



HAL
open science

Étude et réalisation du micro-logiciel du coupleur multi-fonction

Colette Duranti

► **To cite this version:**

Colette Duranti. Étude et réalisation du micro-logiciel du coupleur multi-fonction. Génie logiciel [cs.SE]. 1984. dumas-00312771

HAL Id: dumas-00312771

<https://dumas.ccsd.cnrs.fr/dumas-00312771>

Submitted on 26 Aug 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE AGREE DE GRENOBLE (C.U.E.F.A)



MEMOIRE

présenté en vue d'obtenir

LE DIPLOME D'INGENIEUR DU CONSERVATOIRE
NATIONAL DES ARTS ET METIERS

en

INFORMATIQUE

par

Colette DURANTI



ETUDE ET REALISATION DU MICRO - LOGICIEL
DU COUPLEUR MULTI - FONCTIONS



SOUTENU LE :

JURY

Président : M. RANCHIN

Membres : M. BOLLIET

M. CHERUY

M. CLEAZ

Le travail présenté dans ce mémoire a été réalisé dans le cadre de la formation permanente à BULL/SEMS. Je tiens à remercier la direction locale et le service formation d'ECHIROLLES pour les facilités qu'ils m'ont offertes.

Je remercie très chaleureusement :

Monsieur le Professeur RANCHIN, du Conservatoire National des Arts et Métiers, qui m'a fait l'honneur de présider le jury.

Monsieur le Professeur BOLLIET, de l'Université GRENOBLE II, qui m'a fait l'honneur d'accepter le dossier de thèse.

Monsieur CHERUY, chef de la Division Informatique Industrielle et Monsieur CLEAZ, chef de produits d'application automatismes et contrôle de procédé, pour l'intérêt qu'ils ont toujours manifesté à l'égard de mon travail.

Je veux, enfin, assurer de mon amitié tous mes camarades de l'équipe de développements de systèmes industriels.

S O M M A I R E

	<u>Pages</u>
<u>INTRODUCTION</u>	1
<u>CHAPITRE I : PRESENTATION DU COUPLEUR</u>	3
1. <u>Les objectifs</u>	4
2. <u>Les fonctions reprises et améliorées</u>	7
2.1. L'interface MFI/A	7
2.2. La gestion d'une ligne asynchrone	8
2.3. Les fonctions de télécommande et de téléchargement	9
3. <u>Les fonctions nouvelles</u>	11
3.1. Dans la gestion des lignes asynchrones	11
3.1.1. détection de codes d'arrêt multiples en réception	11
3.1.2. gestion des codes de saturation de "buffer" de périphérique	11
3.1.3. "bufferisation" en réception	12
3.1.4. traitement de procédure de transmission	12
3.1.5. généralisation des fonctions de télé- commande et de téléchargement	13
3.2. Dans la gestion des fonctions temps réel	14
3.2.1. heure et date secourues	14
3.2.2. synchronisation inter-calculateurs	14
3.2.3. 128 temporisations programmables	14
3.2.4. 8 heures de réveil	14
3.2.5. chien de garde calculateur	14
3.2.6. 2 entrées tout ou rien	14
3.2.7. visualisation de l'heure	15

<u>CHAPITRE II : ORGANISATION MATERIELLE DU COUPLEUR</u>	16
1. <u>Organisation générale et choix technologique</u>	18
2. <u>Les interfaces avec l'environnement du calculateur</u>	22
2.1. Les 2 lignes asynchrones	22
2.2. Le chien de garde et les entrées tout ou rien	23
2.3. Le bandeau de visualisation	24
2.4. Les signaux du pupitre de commande et du coupleur X25 SOLAR	27
3. <u>L'interface du CMF avec le bus d'entrées-sorties du SOLAR</u>	28
3.1. La fonction de gestion des lignes asynchrones	29
3.2. La fonction pupitre	31
3.3. La fonction horloge temps réel	31
3.4. Les fonctions temps réel	31
4. <u>L'environnement des 2 micro-processeurs</u>	33
4.1. Les paramètres du système	34
4.1.1. pour la fonction de gestion des 2 lignes asynchrones	34
4.1.2. pour les fonctions de téléchargement et de télécommande	35
4.1.3. pour les fonctions pupitre	35
4.2. Les états du système	35
4.2.1. l'état "POP"	35
4.2.2. l'état "TIP"	36
4.3. Les événements à prendre en compte	37
4.3.1. par le micro-processeur de gestion des fonctions de transmission	37
4.3.2. par le micro-processeur de gestion des fonctions temps réel	38
4.4. Les actions	39
4.4.1. pour la fonction téléinfo	39
4.4.2. pour les fonctions pupitre	39
4.4.3. pour la fonction de téléchargement	40
4.4.4. dans le cadre des fonctions temps réel	40

	<u>Pages</u>
5. <u>Découpage de l'adressage mémoire</u>	41
5.1. Espace d'adressage du microprocesseur de gestion des fonctions de transmission	42
5.2. Espace d'adressage du microprocesseur de gestion des fonctions temps réel	44
<u>CHAPITRE III : LA GESTION DES LIGNES ASYNCHRONES</u>	45
1. <u>Le "driver"</u>	47
1.1. La fonction télérinfo	48
1.1.1. description des échanges émission et réception en mode programmé simple	48
1.1.2. description des échanges émission et réception en mode canal, sans traitement d'informations	52
1.1.3. réception d'une donnée avec défaut	58
1.1.4. réception avec "bufferisation"	60
1.1.5. la gestion des signaux modem	64
1.1.6. les chiens de garde logiciels	68
1.2. Les traitements d'informations	70
1.2.1. les codes d'arrêt	70
1.2.2. les codes spécifiques	71
1.2.3. les fonctions de téléchargement et de télécommande	72
1.2.4. la modification des paramètres du coupleur	75
2. <u>L'interface avec la procédure de l'utilisateur</u>	77
2.1. Les services offerts par le "driver"	78
2.2. Le dialogue entre la procédure et le "driver"	82
2.3. Description des blocs de contrôle	86
2.4. Traitements particuliers	92

	<u>Pages</u>
<u>CHAPITRE IV : LES FONCTIONS PUPITRE</u>	94
1. <u>Le dialogue</u>	98
2. <u>Les fonctions proposées</u>	100
3. <u>La prise en compte des informations reçues sur la ligne</u>	102
4. <u>Les chemins de données et les interfaces</u>	104
4.1. Interface entre la console de service et le micro-processeur	104
4.2. Interface entre le micro-processeur et le processeur SOLAR	104
4.3. Récapitulatif	105
<u>CHAPITRE V : LES FONCTIONS TEMPS REEL</u>	108
1. <u>Le chien de garde calculateur et les entrées tout ou rien</u>	109
1.1. L'initialisation du chien de garde	113
1.2. Les commandes d'armement et de désarmement	113
2. <u>Le calendrier secouru</u>	114
2.1. L'heure secourue	114
2.2. L'heure et la date maintenues	115
2.3. L'initialisation de la date et de l'heure	116
2.4. Gestion de la date et de l'heure par le logiciel SOLAR, en cas de coupure secteur	117
2.5. Le chargement de la batterie	117
3. <u>Le signal de synchronisation</u>	119
3.1. Le principe de fonctionnement	121
3.2. La réalisation	122
4. <u>Le bandeau de visualisation</u>	124
5. <u>L'horloge temps réel</u>	126

	<u>Pages</u>
6. <u>Les temporisations programmables</u>	127
6.1. L'initialisation d'une ou plusieurs temporisations	128
6.2. Les commandes d'armement et de désarmement d'une temporisation	130
7. <u>Les heures de réveil</u>	133
7.1. L'initialisation des heures de réveil	133
7.2. Traitement des heures de réveil	134
<u>CHAPITRE VI : UTILISATION DU CMF DANS LES APPLICATIONS INDUSTRIELLES</u>	136
1. <u>Les configurations autonomes</u>	137
1.1. Sans coupleur CMF	137
1.2. Avec coupleur CMF	137
2. <u>Les configurations hiérarchisées</u>	140
2.1. Sans coupleur CMF	140
2.2. Avec coupleur CMF	141
<u>CONCLUSION</u>	144

GLOSSAIRE

- Buffer : zone mémoire de stockage intermédiaire
- Bufferisation : mémorisation intermédiaire des informations reçues sur la ligne
- CMF : coupleur multi-fonctions
- FIFO : mémorisation avec vidage du 1er élément stocké (first in first out)
- DRIVER : module logiciel qui gère de façon spécifique les entrées sorties du coupleur

INTRODUCTION

Réduire les coûts de mise en oeuvre et d'exploitation, améliorer la sécurité de fonctionnement sont aujourd'hui les objectifs prioritaires pour la conduite des procédés de fabrication et d'une façon plus générale pour la gestion des installations techniques des secteurs industriels :

- chimie, pétrochimie, plastiques.
- métallurgie, sidérurgie.
- industries alimentaires.
- production et distribution d'énergie.

Afin de conforter sa position de fournisseur de systèmes informatiques d'automatisation, SEMS propose une nouvelle gamme de produits SOLAR qui intègre largement les technologies nouvelles. Parmi les composants de cette gamme, SEMS développe une carte regroupant toutes les fonctions de base que doit offrir un système compact d'automatisation. Cette carte, le coupleur multi-fonctions, associée à une seconde carte supportant le processeur et sa mémoire constitue le nouveau calculateur industriel. Une telle configuration offre à l'utilisateur des performances accrues, une fiabilité meilleure, pour un coût sensiblement identique à l'ancienne génération.

Le coupleur multi-fonctions est organisé autour de deux micro-processeurs qui permettent de décharger l'unité centrale : ils assurent une partie des fonctions habituellement réalisées par les logiciels d'application :

- gestion de tout ou partie d'une procédure de transmission.
- amélioration des performances du SOLAR en gestion d'entrées - sorties, par la prise en compte, en réception, de codes d'arrêt multiples ou des codes spécifiques de certains périphériques.
- gestion des fonctions temps réel, indispensables en environnement industriel telles que gestion de temporisations ou d'heures de réveil pour l'activation de tâches périodiques de l'application.

Le coupleur multi-fonctions augmente également, la sécurité du système en offrant à l'utilisateur :

- un calendrier secouru en cas de coupure secteur.

- un chien de garde calculateur et des entrées tout ou rien, outils de surveillance de l'état de bon fonctionnement du calculateur et de son environnement.

- une synchronisation intercalculateurs pour assurer une base de temps identique à tous les calculateurs d'une structure hiérarchisée.

Ce produit a été développé par l'équipe "matériel" de la division "automatismes industriels" ; mon travail, au sein de cette équipe a consisté à étudier et à développer le logiciel des deux micro-processeurs implantés sur le CMF *.

Le travail réalisé a subi des contraintes :

- techniques liées à une gamme existante et nécessitant le respect de la compatibilité avec le matériel et les logiciels existants ;
- de planning de développement et de prix de vente du produit final.

Le présent mémoire a pour but de présenter ce travail en le situant dans son cadre global.

Nous trouverons :

Dans le chapitre I, la présentation des objectifs du CMF.

Dans le chapitre II, la présentation de l'organisation matérielle du coupleur.

Dans les chapitres III, IV et V, la présentation du logiciel intégré au CMF, sujet du travail réalisé.

Dans le chapitre VI, enfin, l'utilisation du CMF, dans 2 types de configuration, où nous nous attacherons à montrer ce qu'il offre, en plus, du point de vue performances, à l'utilisateur.

* coupleur multi-fonctions

CHAPITRE I

PRESENTATION DU COUPLEUR

I. PRESENTATION DU COUPLEUR

1. <u>Les objectifs</u>	4
2. <u>Les fonctions reprises et améliorées</u>	7
2-1. l'interface MFI/A	7
2-2. la gestion d'une ligne asynchrone	8
2-3. les fonctions de télécommande et de téléchargement.	9
3. <u>Les fonctions nouvelles</u>	11
3-1. <u>Au niveau de la gestion des lignes asynchrones.</u>	11
3-1-1. détection de codes d'arrêt multiples en réception.	11
3-1-2. gestion des codes de saturation et désaturation.	11
3-1-3. "bufferisation" en réception.	12
3-1-4. traitement de procédure de transmission.	12
3-1-5. généralisation des fonctions de télécommande et de téléchargement.	13
3-2. <u>Au niveau des fonctions temps réel</u>	14
3-2-1. heure et date secourues.	14
3-2-2. synchronisation inter-calculateurs.	14
3-2-3. 128 compteurs programmables.	14
3-2-4. 8 heures de réveil.	14
3-2-5. un chien de garde calculateur.	14
3-2-6. 2 entrées tout ou rien.	14
3-2-7. le bandeau de visualisation.	15

1. LES OBJECTIFS

Le coupleur multi-fonctions offre, sous une forme compacte, l'ensemble des fonctions de base nécessaires à un mini-système informatique dans le cadre d'applications temps réel, industrielles et scientifiques.

Les objectifs de ce coupleur sont les suivants :

a. Conforter la compétitivité du bas de gamme SOLAR :

* Le développement de ce coupleur a été réalisé en parallèle avec celui d'un mono-processeur avec mémoire incorporée (le processeur 16/30).

* Le CMF = remplace le coupleur de base informatique appelé MFI/A (gestion d'un pupitre de mise au point et de maintenance, de la console de service, de l'horloge temps réel) sur toutes les configurations SOLAR, en améliorant ses fonctionnalités.

= apporte des compléments fonctionnels favorisant l'utilisation des unités centrales SOLAR en qualité de frontal téléchargé dans les domaines des automatismes , du contrôle de procédé et de l'instrumentation ; ces fonctions de téléchargement et de télécommande étaient assurées par le coupleur TIP.

