	512	
636	04	ANL	256	
637	14	STR	256	
638	16	16	523	
639	13	COO	400	
640	01	STH	400	
641	10	OUT	0	
642	20	JMP	613	

Exemple 6 Boucle de réglage PID rapide avec l'instruction PAS 202 et le PCA14 (Moteur-tachymètre)

La vitesse du moteur est contrôlée par un tachymètre, ce moteur doit être capable de suivre une courbe de valeur de consigne réglable $W = f(t)$. Cette courbe a l'allure suivante:



Les 5 phases temporels 0...4 sont définis par les parties de programmes suivantes :

```

100  STR 281 ; Phase 0
      00 25
      SCR 260 } W = 1/2 n max
      00 64
      JMS 150 ---> PID
      STH 281
      JIO 104
  
```

```

110  STR 281 ; Phase 1
      00 25
      SCR 260 } W = 3/4 n max
      00 96
      JMS 150 ---> PID
      STH 281
      JIO 114
  
```

```

120  STR 281 ; Phase 2
      00 25
      SCR 260 } W = 1/4 n max
      00 32
      JMS 150 ---> PID
      STH 281
      JIO 124
  
```

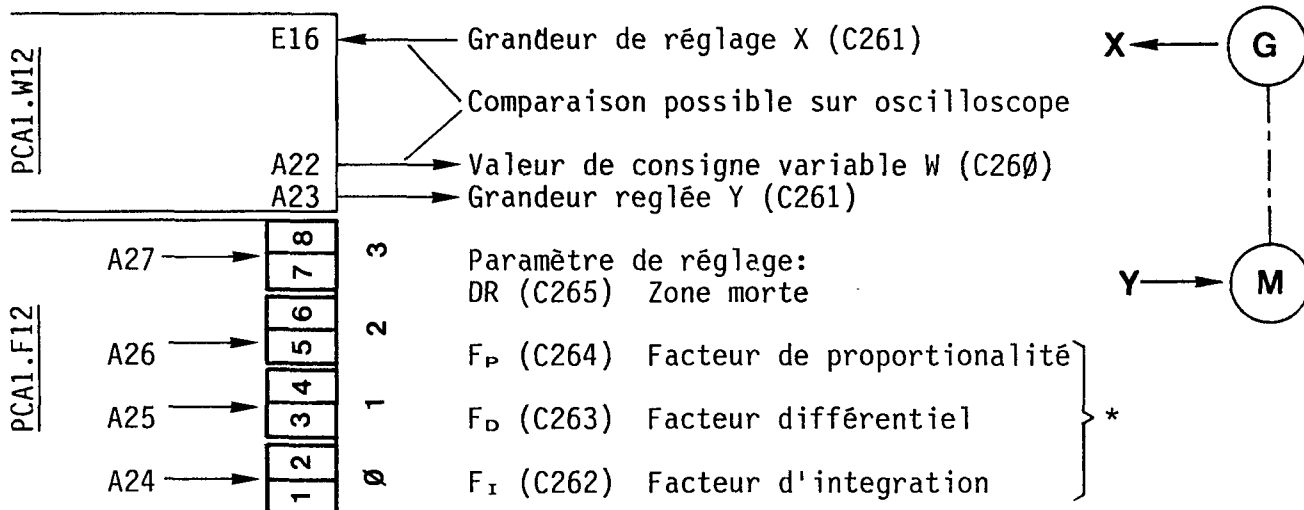
```

130  SCR 300 ; Phase 3
      00 36
      INC 260 } W + 2
      INC 260
      JMS 150 ---> PID
      DEC 300
      STH 300
      JIO 132
  
```

```

140  SCR 300 ; Phase 4
      00 36
      DEC 260 } W - 2
      DEC 260
      JMS 150 ---> PID
      DEC 300
      STH 300
      JIO 142
      JMP 100 --->
  
```

Le module de présélection PCA1.F12 permet de faire varier les paramètres de réglage même pendant la régulation. Letachymètre du moteur est relié au module analogique d'entrée PCA1.W12.



La sous-routine de réglage 150 est bâtie comme suit:

```

--> 150 SEI 0
      SEO 1024 FI
      SCR 1262 FD
      17* 31 FP
      REO 1024
      INI 2
      J10 151
      SEO 27
      SCR 265 DR
      16 31
      REO 27
  
```

Lire les compteurs BCD A24...A27 et transfert sur les compteurs C262...C265

```

170 WIH 280
     OUT 16
     SCR 261
     24 23
  
```

Attendre le temps de prise de l'information
Prise de l'information sur le canal 16
Lire la grandeur X
et transfert sur le compteur C261

```

174 PAS 202
     5 00 31
     6 00 260
     7 00 0
     8 00 0
     9 00 0
    180 00 0
     1 00 0
     2 00 0
     3 00 0
  
```

Bloc de donnée 31
compteurs 260, 261, 262, 263, 264, 265

```

4 JMP 185 **
185 SCR 261
     21 23
     SEO 23
     ORH 23
     REO 23
  
```

Sortie de la grandeur réglée
sur le canal A23

```

190 SCR 260
     21 23
     SEO 23
     ORH 22
     REO 23
  
```

Consigne variable W
sortie sur le canal A22

```

195 STR 280
     00 1
    197 RET 0
  
```

Attendre le temps de prise de l'information
 $T_p = 0,1 \text{ sec}$

*) Les valeurs lues par les compteurs BCD sont multipliées par 10. Par exemple si la présélection = 50, les valeurs F_I, F_D, F_P sont calculées comme suit:
 $50 \times 10 = 500 / 500 : 256 = 1,95$

**) Par l'appel de l'instruction JMP, un changement de programme parallèle a lieu.

Exemple 7 Réglage manuel/PID avec PAS 211 et PCA14 (Module analogique 12 bits PCA1.W32)

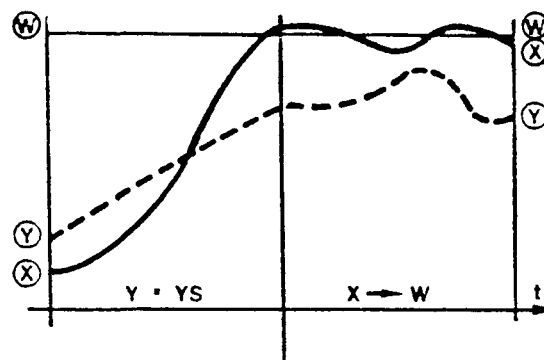
Problème 7

Le programme suivant doit mettre en évidence le passage de la commande manuelle directe de la valeur de réglage à la régulation PID automatique. Les sous-programmes utilisés sont les routines d'entrées et de sorties par lecture de la documentation du module analogique PCA1.W32.

La valeur de consigne W est présélectionnée en $0,1V$ sur le commutateur BCD A24. Le temps de balayage T_0 est réglable en $0,1s$ avec les commutateurs BCD E4...E11. L'opération est lancée avec E0.

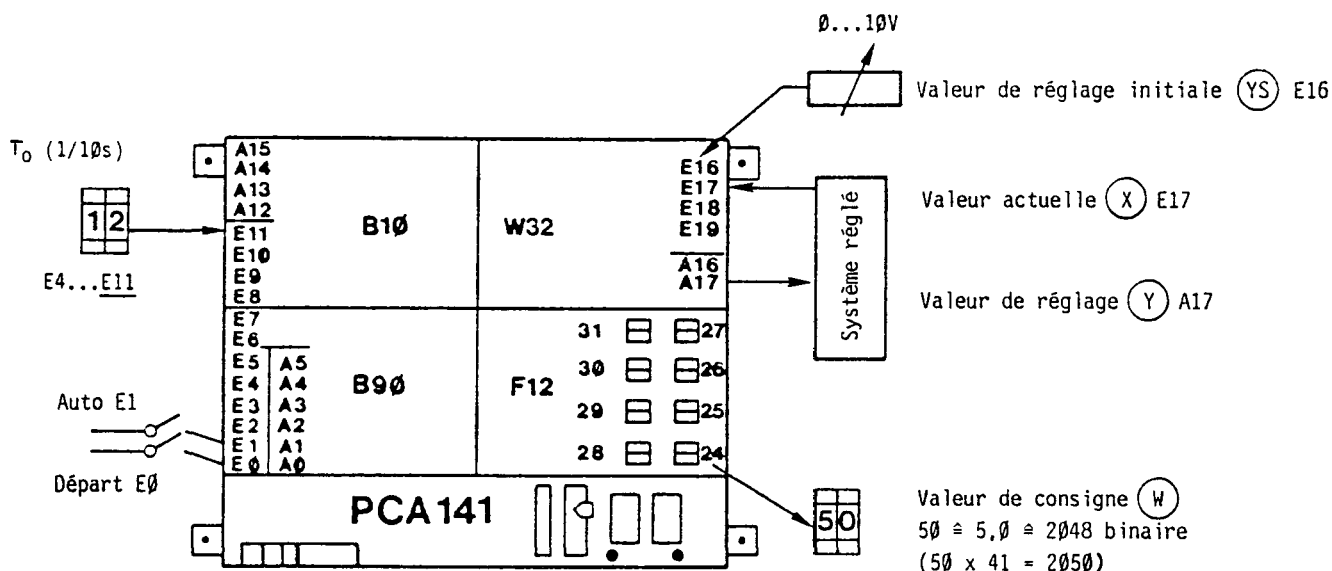
Si E1 = L, la valeur de réglage Y peut être modifiée en commande manuelle (potentiomètre) via l'entrée analogique E16, jusqu'à ce que $X = W$.

Avec E1 = H, on passe en réglage PID, ce qui a pour effet de compenser automatiquement toute perturbation $X \rightarrow W$.

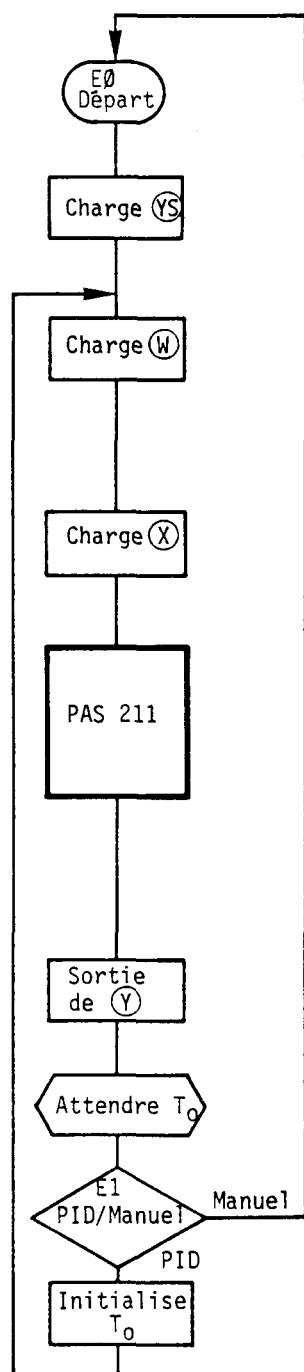


Commande manuelle de la valeur de réglage Y par $Y_S = Y$

Configuration matérielle



Solution 7



50	WIL	Ø	Départ																					
51	JMS	550**	} Lecture de la valeur de réglage (YS) introduite par le canal ana- logique E16 et transfert sur C282																					
52	SCR	282																						
53	31	300																						
54	SEO	24	} Charger la valeur de consigne (W) du commutateur BCD S24 dans C276																					
55	SCR	276																						
56	16	31																						
57	REO	24																						
58	SCR	276	} Multipliés par 41, c'est-à-dire convertir en valeur binaire *																					
59	29	41																						
60	JMS	500**	} Introduction par lecture de la valeur actuelle (X) par le canal anal. E17 et transfert sur C277																					
61	SCR	277																						
62	31	300																						
63	<table><tr><td>PAS</td><td>211</td></tr><tr><td>12</td><td>31</td></tr><tr><td>ØØ</td><td>276</td></tr><tr><td>ØØ</td><td>277</td></tr><tr><td colspan="2">F_I</td></tr><tr><td colspan="2">F_D</td></tr><tr><td colspan="2">F_P</td></tr><tr><td colspan="2">D_R</td></tr><tr><td>ØØ</td><td>282</td></tr><tr><td>ØØ</td><td>Ø</td></tr></table>		PAS	211	12	31	ØØ	276	ØØ	277	F _I		F _D		F _P		D _R		ØØ	282	ØØ	Ø	12 bits, bloc de données 31 (W) (X) / (Y) (YS)	
PAS	211																							
12	31																							
ØØ	276																							
ØØ	277																							
F _I																								
F _D																								
F _P																								
D _R																								
ØØ	282																							
ØØ	Ø																							
73	SCR	301	} Sortir la valeur de réglage (Y) sur le canal de sortie S17																					
74	31	277																						
75	JMS	700**																						
76	WIH	256	Attendre le temps de balayage T _Ø																					
77	STH	1	} Interrogation PID/Manuel																					
78	JIZ	50																						
79	STR	256	} Charger le temps de balayage T _Ø du commutateur BCD et démarrer																					
80	16	11																						
81	JMP	54																						