Nous pouvons ainsi proposer un mini-système, alternative aux micro-systèmes, en deux cartes :

- le mono-processeur 16/30 et
- le coupleur CMF

b. Assurer la compatibilité avec les logiciels existants pour que la transparence soit effective au niveau système d'exploitation, afin d'éviter les problèmes d'intégration du CMF dans le parc existant.

c. Proposer des fonctions supplémentaires déchargeant le processeur SOLAR et augmentant la sécurité du système telles que :

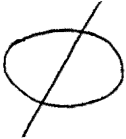
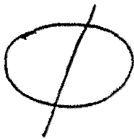
- * gestion de procédure de transmission asynchrone,
- * gestion de 128 temporisations programmables,
- * fonction chien de garde calculateur (traitée actuellement par l'adjonction d'une carte supplémentaire disponible en matériel spécifique).

- * gestion d'un calendrier (date et heure secourues)
- * télémaintenance
- * ...

Le développement de ce coupleur a été soumis à deux impératifs :

- assurer la compatibilité avec les logiciels existants,
- respecter un objectif de prix de vente très compétitif.

Tableau I-1 : Comparaison des services offerts avec ou sans CMF

Services offerts par le SOLAR SANS CMF	Services offerts par le CMF (carte format 1/1)
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Coupleur MFI/A (carte format 1/2)</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Pupitre POP</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Coupleur TIP (carte format 1/2)</div> <div style="border: 1px solid black; padding: 5px;">Coupleur ASX (carte format 1/2)</div>	<ul style="list-style-type: none"> - Gestion de la console de service sur ligne asynchrone. - Gestion d'une horloge temps réel. - Fonctions pupitre. - Fonctions de télécommande et de téléchargement. - Gestion d'une seconde ligne asynchrone.
	<p><u>Au niveau des 2 lignes asynchrones</u></p> <ul style="list-style-type: none"> - détection de codes d'arrêt multiples. - détection de caractères spéciaux - "bufferisation" en réception. - gestion de procédure de transmission. - généralisation des fonctions de téléchargement et de télécommande.
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Coupleur SP31 (matériel spécifique - carte format 1/2)</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Coupleur ITI 16 (matériel spécifique - carte format 1/2)</div> <div style="border: 1px solid black; padding: 5px;">Coupleur GPI 24 (sur lequel est con- necté une horloge externe - carte format 1/2)</div>	<ul style="list-style-type: none"> - Chien de garde calculateur. - Entrées tout ou rien. - Heure et date secourues.
	<ul style="list-style-type: none"> - synchronisation inter-calculateurs. - visualisation de l'heure. - heures de réveil. - temporisations programmables.

2. LES FONCTIONS REPRISES ET AMELIOREES

Proposer un mini-système informatique en 2 cartes, dans le cadre d'applications industrielles ou scientifiques suppose une bonne connaissance des besoins des utilisateurs pour reprendre et améliorer les fonctions proposées au niveau des systèmes SOLAR. Sur le CMF, sont ainsi intégrées les fonctionnalités simplifiées mais améliorées de plusieurs coupleurs SOLAR, en respectant l'objectif de compatibilité avec les logiciels existants pour en éviter toute modification ; ces fonctions sont des fonctions fondamentales qu'on retrouve dans toutes les applications industrielles.

Sont ainsi reprises les fonctions des coupleurs :

- de base informatique ou MFI/A
- de gestion de ligne asynchrone ou ASX
- de téléchargement ou TIP.

2-1. L'interface MFI/A

Le coupleur CMF est destiné à remplacer le coupleur de base informatique appelé MFI/A sur toutes les configurations SOLAR. L'interface MFI regroupe 3 fonctions très différentes :

- la fonction de gestion de la console de service en interface isolée série asynchrone dont la vitesse de transmission peut varier de 110 à 4 800 bauds.
- la fonction d'horloge temps réel qui délivre au SOLAR une horloge de période configurable, à choisir dans la gamme de 1 μ s à 20 ms.
- la fonction de gestion du pupitre opérateur (ou POP) qui permet à un opérateur d'accéder aux registres internes de la machine (A, B, X, Y, C, L, W, K, P, S, SLO, SLE, IM, HV) ainsi qu'à toutes les adresses mémoires présentes dans un système. L'opérateur peut également définir des points d'arrêt programme sur des instructions ou sur des opérands. Ce module permet également la visualisation, par programme, d'informations intéressantes telles que la charge UC ...

Le module POP regroupe 3 éléments :

- le pupitre opérateur monté sur la face avant de l'armoire SOLAR et

composé de touches de commande et de voyants de visualisation,

- la partie de raccordement au bus d'entrées - sorties SOLAR,
- la micro-programmation spécifique implantée sur le processeur SOLAR et qui permet au processeur de dialoguer avec le pupitre et de réaliser les fonctions pupitre.

Au niveau du coupleur multi-fonctions :

=> la ligne asynchrone sur laquelle est connectée la console de service est une ligne "full-Duplex" , avec trois possibilités de connexion : simple courant, V24 simplifié et RS 422. La vitesse de transmission peut varier de 50 bauds à 38 400 bauds.

D'autre part, les échanges sont réalisés en mode canal : le logiciel est déchargé des phases d'émission/réception exécutées par la micro-machine du SOLAR. (Avec l'interface MFI, le logiciel de gestion de la console de service traite les données, caractère par caractère car le coupleur MFI travaille en mode programmé prioritaire).

=> l'horloge temps réel ne gère que la période la plus couramment utilisée par les logiciels SOLAR : celle de 20 ms.

=> la fonction POP n'est plus réalisée à partir du pupitre opérateur - qui disparaît de la configuration- mais par conversationnel sur un organe de dialogue connecté à l'une des deux lignes asynchrones ; ceci améliore l'opérabilité du système. En effet, le pupitre POP est un outil de travail qui permet de réaliser seulement des fonctions de très bas niveau et il s'est avéré indispensable d'offrir à l'utilisateur des moyens de mise au point et de maintenance plus sophistiqués.

2-2. La gestion d'une ligne asynchrone

En plus de la ligne asynchrone réservée à la console de service, le CMF propose une seconde ligne asynchrone avec quatre possibilité de liaison : simple courant, V24 simplifié, V24 complet et RS422.

Cette fonction remplace le coupleur asynchrone une voie, existant actuellement sur SOLAR, et permet le raccordement d'un processeur à un modem de

transmission asynchrone connecté au réseau téléphonique commuté.

Le logiciel d'application est déchargé de la gestion des signaux modem qui est assurée par le micro-logiciel du CMF.

La fonction "POP" est activable à partir de la ligne sur laquelle est connectée la console de service mais aussi sur cette deuxième ligne asynchrone qui, par son interface V24 complète permet d'envisager la télé-maintenance à partir du site constructeur.

Les deux lignes asynchrones sont full-duplex et travaillent

- soit en programmé simple
- soit en canal LDC (canal d'entrées - sorties conçu pour transmettre des données en provenance ou à destination de la mémoire centrale).

2-3. Les fonctions de télécommande et de téléchargement

Elles remplacent le module TIP qui permet :

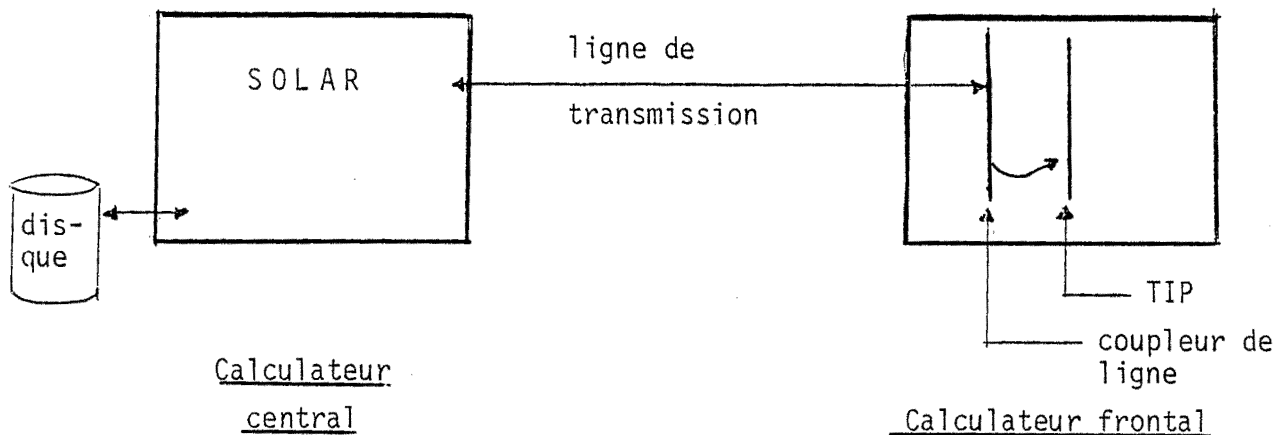
= de recevoir sur une ligne synchrone ou asynchrone des messages réalisant, à partir d'un site distant, les fonctions du pupitre de commande du SOLAR : arrêt ou marche programme (STOP/RUN), initialisation du calculateur (INI), chargement du bootstrap en mémoire vive (LOAD).

= de mettre en marche ou d'arrêter un processeur dans une structure multiprocesseur.

= d'accéder à une mémoire morte, implantée sur le module, définie par le client, contenant la procédure de démarrage du calculateur, en particulier, un programme de chargement de ligne.

Utilisation du module TIP

Soit la configuration représentée par le schéma ci-après :



Le calculateur central est équipé de périphériques lourds et plus particulièrement d'une unité de disque ; par contre, le calculateur frontal est dépourvu de mémoire de masse et c'est un frontal "vide".

Le problème consiste à charger un programme dans la mémoire du frontal, via la ligne de transmission, à partir du calculateur central : c'est l'opération de TELECHARGEMENT.

Un programme de réception est implanté dans des mémoires mortes du TIP. Sur réception de la commande "LOAD", émise sur la ligne par le central, le module TIP du frontal charge en mémoire vive le programme de réception et initialise le pointeur instruction du SOLAR. La commande "RUN" permet ensuite d'exécuter ce programme qui charge dans la mémoire du frontal les informations provenant du central et constituant le programme d'application (ou système) du frontal. Lorsque cette opération est terminée, le logiciel téléchargé est activé.

Le coupleur multi-fonctions reprend ces fonctionnalités mais avec des simplifications et des améliorations :

= la fonction de téléchargement est réalisée dans la partie réception des deux lignes asynchrones - la gestion de cette fonction sur ligne synchrone est assurée dans le cas HDLC x 25 (cf III - 1 - 2 - 3)

= le programme de réception du frontal est implanté dans la mémoire morte du CMF : il n'y a plus de PROM spécifique au téléchargement ; cela élimine tous les composants réalisant le décodage de son adresse et de son interface avec le bus d'entrées - sorties du SOLAR.

3. LES FONCTIONS NOUVELLES

Elles sont liées à deux impératifs au niveau des systèmes industriels :

- décharger l'unité centrale des traitements coûteux en temps d'exécution.
- augmenter la sécurité du système. Leur utilisation nécessite une prise en compte explicite dans les logiciels système.

3.1. Dans la gestion des lignes asynchrones

3.1.1. Détection de codes d'arrêt multiples en réception

Les canaux micro-programmés du SOLAR permettent, pour un échange donné en réception, la détection d'un seul code d'arrêt. Un logiciel qui doit reconnaître d'autres codes d'arrêt ne peut pas travailler en mode canal car il doit analyser chaque caractère ; de ce fait, il travaille en mode réception caractère par caractère, d'où une perte de temps importante au niveau de l'unité centrale. Le CMF peut détecter sur la voie réception de chaque ligne, de 1 à N ($N < 10$) codes d'arrêt différents et permet au logiciel, déchargé de la recherche du code d'arrêt, de fonctionner en mode canal.

3.1.2. Gestion des codes de saturation de "buffer" de périphérique

Certains périphériques (imprimantes, visu graphiques, semi-graphiques...) possèdent un "buffer" de réception : le calculateur émet les données à la vitesse de la ligne, dans les limites de saturation du "buffer", et le périphérique visualise ou imprime les données à sa vitesse propre.

En cas de saturation ou de désaturation du "buffer", le périphérique asservit le calculateur en lui envoyant 2 caractères spéciaux débanalisés : les codes de saturation et de désaturation du "buffer". Pour gérer ce type de périphérique, le logiciel SOLAR travaille, comme dans le cas des codes d'arrêt multiples, en mode réception caractère par caractère avec analyse de chaque caractère. Cette fonction, étant maintenant déportée au niveau du CMF, le logiciel est déchargé du traitement de ces codes qui ne lui sont plus fournis car ils sont analysés et traités par le micro-logiciel du CMF.

3.1.3. "Bufférisation" en réception

Afin d'être utilisée par des logiciels existants, cette fonction n'est pas validée par le logiciel mais par micro-interrupteur sur le coupleur. Le micro-logiciel se charge de mémoriser tout message avant que le logiciel ait initialisé sa voie réception.

Cette fonction permet :

- d'éviter les classiques erreurs de cadence en réception sur les périphériques peu prioritaires en cas de surcharge d'échanges sur une configuration sous-calibrée en puissance d'entrées-sorties.

- de masquer, dans le cas particulier de dialogue entre 2 calculateurs de performances inégales, les problèmes de retournement émission sur réception d'un "driver" lent à s'exécuter.

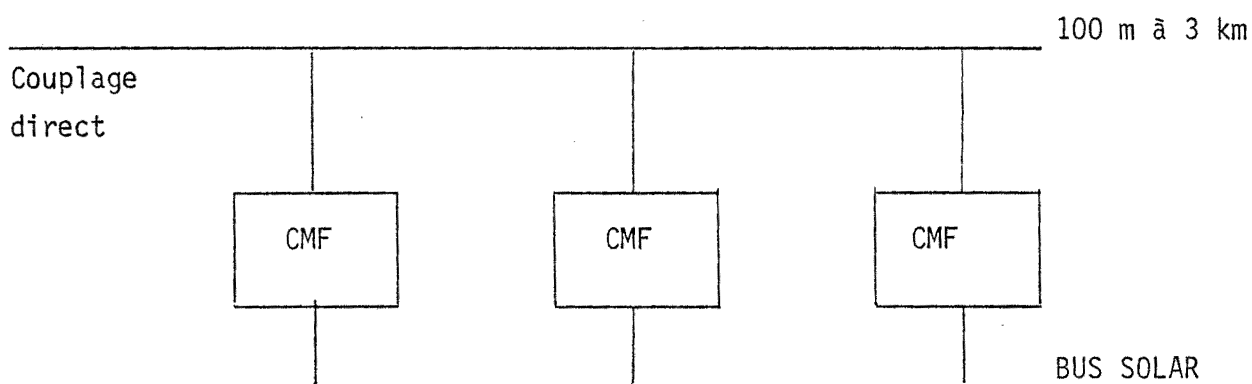
3.1.4. Traitement de procédure de transmission

Des traitements plus complexes des informations émises et reçues sont possibles sur le coupleur. L'utilisateur peut envisager de déporter, dans une mémoire morte (du type EPROM) réservée à cet effet, un driver ou une partie d'un logiciel de traitement de procédure de transmission. Un certain nombre de mécanismes matériels et micro-logiciels ont été introduits dans l'interface matériel/logiciel du coupleur pour permettre cette décentralisation. Cela permet d'offrir aux utilisateurs une structure d'accueil pour un couplage de leur installation aux réseaux industriels. En effet, les réseaux locaux industriels, en cours d'étude chez les industriels fournisseurs de matériel informatique, imposent la réalisation de protocoles de communication plus ou moins importants en taille de programme et en complexité selon le degré de sophistication du réseau.

Le CMF offre une structure d'accueil pouvant recevoir plusieurs types de protocoles de communication.

EXEMPLE

Réseau industriel



Medium : paire torsadée blindée

Niveau électrique : interface RS 422

Dans ce cas, les fréquences de transmission sont fonction de la distance couverte par le réseau.

3.1.5. Généralisation des fonctions de télécommande et de téléchargement dans des environnements X25 ou TRANSPAC

- commande à distance X25

Dans une liaison point à point X25 entre deux SOLAR, un coupleur X25 ayant reçu une trame de télécommande, téléchargement, a la possibilité par l'intermédiaire du coupleur multi-fonctions de réinitialiser le SOLAR et de recharger l'application à partir d'un périphérique "bootstrapable" quelconque. C'est le coupleur multi-fonctions qui se charge d'exécuter les commandes "ARRET", "INI", "LOAD", "RUN" qui lancent l'exécution de la séquence d'initialisation du coupleur X25.

- Mise sous tension d'un SOLAR relié à un réseau TRANSPAC

Dans certaines configurations, un SOLAR relié au réseau TRANSPAC doit pouvoir, après mise sous tension, démarrer normalement et se connecter au réseau sans l'intervention d'un opérateur. A la mise sous tension, si cette fonction est activée par un micro-interrupteur sur le coupleur, le CMF exécute les commandes "ARRET", "INI", "LOAD", "RUN" avec chargement en mémoire vive du SOLAR du contenu d'une mémoire morte implantée sur le CMF et contenant un chargeur de système à travers le réseau.

3.2. Dans la gestion des fonctions temps réel

Les fonctions "temps réel" intégrées au coupleur multi-fonctions sont indispensables dans tout environnement industriel et déchargent le logiciel d'application de traitement coûteux en temps d'exécution.

3.2.1. Heure et date secourues pendant 72 heures, au maximum, en cas de coupure secteur. Le CMF offre la date en jour, mois, année et l'heure en " top " de 20 ms ou de 1 ms.

3.2.2. Une synchronisation automatique inter-calculateurs limite la dérive de l'heure, d'un calculateur à l'autre, à + ou - 5 ms.

3.2.3. 128 temporisations programmables en durée, en répétitivité et en périodicité. Chaque temporisation est une durée absolue dont l'échéance réveille une tâche du logiciel d'application.

3.2.4. 8 heures de réveil qui sont des échéances relatives à l'heure courante et qui servent à réveiller des tâches du logiciel d'application.

3.2.5. Un chien de garde calculateur

Cette fonction est destinée à prévenir l'environnement du calculateur qu'il n'est plus en état d'assurer sa mission. La fonction consiste en un compteur de temps programmable que l'application doit réarmer périodiquement dans un temps inférieur à la période choisie. Si tel n'est pas le cas, un relais est activé pour réaliser des commandes extérieures.

3.2.6. 2 entrées tout ou rien qui permettent de signaler au logiciel d'applications l'arrivée d'évènements extérieurs (tel le déclenchement du chien de garde d'un autre calculateur). Les outils de base (chien de garde et entrée TOR) pour construire des architectures redondantes de calculateurs SOLAR sont ainsi disponibles.

3.2.7. Possibilité d'afficher sur un bandeau de visualisation

- l'heure (en heures, minutes, secondes)
- la charge de traitement de l'unité centrale
- et l'état du relais du chien de garde (il précise l'état de bon ou de mauvais fonctionnement du calculateur).

CHAPITRE II

ORGANISATION MATERIELLE DU COUPLEUR

II. <u>ORGANISATION MATERIELLE DU COUPLEUR</u>	16
1. <u>ORGANISATION GENERALE ET CHOIX TECHNOLOGIQUE</u>	18
2. <u>LES INTERFACES AVEC L'ENVIRONNEMENT DU CALCULATEUR</u>	22
2.1. Les 2 lignes asynchrones	22
2.2. Le chien de garde et les entrées tout ou rien	23
2.3. Le bandeau de visualisation	24
2.4. Les signaux du pupitre de commande et du coupleur X25 SOLAR	27
3. <u>L'INTERFACE AVEC LE BUS D'ENTREES-SORTIES DU SOLAR</u>	28
3.1. La fonction de gestion des 2 lignes asynchrones	29
3.2. La fonction pupitre	31
3.3. La fonction horloge temps réel	31
3.4. La fonction temps réel	31
4. <u>L'ENVIRONNEMENT DES 2 MICRO-PROCESSEURS</u>	33
4.1. Les paramètres du système	34
4.1.1. Pour la gestion des 2 lignes asynchrones	34
4.1.2. Pour les fonctions de téléchargement et de télécommande	35
4.1.3. Pour les fonctions pupitre	35
4.2. Les états du système	35
4.3. Les événements à prendre en compte	37
4.3.1. Par le micro-processeur de gestion des fonctions de transmission	37
4.3.2. Par le micro-processeur temps réel	38
4.4. Les actions	39
4.4.1. Pour la fonction téléinfo	39
4.4.2. Pour les fonctions pupitre	39
4.4.3. Pour la fonction de téléchargement	40
4.4.4. Pour les fonctions temps réel	40

5. LE DECOUPAGE DE L'ESPACE D'ADRESSAGE

PAGES

5.1. Du micro-processeur gérant les fonctions de transmission

42

5.2. Du micro-processeur gérant les fonctions temps réel.

44

1. ORGANISATION GENERALE ET CHOIX TECHNOLOGIQUE

Du point de vue matériel, le coupleur se compose de 4 parties :

a) Les interfaces du coupleur avec l'environnement du calculateur :

- les 2 lignes asynchrones
- le chien de garde calculateur
- les 2 entrées tout ou rien
- le bandeau de visualisation de l'heure
- le pupitre de commande du SOLAR
- les signaux de télécommande du coupleur X25 SOLAR

b) L'interface du coupleur avec le bus d'entrées-sorties du SOLAR

c) La partie micro-processeur de gestion des fonctions temps réel

d) La partie micro-processeur de gestion des fonctions de transmission. Ce micro-processeur gère les 2 lignes asynchrones, les fonctions pupitre et les fonctions de téléchargement et de télécommande.

Nous avons choisi la solution à 2 micro-processeurs sans aucune fenêtre de visibilité, de l'un à l'autre, pour deux raisons :

- l'indépendance des 2 fonctions c et d, entre elles,
- la charge de traitement de chaque micro-processeur

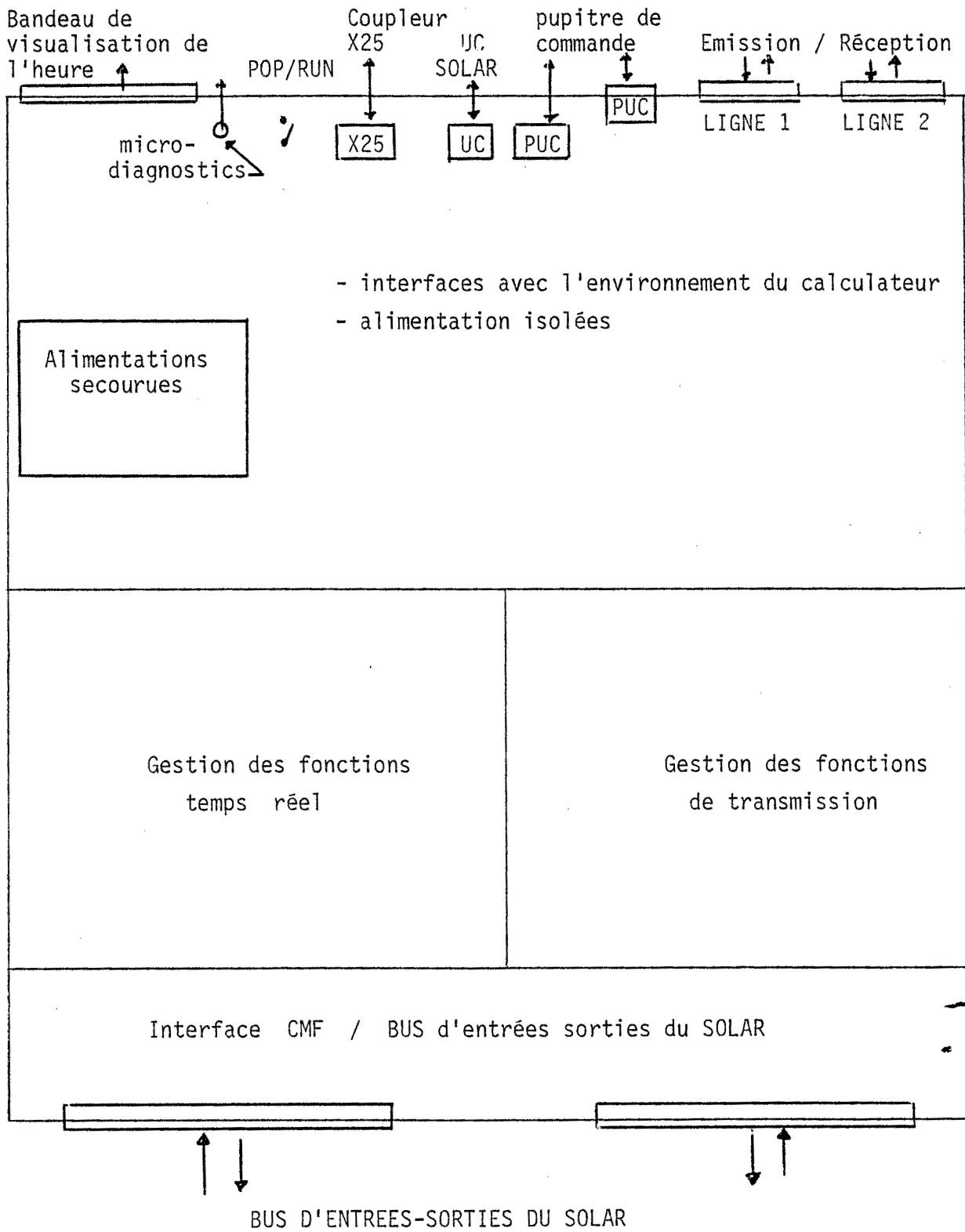


FIGURE 2-1 : Organisation matérielle du CMF

Le coupleur CMF, développé à base de composants discrets, est une carte format 1/1 SOLAR rallongée :

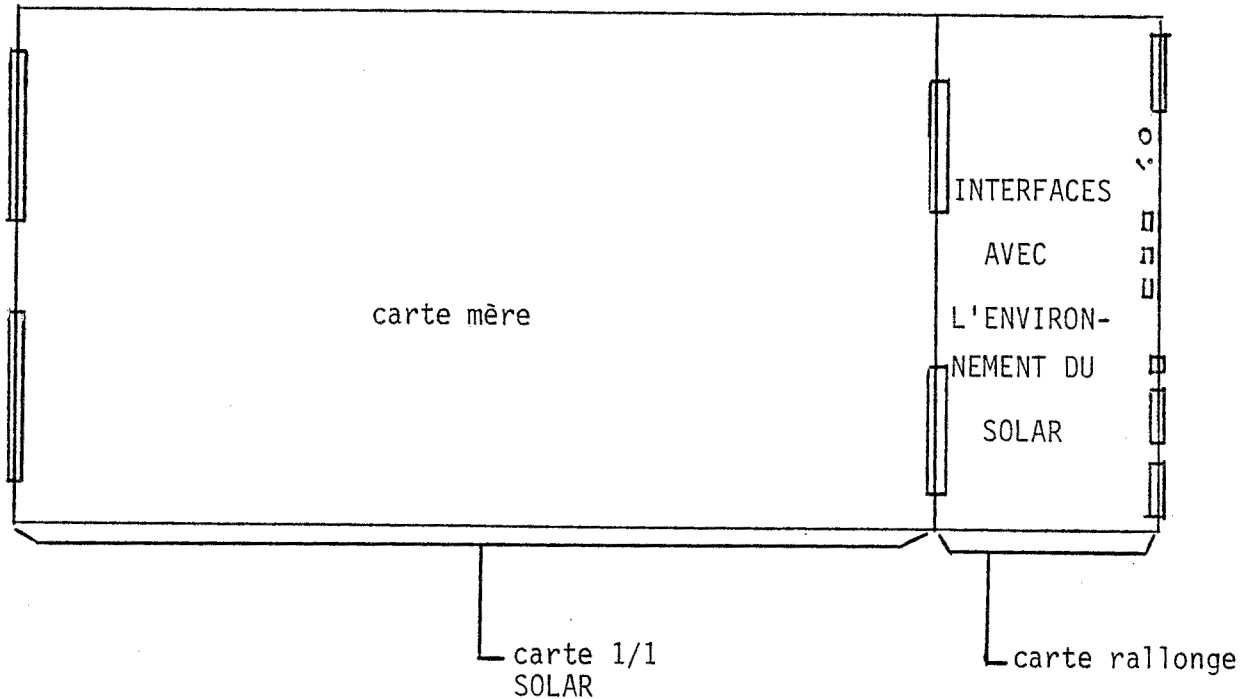


FIGURE 2.2 : coupleur CMF en version composants discrets

Pour ramener la carte au format standard 1/1 SOLAR, nous avons regroupé un certain nombre de fonctions dans un réseau prédiffusé.