Comme décrit dans l'énoncé de la tâche, ce programme permet d'amener manuellement, en variant le potentiomètre (YS) (E16), la valeur actuelle aux environs de la valeur de consigne (W). On passe ensuite sans à-coups en réglage PID en fermant E1.

*) Pour les sous-programmes, voir chapitre "PCA1.W3..", manuel Hardware PCA1.

**) Le sous-programme 550 correspond au sous-programme 500 en remplaçant la deuxième ligne par OUT 16.

Exemple 8 Registre à décalage rotatif avant/arrière avec remise à zéro
 (Utilisation de l'instruction PAS 250)

Fonctionnement:

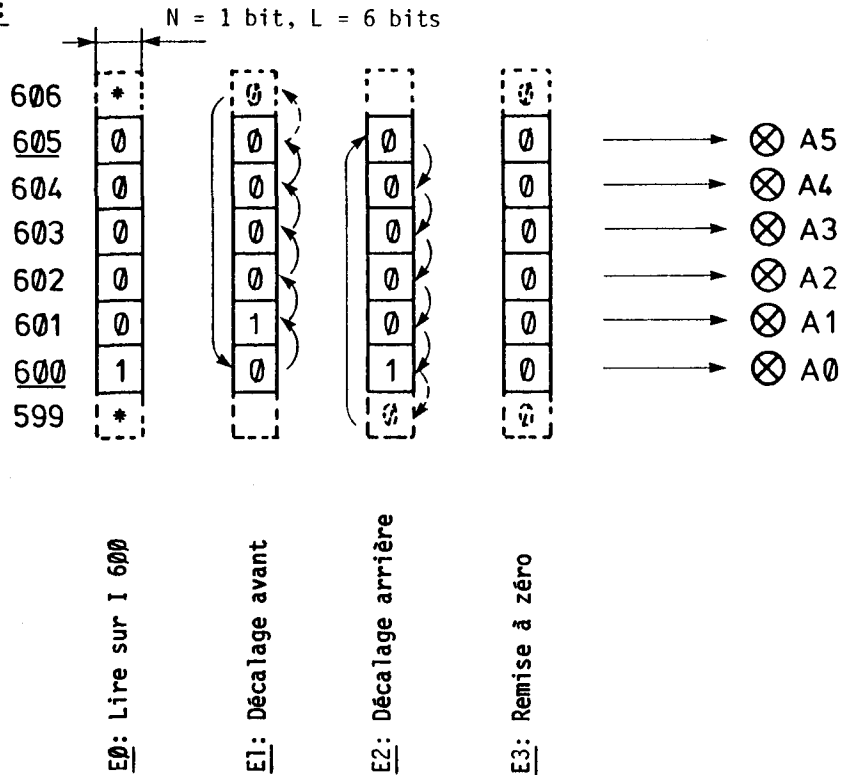
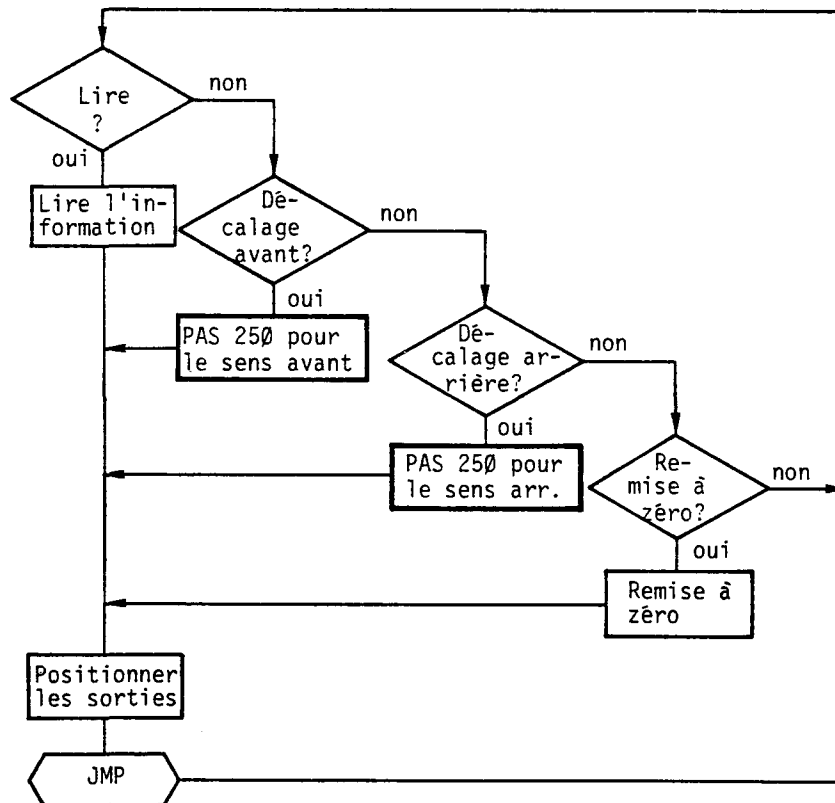


Diagramme de flux



Solution 8

```

1001 01 STH 0
1002 09 DYN 500
1003 22 JIZ 1007
1004 10 OUT 600
1005 20 JMP 1055 ---->