La réalisation à base d'un réseau prédiffusé demande un investissement très important mais nous l'avons retenue pour plusieurs raisons :

- des constructeurs tels que NS ou MOTOROLA savent intégrer, actuellement, dans un boîtier prédiffusé 2400 portes (soit 120 boîtiers environ); c'est un nouveau pas technologique entre le boîtier de base et le VLSI (intégration globale) qui permet de personnaliser ses réalisations. Nous ramenons ainsi la dimension du CMF au format 1/1 SOLAR.

- le gain au niveau de la consommation et de la dissipation thermique du coupleur est appréciable .

- le circuit est testé à sa conception par le fabricant, avec des algorithmes de simulation . La testabilité du coupleur en est fortement simplifiée.

- l'amortissement de l'investissement de base est couvert par un objectif de vente de l'ordre de 2000 unités. Dans le domaine industriel, les choix faits sont certes des choix techniques mais sont surtout des choix liés aux réalités économiques.

- le réseau prédiffusé englobe l'interface avec le bus d'entrées-sorties SOLAR et l'interface avec le micro-processeur de gestion des fonctions de transmission ; cela nous permet de porter l'architecture prévue pour la gestion des lignes asynchrones du CMF sur un coupleur multiplexé intelligent, le MUX4, dont le développement est une suite logique de celui du CMF. Le MUX4 est un coupleur format 1/2 qui offre à l'utilisateur 4 lignes asynchrones gérant toutes les fonctions traitées sur les 2 lignes asynchrones du CMF.

2. LES INTERFACES AVEC L'ENVIRONNEMENT DU CALCULATEUR

2.1. Les 2 lignes asynchrones sont munies, chacune, d'interfaces :

- simple courant,
- RS422,
- V24 simplifiée

La ligne 2 est, de plus, équipée d'une interface V24 complète.

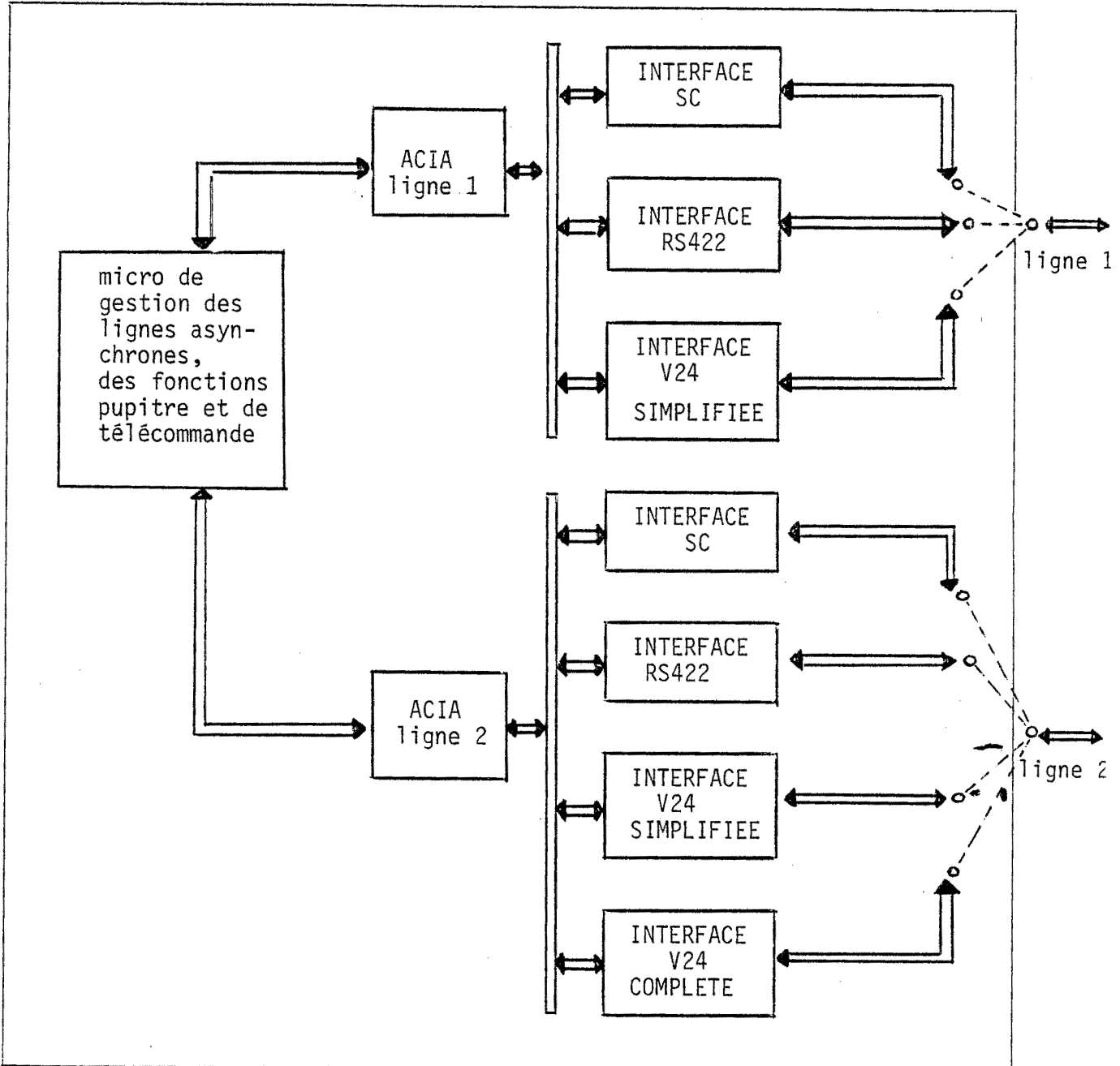


FIGURE 2-3 : Interface lignes

Pour dialoguer avec les lignes, le micro-processeur de gestion des fonctions de transmission dispose, par ligne, des 4 registres de l'ACIA :

2.1. a) le registre de commande par l'intermédiaire duquel :

- il initialise l'interface
- il valide l'émission ou la réception
 - . avec ou sans interruptions,
 - . avec précision du format des données (nombre de bits d'informations, nombre de bits d'arrêt et parité)
 - . avec gestion de la commande DPE (demande pour émettre).

2.1. b) le registre d'état dans lequel il trouve l'état de la voie émission ou de la voie réception :

- une donnée a été reçue avec ou sans erreur,
- une donnée peut être émise car le registre d'émission est vide
- les signaux modem PAE (prêt à émettre) et DP (détection de porteuse) sont ou ne sont pas positionnés.

2.1. c) le registre de sortie informations dans lequel il charge la donnée à émettre.

2.1. d) le registre d'entrée informations où il peut lire la donnée reçue.

2.2. le chien de garde et les entrées tout ou rien

Le chien de garde a pour rôle de prévenir l'environnement du calculateur que l'équipement est en défaut ou hors tension. C'est un compteur de temps programmable, géré par le micro-processeur temps réel, que l'application doit réarmer périodiquement dans un temps inférieur à la période choisie ; si tel n'est pas le cas, le chien de garde est activé, par le micro-logiciel.

Les 2 entrées TOR permettent de signaler au logiciel d'application l'arrivée d'événements extérieurs, tels que le chien de garde d'un autre calculateur.

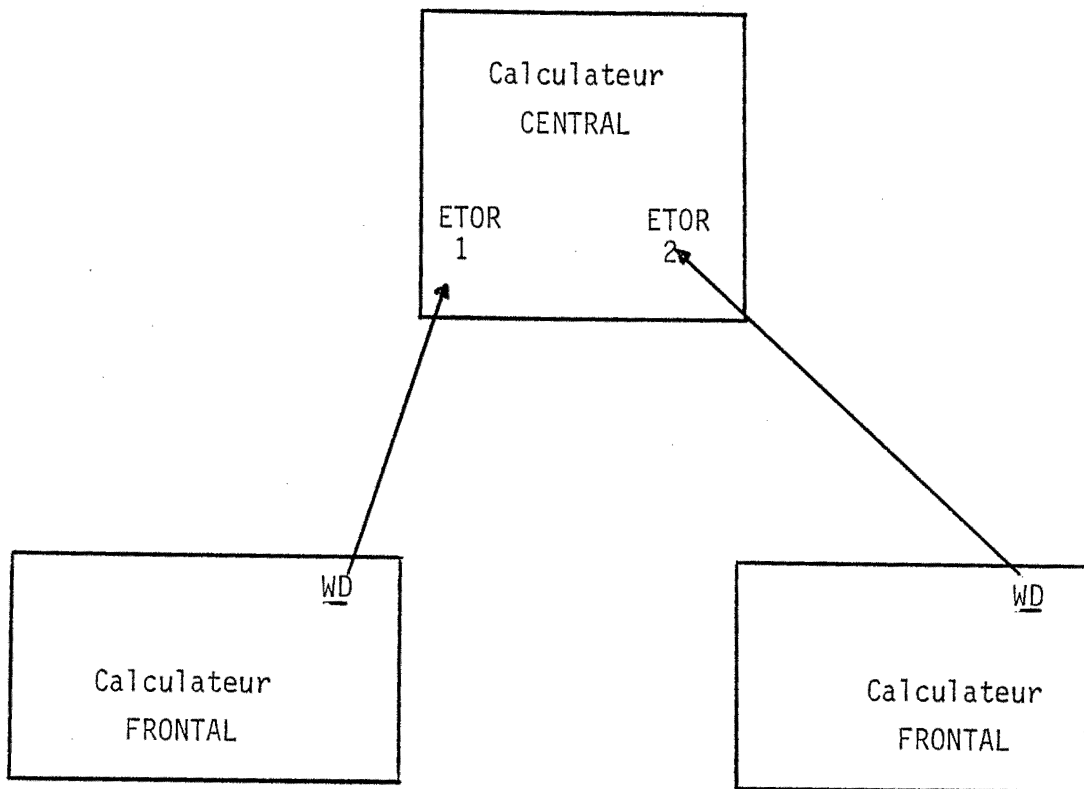


Figure 2.4 : exemple d'utilisation du chien de garde et des Entrées TOR

Le chien de garde est commandé par le micro-processeur temps réel. Les entrées TOR sont interfacées directement avec le bus d'entrées-sorties du SOLAR.

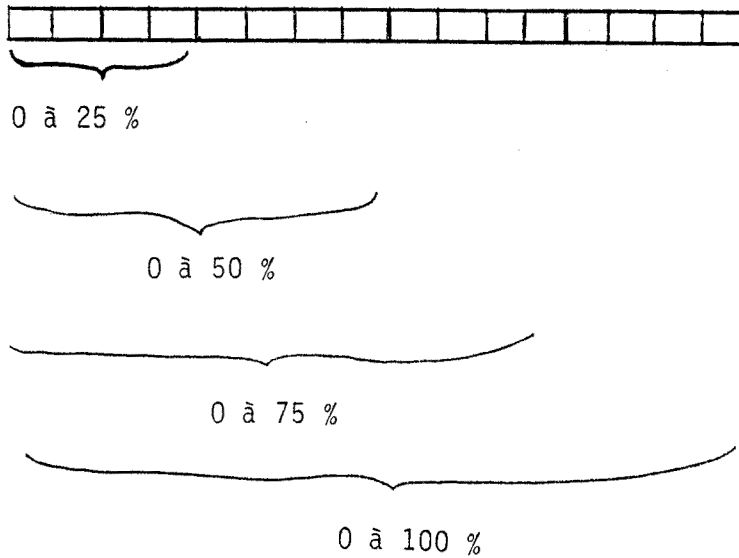
2.3 Le bandeau de visualisation

Le bandeau de visualisation est un prolongement du CMF et permet :

2.3. a) de visualiser l'heure secourue du système sous la forme :



2.3. b) d'indiquer la charge de traitement de l'unité centrale, de 0 à 100 %.