```

Lecture

```

1007 01 STH 1
1008 09 DYN 501
1009 22 JIZ 1026

```

1011	29	PAS	250
1012	00	00	600
1013	00	00	605
1014	00	00	1
1015	00	00	0
1016	00	00	0
1017	00	00	0
1018	00	00	0
1019	00	00	0
1020	00	00	0

; SADD
; DADD
; 1 bit

Décales dans le sens avant

```

1022 20 JMP 1055 ---->

```

```

1026 01 STH 2
1027 09 DYN 502
1028 22 JIZ 1044

```

Décales dans le sens arrière

1031	29	PAS	250
1032	00	00	605
1033	00	00	600
1034	00	00	1
1035	00	00	0
1036	00	00	0
1037	00	00	0
1038	00	00	0
1039	00	00	0
1040	00	00	0

; SADD
; DADD
; 1 bit

```

1042 20 JMP 1055 ---->

```

```

1044 01 STH 3
1045 09 DYN 503
1046 22 JIZ 1055 ---->
1047 16 SEI 0
1048 12 REO 1599
1049 27 INI 7
1050 21 JIO 1048

```

Remise à zéro

```

1055 16 SEI 0
1056 01 STH 1600
1057 10 OUT 1000
1058 27 INI 5
1059 21 JIO 1056
1060 20 JMP 1001

```

Positionnement des
sorties S0...S5

Cet exemple correspond à un registre simple et en donne le principe de fonctionnement. Dans ce cas l'utilisation de cette instruction ne présente aucun avantage. Par contre lors de registre à décalage de taille importante et temps d'exécution critique l'instruction PAS 250 apporte un certain confort à la programmation.

Exemple 9 Registre à décalage BCD avant/arrière (Utilisation du PAS 250)

Description

Le registre à décalage possède 4 positions à 4 bits. L'état d'entrée sera lu sur des codeurs BCD connectés aux entrées E4...E11. En décalage avant, ce sera le chiffre de gauche (E4...E7) qui sera lu, tandis qu'en décalage arrière, ce sera le chiffre de droite (E8...E11) qui sera lu.

Le registre est formé sur les indicateurs 404...419. Les instructions de décalage sont données par l'entrée E0 (décalage avant) resp. l'entrée E1 (décalage arrière). Le contenu du registre sera affiché dans le champ opérand du clavier de programmation.

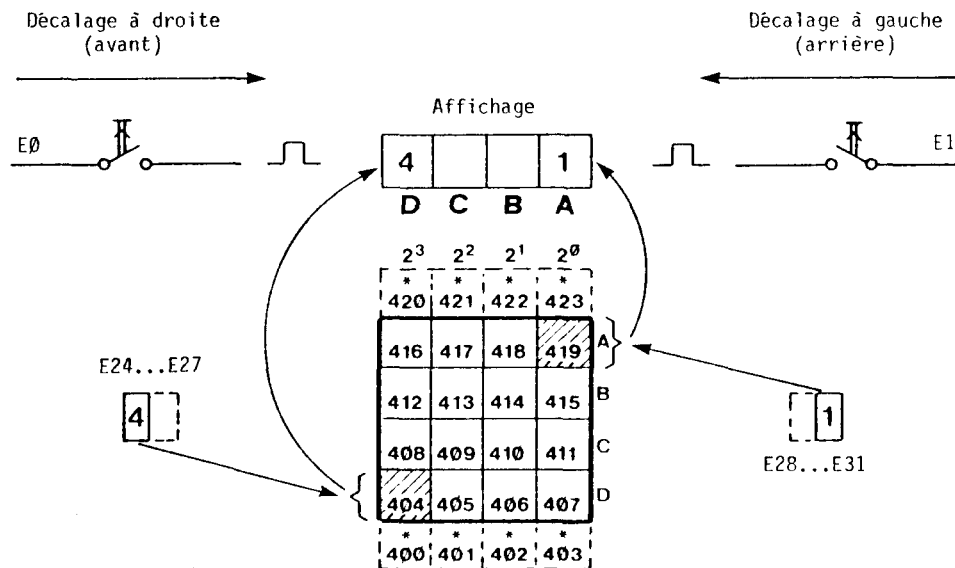
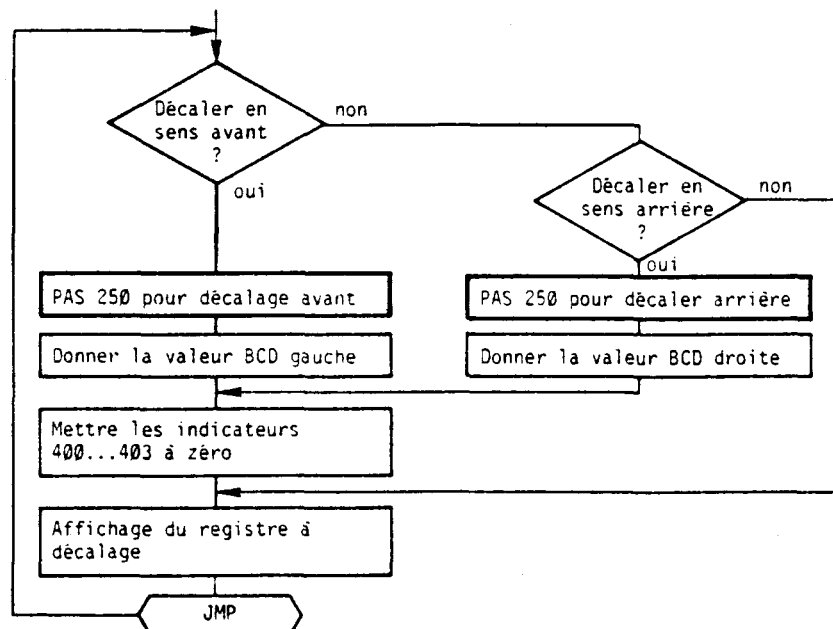
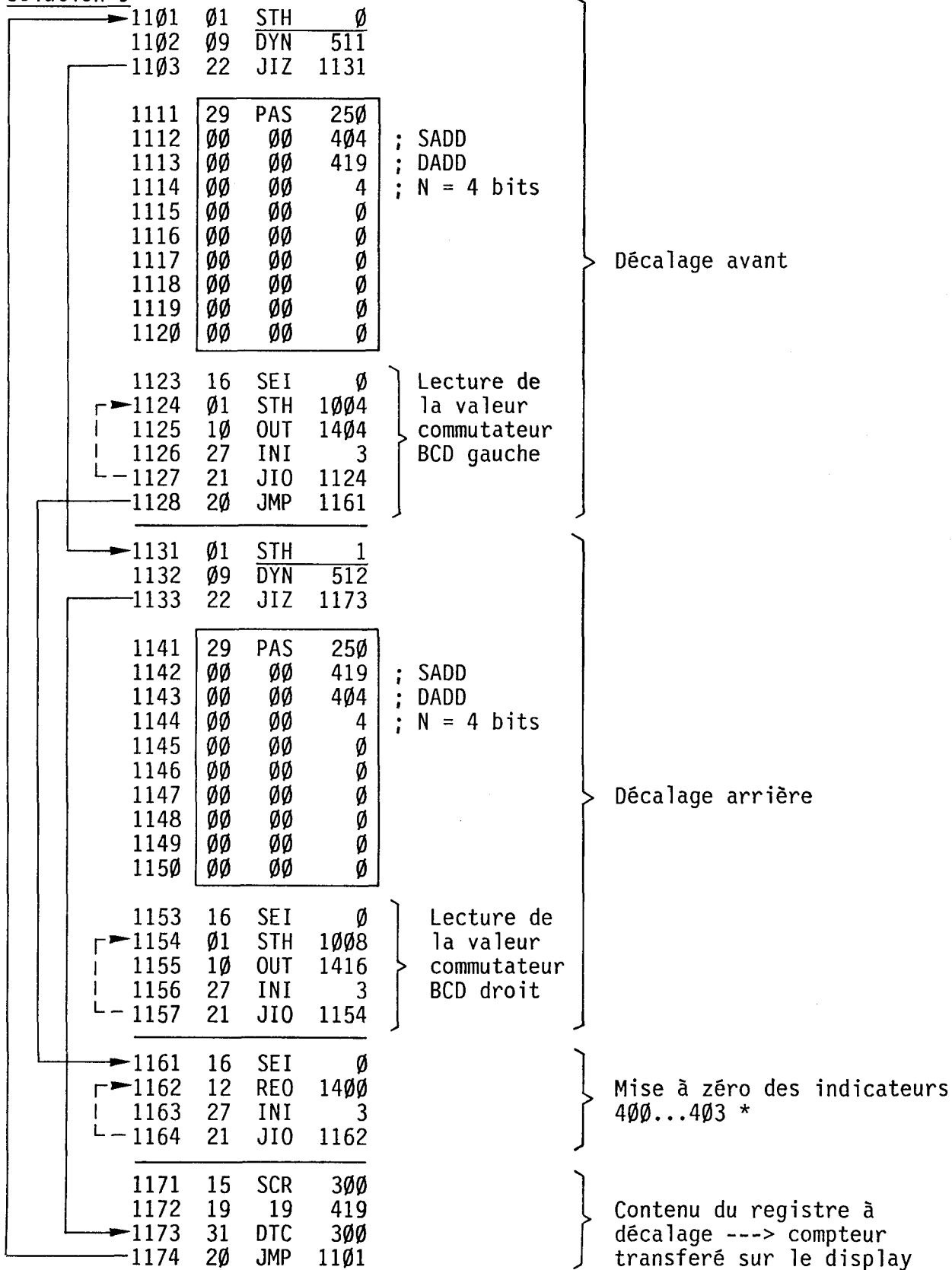