2.3. c) d'indiquer l'état du chien de garde

Le bandeau de visualisation est interfacé avec :

- le micro-processeur temps réel pour l'affichage de l'heure et de l'état du chien de garde,
- et directement avec le bus d'entrées-sorties du SOLAR pour l'affichage de la charge de l'unité centrale.

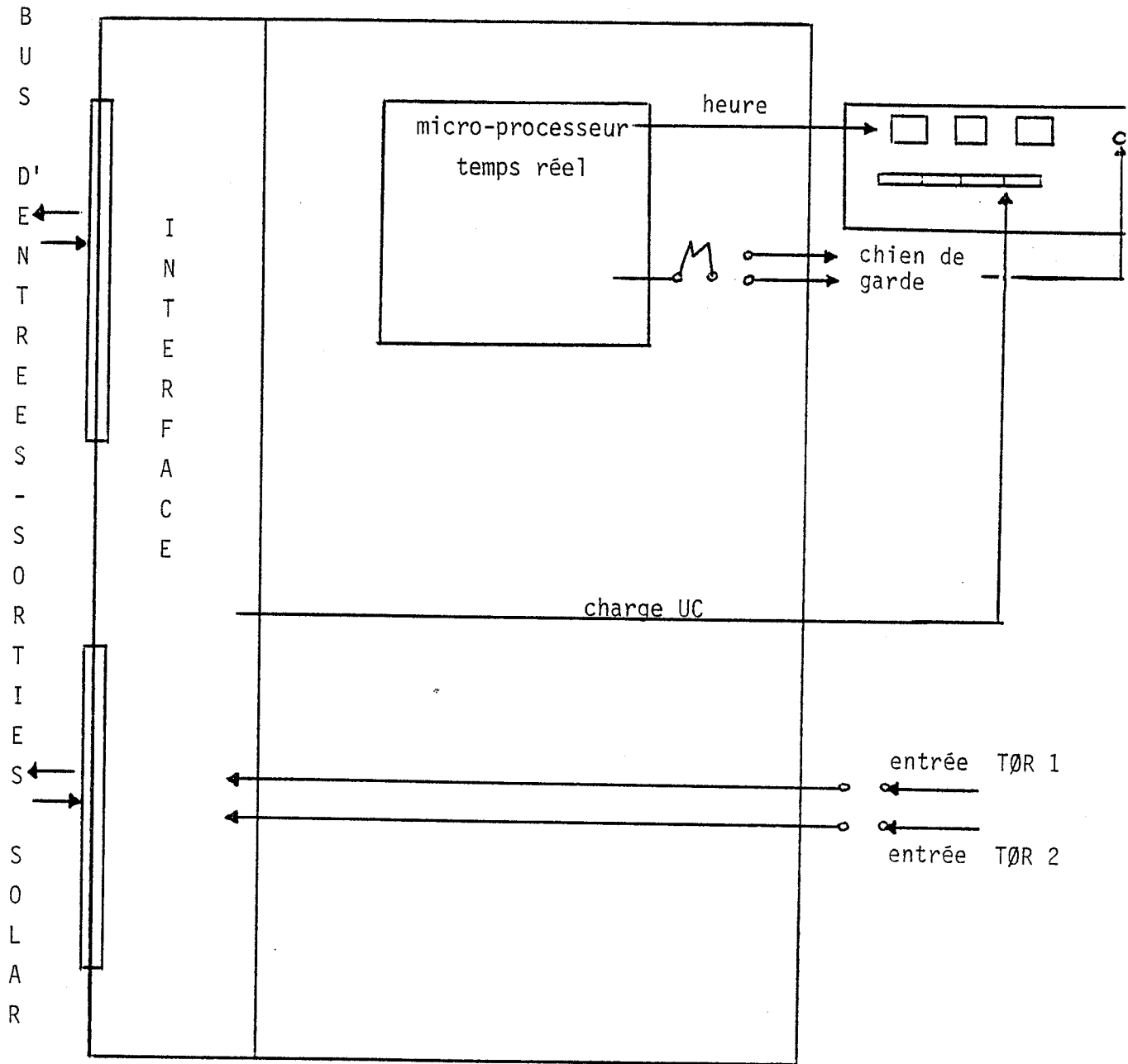


Figure 2.5 : le chien de garde, les entrées TOR et le bandeau de visualisation.

2.4. Les signaux du pupitre de commande et du coupleur X25 SOLAR

La gestion des fonctions de téléchargement et de télécommande a nécessité le transit des signaux de commande du PUC SOLAR par le coupleur CMF.

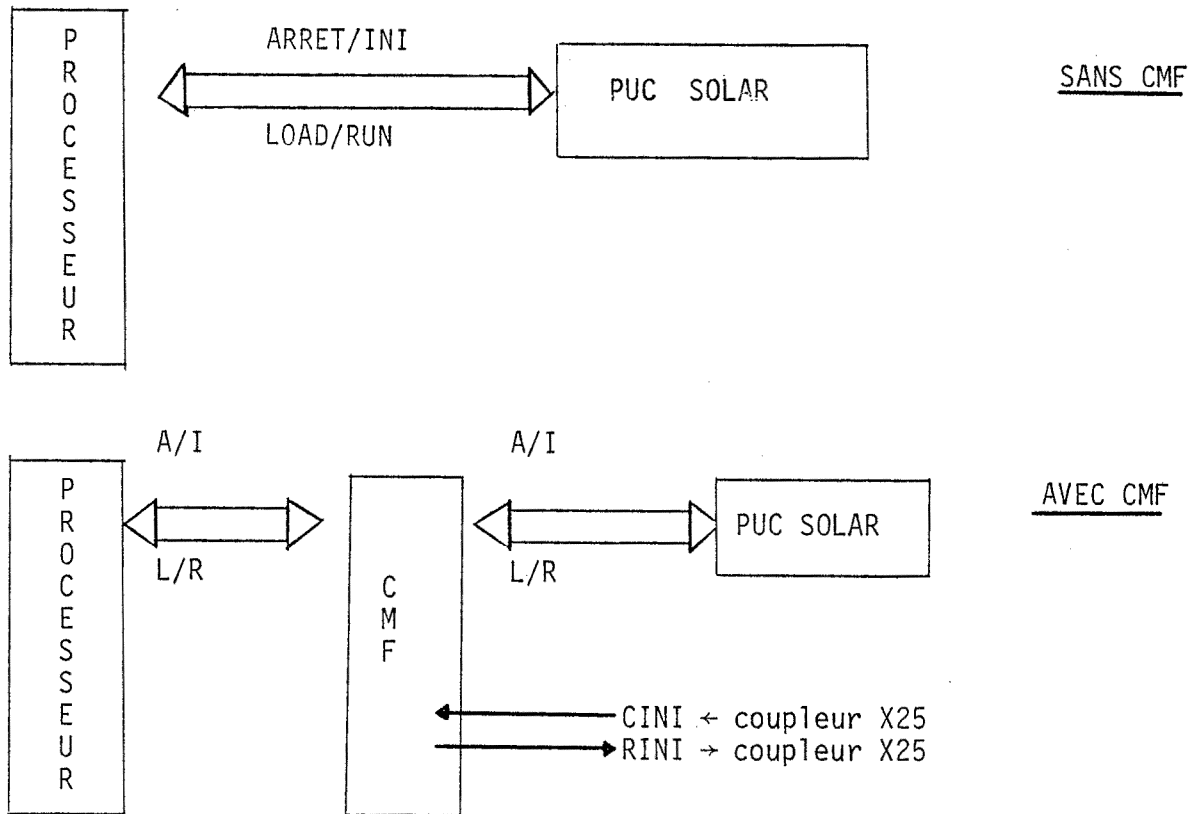


Figure 2.6 : les signaux du pupitre de commande et du coupleur X25

Le signal CINI, généré par le coupleur X25 SOLAR, demande au CMF de réinitialiser le SOLAR et de recharger l'application à partir d'un périphérique bootstrapable quelconque ; le CMF se charge d'exécuter les commandes "ARRET", "INI", "LOAD", "RUN" qui lancent la séquence d'initialisation du coupleur X25.

Le signal RINI, généré par le CMF, est un accusé de réception et d'exécution de la commande CINI, destiné au coupleur X25.

C'est le micro-processeur de gestion des fonctions de transmission qui précise au processeur le type d'action à réaliser : ARRET, INI, LOAD ou RUN.

3. L'INTERFACE DU CMF AVEC LE BUS D'ENTREES-SORTIES DU SOLAR

Vu par le SOLAR, le coupleur multi-fonctions regroupe 4 fonctions :

- la fonction de gestion des 2 lignes asynchrones
- la fonction pupitre opérateur (ou "POP")
- la fonction temps réel
- la fonction horloge temps réel

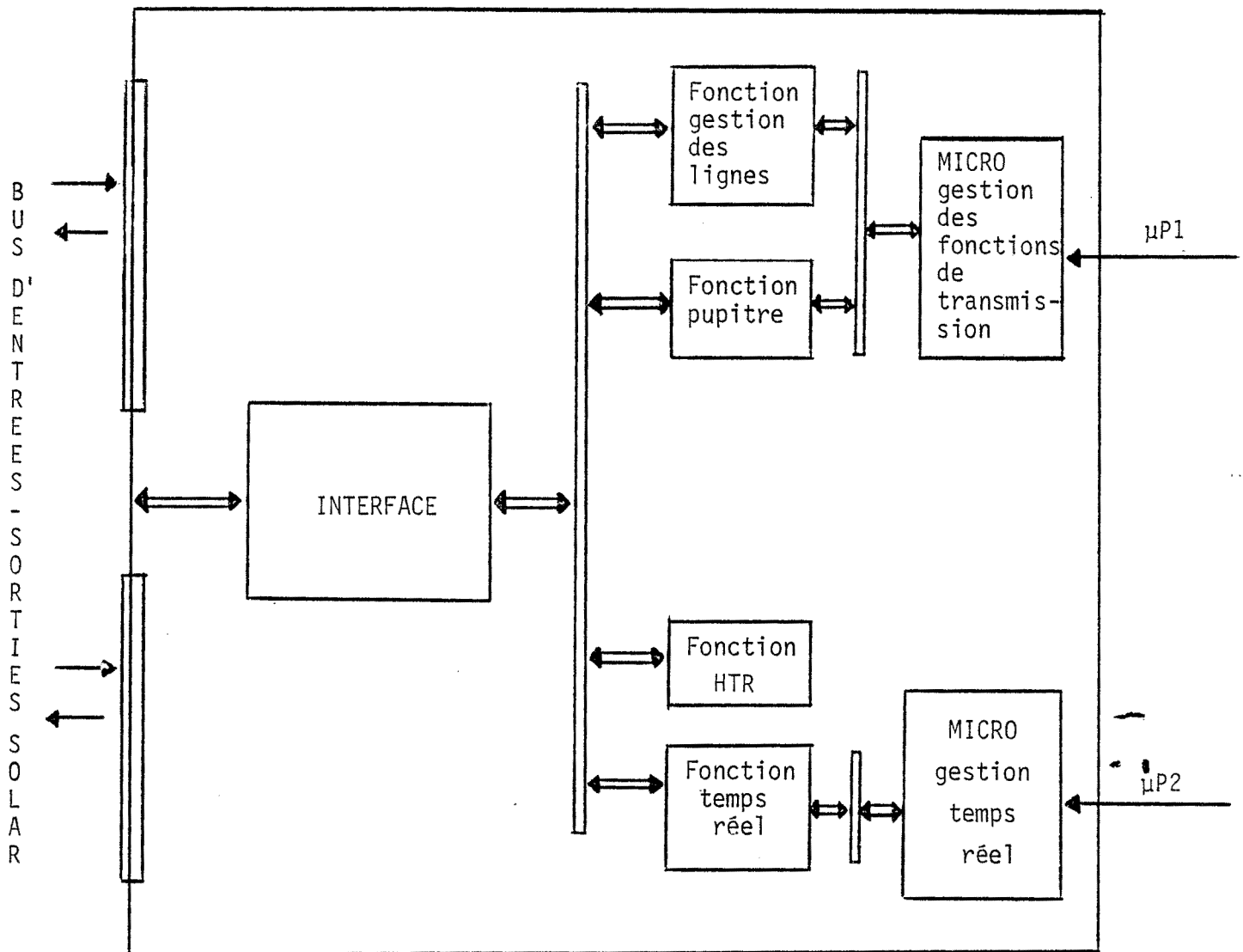
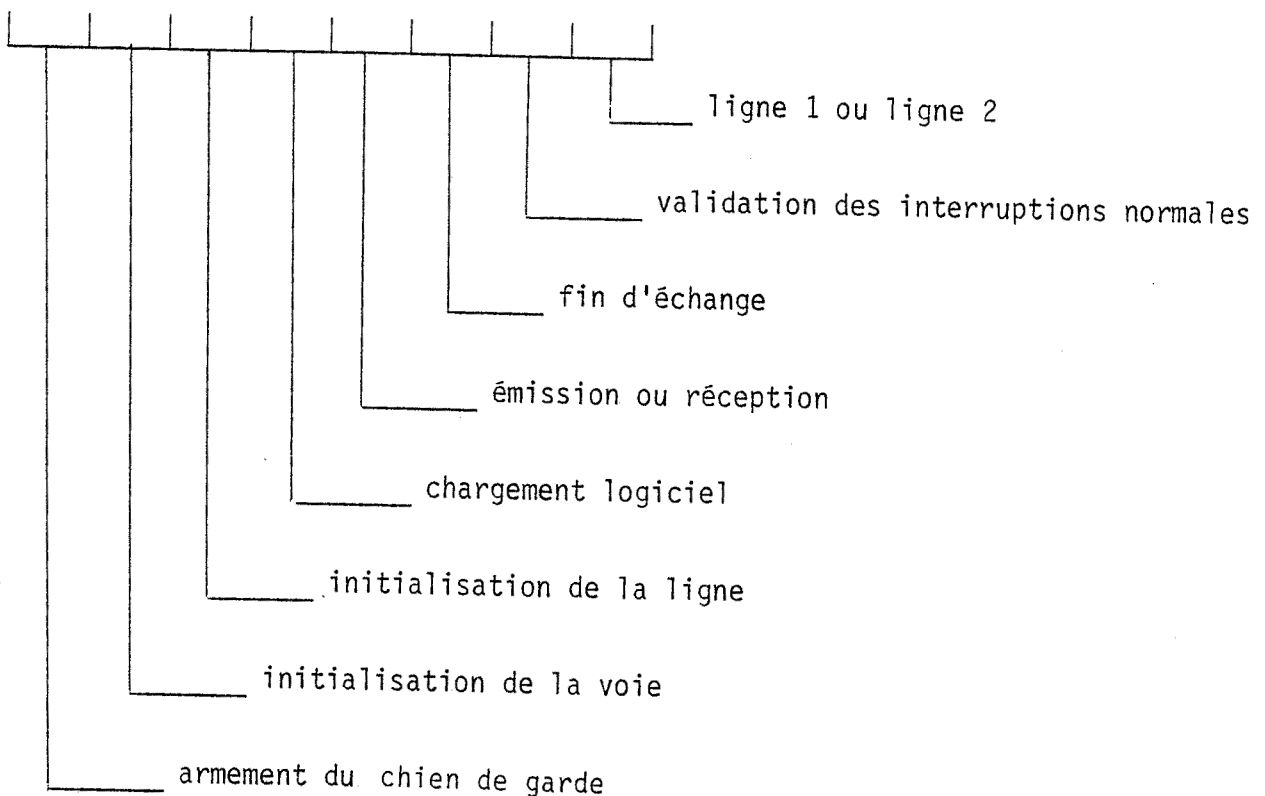