Diagramme du déroulement



*) Les indicateurs 400...403 et 420...423 doivent être réservés comme mémoire tampon.

Solution 9

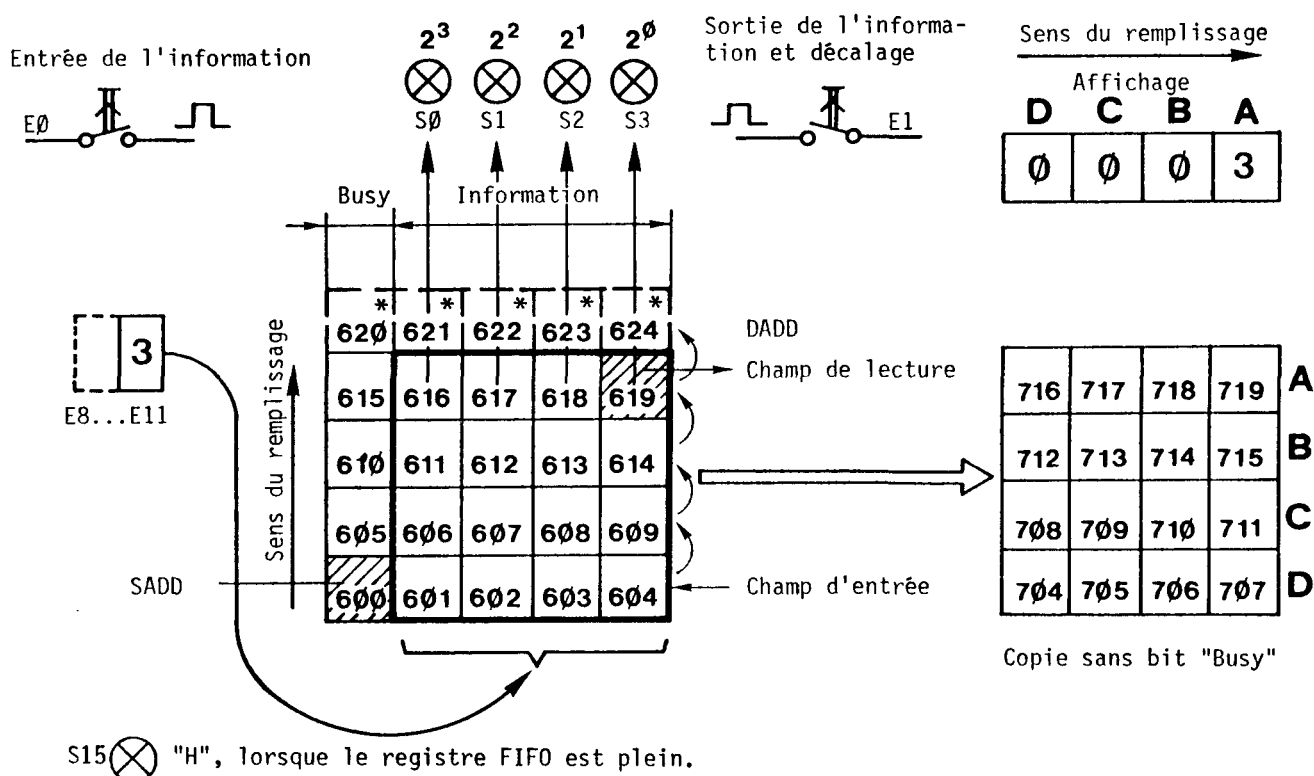


*) Les indicateurs 400...403 doivent être positionné à zéro afin de pouvoir lire 20 bits BCD dans le compteur avec le code 19.

Exemple 10 Registre FIFO (utilisation de l'instruction PAS 251)

Description

Le registre FIFO possède 4 positions à 4 bits. L'état d'entrée sera lu sur un codeur BCD connecté aux entrées E8...E11. Le registre est formé par les indicateurs 600...619. Par l'entrée E0 une valeur sera mémorisée dans le registre FIFO. Par l'entrée E1 la première valeur dans le FIFO sera positionnée sur les sorties S0...S3. La sortie S15 sera activée lorsque le registre sera plein. Les chiffres seront affichés dans le champ opérant du clavier de programmation.



Afin de pouvoir afficher le contenu du registre FIFO dans le champ opérant du clavier de programmation, il faut copier sur une autre plage d'indicateur uniquement l'information (sans le bit "Busy"). Ensuite un compteur peut être chargé avec le code 19 (5 x 4 bits BCD) et affiché.

*) Les indicateurs 620...624 doivent être réservés afin de pouvoir procéder au décalage.

Solution 10

```

1201 01 STH 0
1202 09 DYN 520

```

```

1203 22 JIZ 1227
1204 11 SEO 600
1205 16 SEI 0
1206 01 STH 1008
1207 10 OUT 1601
1208 27 INI 3
1209 21 JIO 1206

```

; Positionner le Busy

Charger la
valeur BCD

```

1211 29 PAS 251
1212 00 00 600
1213 00 00 619
1214 00 00 5
1215 00 00 0
1216 00 00 0
1217 00 00 0
1218 00 00 0
1219 00 00 0
1220 00 00 0

```

; SADD
; DADD
; 5 bits (4 + Busy)Lecture de la valeur
dans le champ d'entrée

```

1222 01 STH 600 ; Busy ?
1223 10 OUT 15 ; FIFO plein
1224 20 JMP 1260 ---->

```

```

1227 01 STH 1
1228 09 DYN 521
1229 22 JIZ 1292

```

```

1231 16 SEI 0
1232 01 STH 1616
1233 10 OUT 1000
1234 27 INI 3
1235 21 JIO 1232

```

Sortie de la
valeur BCDPlacement de la pre-
mière valeur sur les
sorties S0...S3

```

1241 29 PAS 250
1242 00 00 600
1243 00 00 619
1244 00 00 5
1245 00 00 0
1246 00 00 0
1247 00 00 0
1248 00 00 0
1249 00 00 0
1250 00 00 0

```

; SADD
; DADD
; N = 5 bitsDéplacement &
mettre le champ
d'entrée à zéro

```

1252 16 SEI 0
1253 12 REO 1600
1254 27 INI 4
1255 21 JIO 1253
1256 12 REO 15 ; FIFO libre

```

①

Copie des FIFO sur les indicateurs 704...719 pour affichage DTC

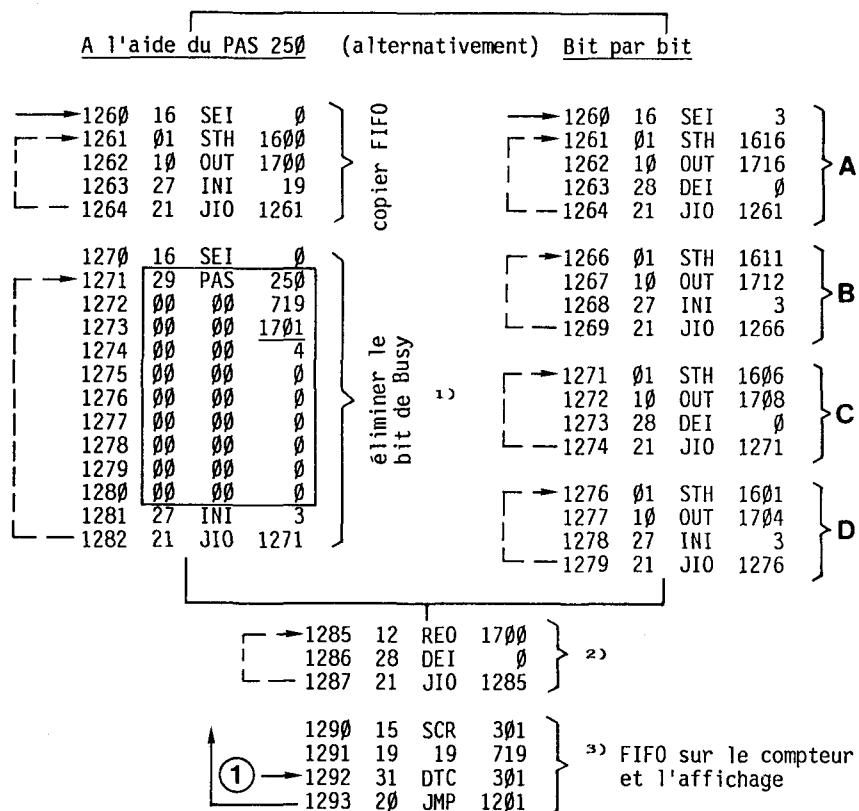
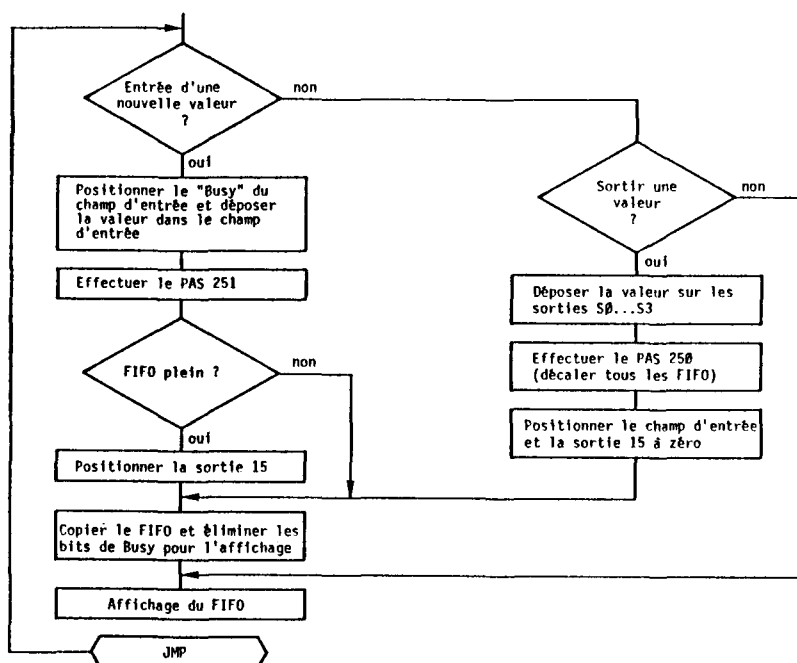