FIGURE 2-7 : Interface CMF/bus d'entrées-sorties SOLAR

Pour dialoguer avec le logiciel SOLAR ou le canal micro-programmé, les micro-processeurs disposent d'un certain nombre de registres par fonction.

3.1. Pour la fonction de gestion des lignes asynchrones, le $\mu P1$. (voir figure 2.7) dispose de 5 registres, par ligne :

3.1. a) le registre de commande 1 dans lequel il trouve tous les paramètres lui permettant d'initialiser ou d'arrêter un échange en émission ou en réception



Cette organisation du mot de commande permet, par association de bits de réaliser :

- l'initialisation de la ligne
- l'initialisation de la voie émission ou de la voie réception
- l'armement du chien de garde émission ou réception
- le chargement de logiciel et de paramètres (cf III.1.2.4.)
- le lancement d'un échange en émission ou en réception
- la fin d'un échange en émission ou en réception

3.1. b) Le registre de commande 2 permettant d'initialiser un code d'arrêt. Les commandes venant des registres de commande 1 ou 2 sont stockées dans "FIFO" de 16 positions pour permettre au logiciel SOLAR d'exécuter plusieurs commandes successives

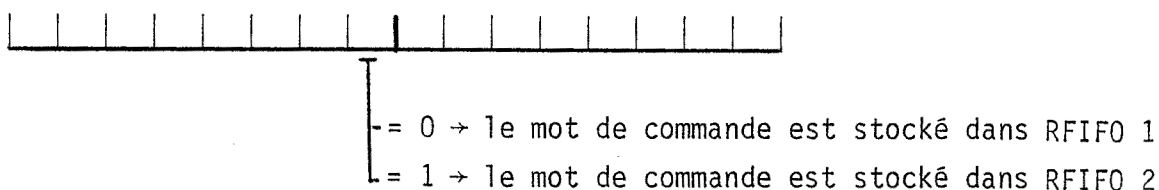
3.1. c) Le registre d'état dans lequel le micro-logiciel rend au SOLAR le type d'événements à prendre en compte :

- état de la voie émission (occupée ou pas)
- état de la voie réception (un caractère présent ou pas)
- état des signaux modem, sur la ligne 2 (IA, PDP, PAE, DP)
- erreurs détectées (cadence, parité, format)
- arrivée à échéance des chiens de garde
- fin d'échange en émission ou en réception
- messages de la procédure si une procédure est implémentée sur le coupleur.

3.1. d) Le registre d'entrée informations dans lequel, le micro-logiciel mémorise l'information reçue sur la ligne.

3.1. e) Le registre de sortie informations dans lequel il trouve la donnée à émettre sur la ligne.

Le registre de commande 1 et le registre de commande 2 sont, vu du côté SOLAR, le même registre de commande :



Le registre de commande et le registre de sorties informations sont écrits par le logiciel SOLAR ou le canal micro-programmé.

Le registre d'entrée informations est lu par le logiciel SOLAR ou le canal micro-programmé. Le registre d'état est lu par le logiciel SOLAR.

3.2. Pour la fonction pupitre, le $\mu P1$ dispose de 3 registres pour dialoguer avec le processeur SOLAR.

3.2. a) Le registre de commande précise au processeur SOLAR le type d'action à réaliser

- écriture ou lecture mémoire ou registre
- positionnement ou effacement d'un ou plusieurs points d'arrêt
- incrémentation d'une adresse mémoire ou d'un numéro de registre

3.2. b) Le registre d'entrée informations précise au processeur l'adresse mémoire, le numéro de registre ou la donnée à traiter.

3.2. c) Le registre de sortie informations rend au micro-logiciel l'adresse mémoire, le numéro de registre ou la donnée traitée par le processeur.

Le registre de commande et le registre de sortie informations sont écrits par le processeur SOLAR.

Le registre d'entrée informations est lu par le processeur SOLAR.

3.3. La fonction horloge temps réel (HTR) est utilisée par le logiciel SOLAR pour activer périodiquement certaines tâches de son système ; elle est assurée par le matériel et ne nécessite qu'un seul registre de commande pour la valider ou la dévalider ; le matériel fournit périodiquement une interruption exception au logiciel SOLAR.

3.4. Les fonctions temps réel sont réalisées par le micro-processeur $\mu P2$ (voir figure 2.7). Pour dialoguer avec le logiciel SOLAR ou le canal micro-programmé, le micro-processeur $\mu P2$ dispose de 4 registres :

3.4. a) Le registre de commande dans lequel il trouve tous les paramètres lui permettant :

- d'initialiser la date, l'heure, les heures de réveils, les temporisations, ou le chien de garde
- de rendre au logiciel SOLAR l'heure ou la date

- d'armer ou de désarmer le chien de garde
- d'activer une temporisation

3.4. b) Le registre d'entrée informations dans lequel il mémorise une des données à rendre au SOLAR (date ou heure)

3.4. c) Le registre de sortie informations dans lequel il trouve une des données émises par le SOLAR pour initialiser la date, l'heure, les heures de réveil, les temporisations ou le chien de garde.

3.4. d) Le registre d'état dans lequel il précise au logiciel SOLAR l'événement arrivant à échéance :

- temporisation
- heure de réveil
- fin d'échange canal en émission (fin d'initialisation) ou en réception (de l'heure ou de la date)
- défaut :
 - . défaut batterie (alimentant l'heure secourue)
 - . défaut horloge (qui délivre le top 10 ms)
 - . changement d'état d'une entrée tout ou rien.

Dans ce mot d'état, nous avons également introduit la notion d'occupation du micro-logiciel temps réel, pour éviter les conflits d'accès aux fonctions temps réel dans une structure multitâche car toutes les fonctions traitées transitent par les mêmes registres d'interface.

Le registre de commande et le registre de sortie informations sont écrits par le logiciel SOLAR ou le canal micro-programmé. Le registre d'entrée informations est lu par le logiciel SOLAR ou le canal micro-programmé, le registre d'état est lu par le logiciel SOLAR.

4. L'ENVIRONNEMENT DES 2 MICRO-PROCESSEURS

Les 2 micro-processeurs sont des micro-processeurs 8 bits (6 809 de MOTOROLA). Ce type de micro-processeur a été choisi pour la richesse de son code d'ordres, la simplicité de sa mise en oeuvre et surtout parce que nous disposons de tous les outils matériels et logiciels indispensables à la mise au point et à la production de programmes.

L'interface entre ces micro-processeurs et le matériel est réalisée grâce à des registres qui mémorisent :

a) Pour le micro-processeur de gestion des fonctions de transmission

= les paramètres du système : ce sont des informations définies à la configuration du système :

- par micro-interrupteurs ou par "track-switchs" sur le CMF réalisé à base de composants discrets.

- par conversationnel à partir d'un outil de dialogue connecté sur la ligne 1 d'un CMF version réseau prédiffusé.

= les états du système : ce sont des informations évolutives, dépendantes de l'environnement SOLAR (tel l'état STOP ou RUN du pupitre de commande).

b) Pour les deux micro-processeurs

= les événements à prendre en compte viennent de l'interface ligne, de l'environnement extérieur (tel le pupitre de commande, les entrées TOR...) ou de l'interface SOLAR ; ils impliquent une action du micro-logiciel.

La priorité donnée à chaque événement dépend du niveau d'interruption qui lui est affecté.

Au niveau du micro-processeur 6809, il existe 3 types d'interruption matérielle :

- NMI : interruption non masquable, de plus grande priorité
- FIRQ : interruption masquable, de priorité inférieure à NMI
- IRQ : interruption masquable, de priorité inférieure à FIRQ.

L'interruption de type NMI n'est pas utilisée car elle est affectée à l'outil de mise au point de programme.

Une tâche IRQ peut être interrompue par une tâche FIRQ.

= les actions à réaliser dépendent des événements reçus.

Nous rappelons que le micro-processeur de gestion des fonctions de transmission réalise la gestion :

- des 2 lignes asynchrones,
 - des fonctions pupitres,
- et - des fonctions de télécommande et de téléchargement.

4.1. Les paramètres du système

4.1.1. Pour la fonction de gestion des 2 lignes asynchrones

4.1.1. a) le format des caractères à recevoir ou à émettre sur chaque ligne : est utilisé pour initialiser l'interface micro-ligne.

4.1.1. b) la vitesse de la ligne 2 : ce paramètre permet de calibrer certains signaux de service (DPE - demande pour émettre).

4.1.1. c) le mode de fonctionnement du modem (half ou full duplex) : doit être connu par le micro-logiciel pour gérer les signaux modem (cf III.1.1.5).

4.1.1. d) l'option de "bufférisation" : évite la perte de caractères en réception dans le cas d'une émission réception half-duplex et d'un temps de retournement trop long de la part du logiciel SOLAR.

4.1.1. e) l'option traitement de procédure : permet au micro-logiciel du CMF de gérer la procédure de gestion de ligne asynchrone implantée sur le coupleur.

4.1.2. Pour les fonctions de téléchargement et de télécommande

4.1.2. a) l'option "TIP VALIDE" : sur une ligne asynchrone, permet d'assurer dans une structure hiérarchisée, la fonction de téléchargement.

4.1.2. b) le numéro de bootstrap : donne le périphérique concerné par la fonction de téléchargement.

4.1.2. c) le numéro de frontal : dans une configuration multi-points, permet au micro de savoir si le message TIP qui transite sur la ligne lui est adressé.

4.1.2. d) l'option "TRANSPAC", à la mise sous tension du système, active la séquence de démarrage et de connexion au réseau TRANSPAC.

4.1.3. Pour les fonctions pupitre

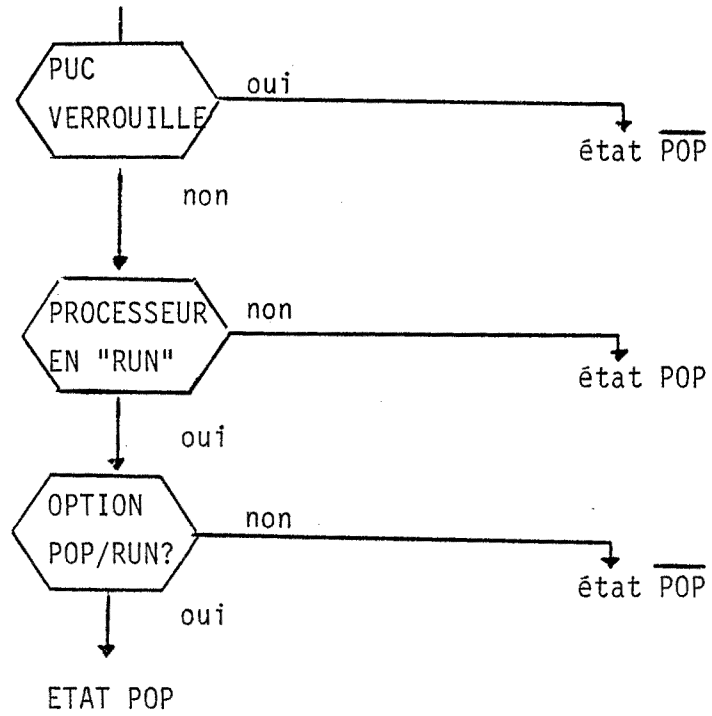
L'option "POP/RUN" valide la fonction pupitre même si le processeur est en mode "RUN"

4.2. LES ETATS DU SYSTEME

4.2.1. l'état "POP" indique au micro-logiciel l'aiguillage à réaliser sur réception d'un caractère : vers le module de gestion des fonctions pupitre ou vers un autre module.