Diagramme du déroulement



- 1) Les indicateurs 697...700 doivent être réservés comme mémoire tampon pour le bon fonctionnement du PAS 250.
- 2) Les indicateurs 700...703 doivent être positionnés à zéro afin de charger correctement un compteur avec le code 19 (5 x 4 bits BCD).
- 3) Tout le programme se trouvant sur cette page est nécessaire que si la valeur doit être affichée par DTC.

Exemple 11 Signalisations de dérangements
 (Mémorisation dans le registre FIFO par PAS 251)

Description

Divers dérangements doivent être sortis en textes clairs. Il faut veiller à ce que les dérangements soient imprimés dans l'ordre de leur apparition. Par ailleurs, des signalisations de dérangements ultérieures ne doivent pas être perdues lorsque les nouvelles annonces de dérangement apparaissent pendant l'impression.

Problème 11

4 dérangements différents doivent être simulés par les entrées 0...3 et imprimés comme suit, en fonction de leur ordre, avec la date et l'heure.

(pour simplifier, seule l'heure où l'impression du dérangement est effectuée sera imprimée et non pas celle où le dérangement a eu lieu).

E0 : Date, heure
 Alarme 0
 Commentaire

E1 : Date, heure
 Alarme 1
 Commentaire

E2 : Date, heure
 Alarme 2
 Commentaire

E3 : Date, heure
 Alarme 3
 Commentaire

Solution 11

Le problème est résolu à l'aide du registre FIFO. A chaque signalisation d'un dérangement un code de reconnaissance est attribué. En cas de dérangement, le code correspondant est enregistré au niveau d'entrée du FIFO et, en même temps, le bit Busy (SADD) est mis sur "H". Ensuite est traité le PAS 251 et l'information (Code + Busy) est déplacée à la dernière place libre du FIFO (si le FIFO est vide: jusqu'au niveau de sortie). Lorsque le bit Busy du niveau de sortie se trouve sur "H" la nature du dérangement est recherchée à l'aide du code. (Code 00 signifie alarme 0, code 10 ---> alarme 1, code 01 ---> alarme 2, code 11 ---> alarme 3).

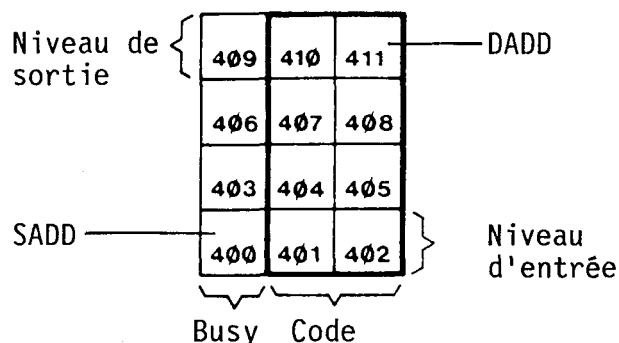
Suivant le code, le compteur C319 est mis à la valeur 0, 10, 20 ou 30. Ensuite le registre d'index est chargé avec la valeur du compteur C319.

Grâce à la sortie de texte indexée

PAS 23
00 1110

la valeur du registre d'index (0, 10, 20 ou 30) est ajoutée au numéro de texte 110 et ainsi, le texte No 110, 120, 130 ou 140 est sorti, ce qui correspond aux alarmes 0, 1, 2 ou 3. Après la sortie de texte, le registre FIFO est déplacé par PAS 250.

Le registre FIFO se présente comme suit:



Interprétation du niveau de sortie:

1	0	0
409	410	411

Alarme 0 ---> C319 chargé avec la valeur 0

1	1	0
409	410	411

Alarme 1 ---> C319 chargé avec la valeur 10

1	0	1
409	410	411

Alarme 2 ---> C319 chargé avec la valeur 20

1	1	1
409	410	411

Alarme 3 ---> C319 chargé avec la valeur 30

Busy Code

Programme de l'exemple 11

```

1301 29 PAS 100
1302 00 00 902 ; 4800 Baud
1303 00 00 425 ; TXB
1304 00 00 254
1305 00 00 254
1306 00 00 0
1307 00 00 0
1308 00 00 0
1309 00 00 0
1310 00 00 0

```

```

1321 01 STH 0 ; Alarme 0
1322 09 DYN 420
1323 22 JIZ 1327
1324 11 SEO 400 ; SADD
1325 12 REO 401 } Code pour
1326 12 REO 402 } alarme 0
1327 01 STH 1 ; Alarme 1
1328 09 DYN 421
1329 22 JIZ 1333
1330 11 SEO 400 ; SADD
1331 11 SEO 401 } Code pour
1332 12 REO 402 } alarme 1
1333 01 STH 2 ; Alarme 2
1334 09 DYN 422
1335 22 JIZ 1339
1336 11 SEO 400 ; SADD
1337 12 REO 401 } Code pour
1338 11 SEO 402 } alarme 2
1339 01 STH 3 ; Alarme 3
1340 09 DYN 423
1341 22 JIZ 1345
1342 11 SEO 400 ; SADD
1343 11 SEO 401 } Code pour
1344 11 SEO 402 } alarme 4
1345 01 STH 400 ; SADD = "H"?
1346 22 JIZ 1361

```

```

1351 29 PAS 251 ; FIFO
1352 00 00 400 ; SADD
1353 00 00 411 ; DADD
1354 00 00 3 ; Largeur = 3 bits
1355 00 00 0
1356 00 00 0
1357 00 00 0
1358 00 00 0
1359 00 00 0
1360 00 00 0

```

```

1361 02 STL 409 ; Flag de lec-
1362 05 ORH 425 ture 0 ou TXB
1363 21 JIO 1321
1364 15 SCR 319
1365 00 00 0
1366 01 STH 410 ; DADD-1
1367 15 SCR 319
1368 27 27 10
1369 01 STH 411 ; DADD
1370 15 SCR 319
1371 27 27 20
1372 16 SEI 319
1373 29 PAS 23
1374 00 00 1110 ; Texte 110+C319
1375 12 REO 409 ; Niveau de
; sortie Busy

1381 29 PAS 250 ; Décalage de
1382 00 00 400 ; SADD suite
1383 00 00 411 ; DADD
1384 00 00 3 ; Largeur =
1385 00 00 0 3 bits
1386 00 00 0
1387 00 00 0
1388 00 00 0
1389 00 00 0
1390 00 00 0
1391 20 JMP 1321

```


Vue d'ensemble alphabétique des instructions 2 + 3

Instruction	Chapitre	Page
ADD	I2	4I
CMP	I2	6I
CLA	I3	7I
CLK	I6	14I
DBN, BND	I3	7I
DIV	I2	5I
EWP	I6	15I
EXG	I3	8I
INR, DER	I5	12I
LAC, SAC	I4	11I
LAR, SAR	I4	10I
MUL	I2	5I
NOP	I6	15I
NOP 1111	H1	1H
NOP 1248	I1	1I
PAS 16/17	H2	2H
PAS 19	H2	5H
PAS 23	H2	5H
PAS 24	H2	5H
PAS 50	H2	6H
PAS 54/56	H2	7H
PAS 55/57	H2	12H
PAS 58	H2	16H
PAS 190 (ou PAS 19)	H3	20H
PAS 200...212	H3	28H
PAS 250/251	H3	41H
ROR	I3	8I
ROA	I3	8I
RRE, WRE	I4	12I
RRG, WRG	I4	10I
SEW	I5	13I
SHI	I6	14I
SNC	I5	13I
SQR	I2	6I
SUB	I2	5I
TXT	I6	15I
WEL, WEU	I4	11I

Notes:

SAIA SA

Electronique Industrielle et Composants
3280 Morat/Suisse

Centrale	Téléphone 037 727 111
	Téléfax 037 71 44 43
	Télex 942 127
Vente Suisse	Téléphone 037 727 727
	Téléfax 037 71 19 83

Nos représentations nationales

Belgique	Landis & Gyr Belge SA, Dépt. Industrie Avenue des Anciens Combattants 190, B-1140 Bruxelles ☎ 02 244 02 11, Tx 65 930, Fax 02 242 88 31
Danmark	Skandia-Havemann Vallensbækvej 46, DK-2625 Vallensbæk ☎ 02 64 33 33, Tx 33 383, Fax 02 64 22 45
Deutschland	SAIA GmbH Flinschstrasse 67, D-6000 Frankfurt 60 ☎ 069 42 09 93-0, Ttx 69 99 375, Fax 069 42 56 54
España	Landis & Gyr BC SA Batalla del Salado 25, Apartado 575, 28045 Madrid ☎ 91 467 1900, Tx 22 976, Fax 91 239 44 79
France	SAIA Sàrl 10, Blvd. Louise Michel, F-92230 Gennevilliers ☎ 1 40 86 03 45, Tx 613 189, Fax 1 47 91 40 13
Great Britain	A.S.A.P. Ltd. Unit 15D, Compton Place, Surrey Avenue, Camberley, Surrey GU15 3DX ☎ 0276 691 580, Fax 0276 69 15 81
Italia	SAIA S.r.l. Via Cadamosto 3 20094 Corsico MI
Nederland	Landis & Gyr BV, Div. Electrowater Kampenringweg 45, Postbus 444, NL-2800 AK-Gouda ☎ 01820 65 683, Tx 20 657, Fax 01820 32 437
Norge	Malthe Winje & Co A/S Cort Adlersgt. 14, Postboks 2440, Solli, N-0202 Oslo 2 ☎ 02 55 86 40, Tx 19 629, Fax 02 55 22 11
Österreich COMECON	Landis & Gyr Gesellschaft m.b.H Breitenfurterstrasse 148, Postfach 9, A-1230 Wien ☎ 0222 84 26 26-0, Tx 132 706, Fax 0222 84 26 26 313
Portugal	Infocontrol Electronica e Automatismo LDA. Av. da Igreja No. 68-1º Esq., P-1700 Lisboa ☎ 01 77 51 61-65, Tx 63 454, Fax 01 77 56 87
Suomi Finland	OY Landis & Gyr AB SF-02430 Masala ☎ 8 0297 31, Tx 100 11 53, Fax 8 0297 5531
Sverige	Beving Elektronik AB St. Eriksgatan 113a, Box 21 104, S-10031 Stockholm ☎ 08 15 17 80, Tx 100 40, Fax 08 33 68 63
USA	After sales services: Maxmar Controls Inc. 99 Castleton Street, Pleasantville, New York 10570-3403 ☎ 914 747 3540, Fax 914 747 3567
Australia	Landis & Gyr (Australia) Pty Ltd 411 Ferntree Gully Road, P.O. Box 202, Mount Waverley, Vic. 3149 ☎ 3 544-2322, Tx 32 244, Fax 3 543 74 96
Argentina	Electromedidor S.A.I. y C. Defensa 320, RA-1065 Buenos Aires ☎ 1 337 125, Tx 23 377, Fax 1 33 19 582