Cet état dépend :

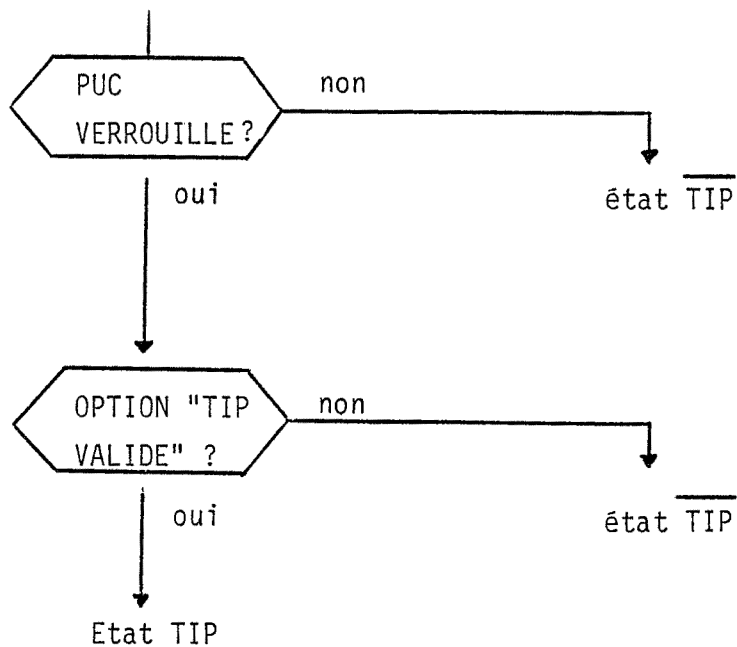
- du verrouillage du pupitre de commande
- de l'état "STOP" ou "RUN" du processeur
- de l'option "POP/RUN" (cf II.4.1.3.)



4.2.2. l'état "TIP", sur chaque ligne, conditionne la prise en compte et l'exécution de la fonction de téléchargement.

Cet état dépend :

- du verrouillage du pupitre de commande
- de l'option "TIP valide" (cf II.4.1.2.)



4.3. Les événements à prendre en compte

4.3.1. Par le micro-processeur de gestion des fonctions de transmission 8 événements génèrent des interruptions FIRQ : 2 viennent de l'interface ligne et 6 de l'interface SOLAR.

Une interruption FIRQ est générée par :

= l'ACIA de la ligne 1 ou de la ligne 2 sur :

- . buffer émission vide
- . buffer réception plein (cet événement est le plus prioritaire car pour éviter une erreur de cadence, il doit être traité à la vitesse de la ligne ; nous avons donc introduit la possibilité de le démasquer, sous tâche FIRQ, pour en permettre la prise en compte et le traitement par interruption d'une autre tâche FIRQ).

= l'interface avec le bus d'entrées-sorties du SOLAR sur :

- . écriture du registre de commande 1 ou écriture du registre de commande 2 par le logiciel SOLAR
- . écriture du registre de sorties informations de la ligne 1 ou de la ligne 2 par le logiciel SOLAR ou le canal micro-programmé,
- . lecture du registre d'entrées-informations de la ligne 1 ou de la ligne 2 par le logiciel SOLAR ou le canal micro-programmé (voir les registres § II.3.1.)

4 événements masquables globalement génèrent une interruption IRQ

= la lecture du registre d'état (Cf II.3.1.) de la ligne 1 ou de la ligne 2 par le logiciel SOLAR

= le passage du processeur à l'état "STOP"

= le "TOP" 10 ms utilisé pour gérer les chiens de garde émission ou réception des 2 lignes.

Quelques événements sont des événements non prioritaires et n'interrompent pas la tâche en cours :

= l'indicateur d'appel : c'est la demande de connexion du correspondant, dans un réseau commuté (Cf III.1.1.5)

= CINI X25: cet événement est généré par le coupleur X25 SOLAR, sur réception d'une trame de télécommande et demande au micro-logiciel d'exécuter la séquence d'initialisation du coupleur X25.

= le processeur SOLAR a adressé (en lecture ou en écriture) un des registres de la fonction pupitre (Cf. II.3.2.)

4.3.2. Par le micro-processeur de gestion des fonctions temps réel

5 événements génèrent une interruption FIRQ

= le signal de synchronisation, généré dans une structure hiérarchisée par le calculateur maître et utilisé par le micro-logiciel pour réajuster sa base de temps (Cf V.3).

= les commandes d'armement ou de désarmement du chien de garde venant de l'interface SOLAR et permettant au micro-logiciel d'initialiser ou d'arrêter la fonction chien de garde.

= la commande de fin d'échange exécutée par le canal micro-programmé sur fin d'émission (initialisation de la date, de l'heure, des heures de réveil, des temporisations ou du chien de garde) ou sur fin réception (lecture de la date ou de l'heure).

= un défaut :

- . défaut batterie (alimentation de l'heure secourue)
- . changement d'état d'une des 2 entrées tout ou rien.

= le TOP "1 ms" utilisé dans la phase de synchronisation de l'heure (Cf. V.3).

7 événements venant de l'interface SOLAR génèrent une interruption IRQ.

Ce sont les demandes faites par le logiciel SOLAR au micro (Cf. II.3.4.)

- lecture ou écriture de la date
- lecture ou écriture de l'heure
- initialisation des heures de réveil
- initialisation des temporisations
- initialisation du chien de garde
- armement ou désarmement d'une temporisation

C'est l'écriture du registre de commande ou du registre de sortie informations ou la lecture du registre d'entrée informations (Cf. II.3.4) qui provoque l'interruption.

Le TOP 10 ms, utilisé pour gérer l'heure, la date, les temporisations, les heures de réveil, le chien de garde et le signal de synchronisation ne génère pas d'interruption : il est traité dans la tâche de fond qui est la tâche principale du micro-logiciel temps réel.

4.4. Les actions

4.4.1. Pour la fonction téléinfo, le micro-logiciel gère :

- les interruptions normales émission et réception
- les interruptions exception émission et réception
- le mot d'état rendu au SOLAR
- l'émission / réception d'un caractère au niveau de l'interface SOLAR
- les signaux modem
- l'émission/réception d'un caractère au niveau de l'interface ligne.

4.4.2. Pour les fonctions pupitre, les actions du micro-processeur se limitent à lire ou à écrire les registres de dialogue avec le processeur SOLAR (Cf. II.3.2) et à éditer les résultats (rendus par le processeur et mis en forme par le micro-logiciel) sur l'organe de visualisation par écriture ou lecture des registres d'interface avec la ligne (Cf. II.2.1).

4.4.3. Pour la fonction de téléchargement, le micro-logiciel précise au processeur le type de commande à réaliser, par l'intermédiaire d'un registre.

- ARRET
- INI
- LOAD (associée à un numéro de bootstrap)
- RUN

4.4.4. Dans le cadre des fonctions temps réel, les actions sont destinées :

- à alerter le logiciel SOLAR :
 - . sur détection d'un défaut
 - . sur l'échéance d'une temporisation ou d'une heure de réveil
- à rendre au SOLAR des informations concernant l'heure ou la date
- à initialiser l'heure, la date, les heures de réveil, les temporisations ou le chien de garde
- à alerter l'environnement extérieur de l'échéance du chien de garde

5. SUR CHAQUE MICRO-PROCESSEUR, LE DECOUPAGE DE L'ADRESSAGE MEMOIRE EST LIE

= à l'implantation du module de mise au point de programme qui doit cohabiter avec le micro-logiciel pendant la phase de test du programme .

= au souci constant de gain de composants (dans la fonction de décodage de l'adresse) .

= à l'espace réservé au micro-logiciel

. 8 ou 16 K ** de mémoire morte en EPROM

. 8 K de mémoire vive pour $\mu P1$ * et 2 K de mémoire vive pour $\mu P2$ * .

et de plus pour $\mu P1$ * :

= à l'espace réservé à l'utilisateur pour l'intégration d'une procédure de gestion de lignes asynchrones

. 8 ou 16 K de mémoire morte ou EPROM

et . 8 K de mémoire vive

ou . 16 K de mémoire vive .

(* $\mu P1$, $\mu P2$ - voir figure 2.7)

(** 1 K = 1024 octets)

5.1. Espace d'adressage du micro-processeur de gestion des fonctions de transmission

5.1.1. L'utilisateur utilise la mémoire vive et la mémoire morte

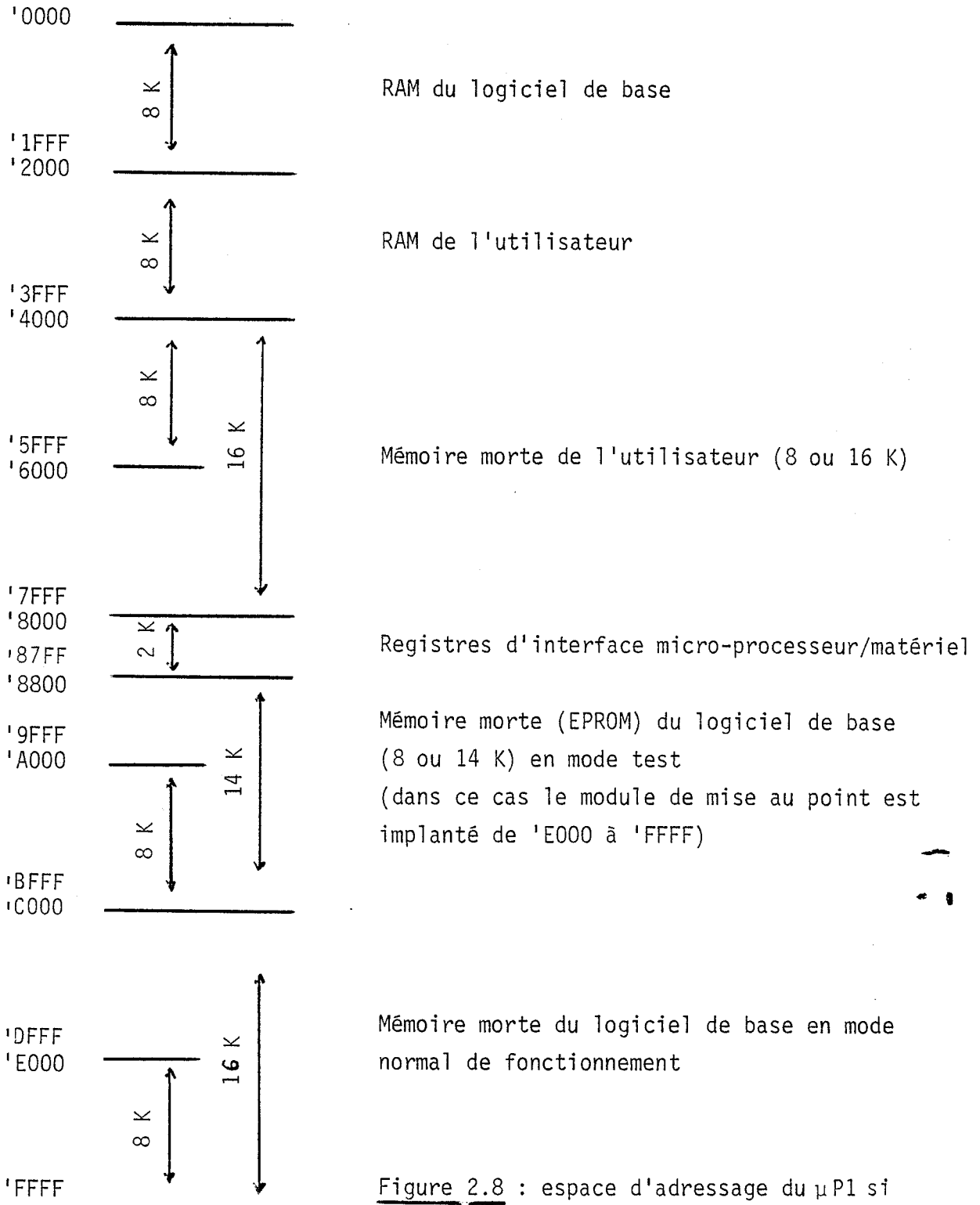


Figure 2.8 : espace d'adressage du $\mu P1$ si l'utilisateur utilise la mémoire vive et la mémoire morte.

5.1.2. L'utilisateur utilise la mémoire vive seule

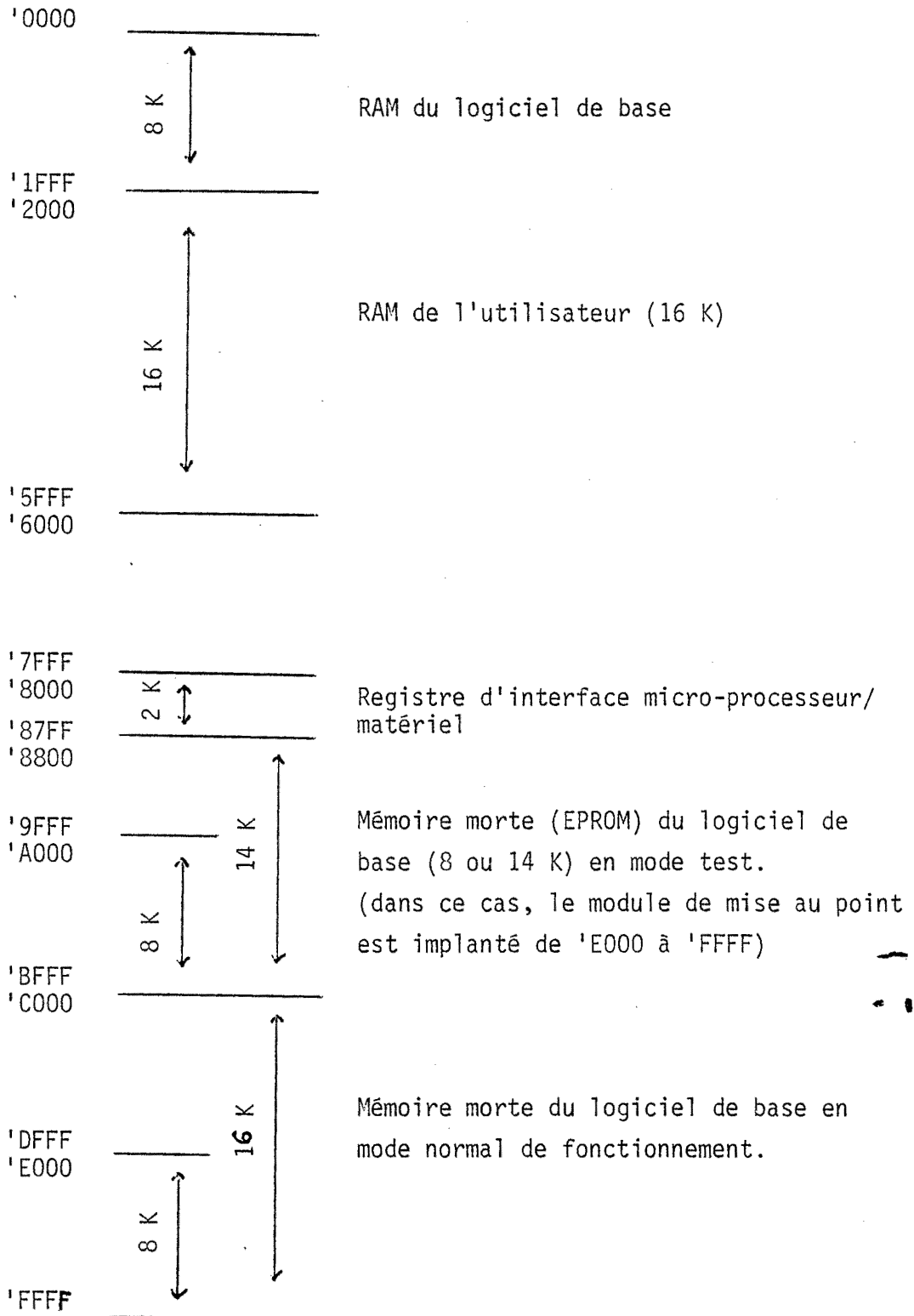
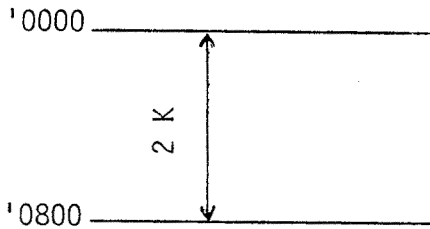
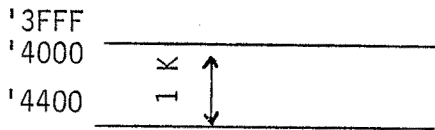


Figure 2.9. : espace d'adressage de $\mu P1$ si l'utilisateur utilise la mémoire vive seule.

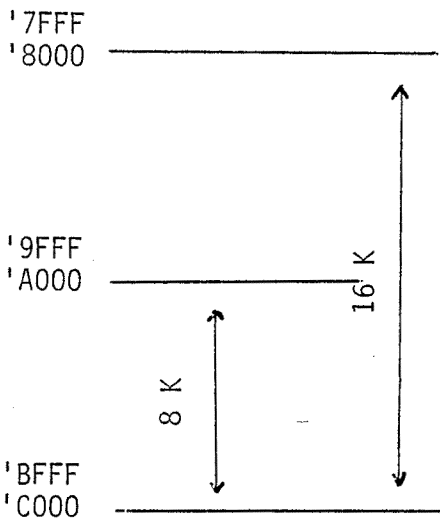
5.2. Espace d'adressage du micro-processeur de gestion des fonctions temps réel



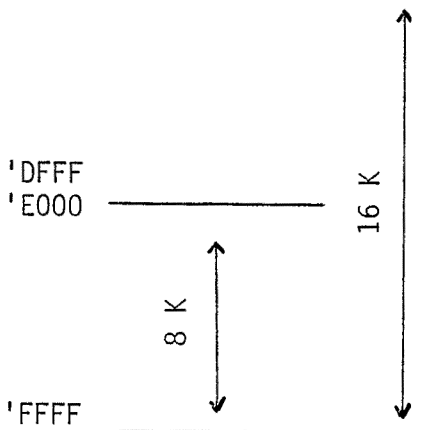
RAM du logiciel temps réel



Registres d'interface micro-processeur/matériel



Mémoire morte du logiciel temps réel en mode test (8 ou 16 K)



Mémoire morte du logiciel temps réel en mode normal de fonctionnement (8 ou 16 K)

CHAPITRE III

LA GESTION DES LIGNES ASYNCHRONES

III. LA GESTION DES LIGNES ASYNCHRONES

1. <u>LE "DRIVER"</u>	47
1.1. <u>La fonction téléinfo</u>	48
1.1.1. Emission et réception en mode programmé simple	48
1.1.2. Emission et réception en mode canal, sans défaut	52
1.1.3. Réception en mode canal, avec défaut	58
1.1.4. Réception avec "bufferisation"	60
1.1.5. Gestion des signaux modem	64
1.1.6. Les chiens de garde logiciel	68
1.2. <u>Les traitements</u>	70
1.2.1. Les codes d'arrêt	70
1.2.2. Les codes spécifiques	71
1.2.3. Les fonctions téléchargement et de télécommande	72
1.2.4. La modification des paramètres du coupleur	75
2. <u>L'INTERFACE AVEC LA PROCEDURE DE L'UTILISATEUR</u>	77
2.1. <u>Les services offerts par le "driver"</u>	78
2.1.1. Initialisation d'une voie	79
2.1.2. Réception des données d'une ligne	79
2.1.3. Emission des données vers l'interface SOLAR	80
2.1.4. Réception des données de l'interface SOLAR	80
2.1.5. Emission de données vers le SOLAR	80
2.1.6. Réveil du SOLAR	81
2.1.7. L'état et la gestion des signaux modem	81
2.1.8. Chiens de garde logiciels	81
2.1.9. Horloge temps réel	82
2.1.10. Heure du système	82
2.2. <u>Le dialogue entre l'utilisateur et le driver</u>	82

	<u>PAGES</u>
2.3. <u>Description des blocs de contrôle</u>	86
2.3.1. Initialisation d'une voie	86
2.3.2. Gestion de l'heure	87
2.3.3. Horloge temps réel	87
2.3.4. Signaux modem	87
2.3.5. Chiens de garde logiciels	88
2.3.6. Réception de la ligne	88
2.3.7. Réception de l'interface SOLAR	90
2.3.8. Emission sur la ligne	90
2.3.9. Emission vers l'interface SOLAR	90
2.3.10. Réveil du SOLAR	91
2.4. <u>Traitements particuliers</u>	92
2.4.1. Les codes spécifiques	92
2.4.2. Les codes d'arrêt	93

Le micro-logiciel gérant les fonctions de transmission est composé :

- d'une partie "driver" dont le rôle est de gérer la partie matérielle du coupleur

- d'une partie interface utilisateur dont le rôle est de permettre l'implémentation d'une procédure de transmission asynchrone half-duplex ou full duplex pour décharger l'application et augmenter ainsi sa capacité de traitement. Cette interface est suffisamment souple pour dialoguer avec plusieurs types de procédure.

1. LE "DRIVER"

Le rôle du "driver" est de gérer la partie matérielle du coupleur ; de ce fait, il est étroitement dépendant de l'organisation matérielle. Il est scindé en 4 grandes parties :

- la phase d'initialisation dans laquelle il réalise toutes les initialisations logicielles et matérielles et où il traite éventuellement les commandes de téléchargement TRANSPAC ou X25.

- la tâche de fond qui lui permet d'attendre un événement, de dialoguer avec la procédure de l'utilisateur ou de gérer les fonctions pupitre.

- la tâche de traitement des interruptions FIRQ

- la tâche de traitement des interruptions IRQ

Ces deux dernières tâches réalisent la plupart des fonctions de la gestion des lignes asynchrones :

- = émission/réception de caractères en mode programmé simple ou en mode canal, avec ou sans "bufferisation", avec gestion des défauts en réception et, sur la ligne 2, des signaux modem

= traitement

- des codes d'arrêt
- des codes spécifiques
- du téléchargement et de la télécommande TIP
- des services proposés à une procédure de transmission

= modification des paramètres du coupleur ou chargement de logiciel

1.1. La fonction téléinfo

Chacune des lignes est full duplex et travaille dans l'un des modes suivants :

- programmé simple
- canal LDC

Le coupleur génère donc :

- une interruption normale (interruption d'entretien d'échange) en émission et en réception pour le micro-logiciel du canal LDC
- une interruption exception (interruption de fin d'échange ou d'événement exceptionnel) en émission et en réception pour le logiciel SOLAR.

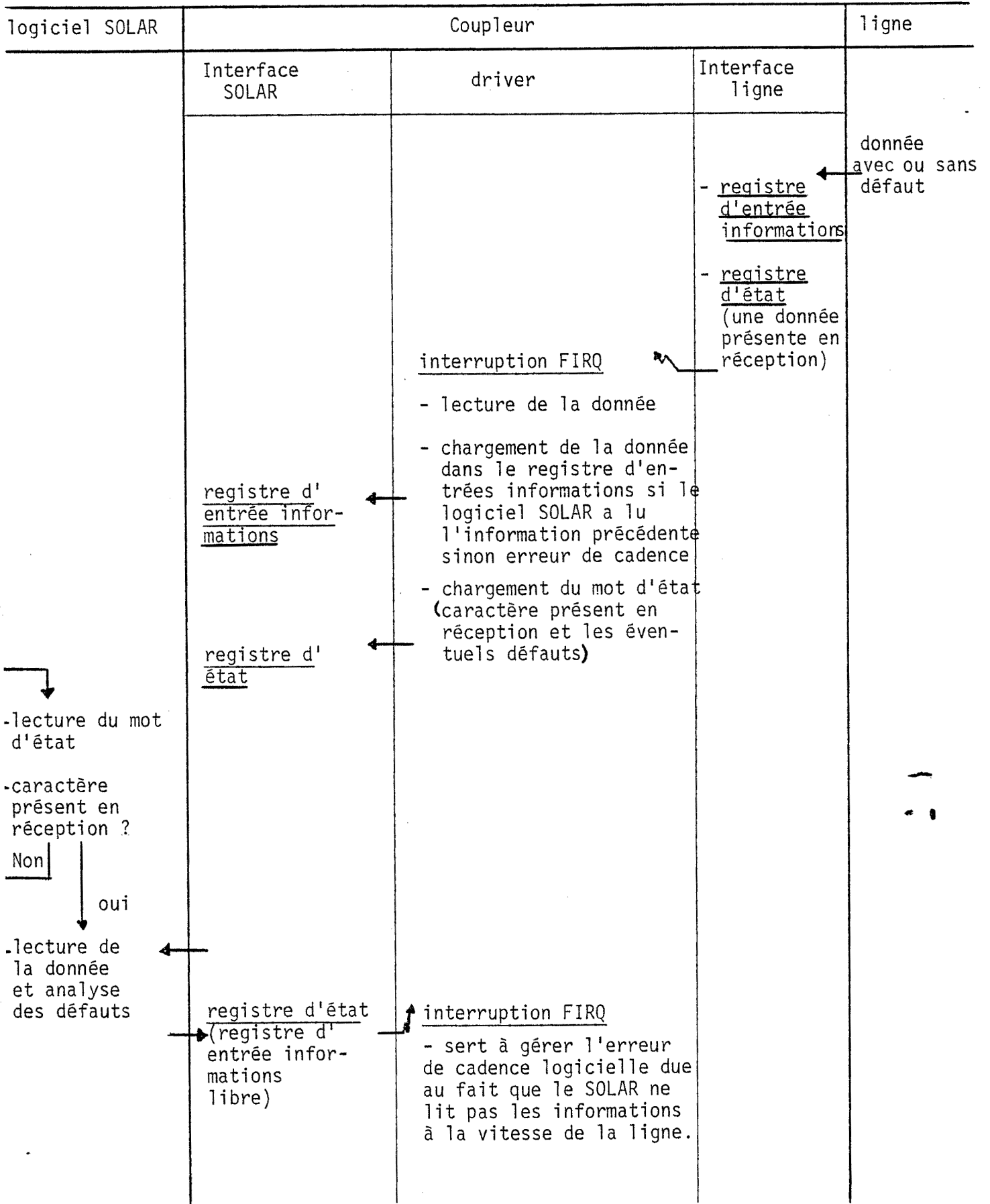
Le mode programmé prioritaire est abandonné car c'est maintenant le CMF qui gère les fonctions de détection des codes d'arrêt multiples et des codes spécifiques.

1.1.1. Description des échanges émission et réception en mode programmé simple

Ils sont identiques sur la ligne 1 et sur la ligne 2 car le mode programmé simple est incompatible avec une gestion des signaux modem ; la ligne 2 ne doit pas être munie d'une interface V24 complète si l'utilisateur l'utilise en mode programmé simple.

1.1.1. b) En réception

Tableau 3.2. : Réception en mode programmé simple

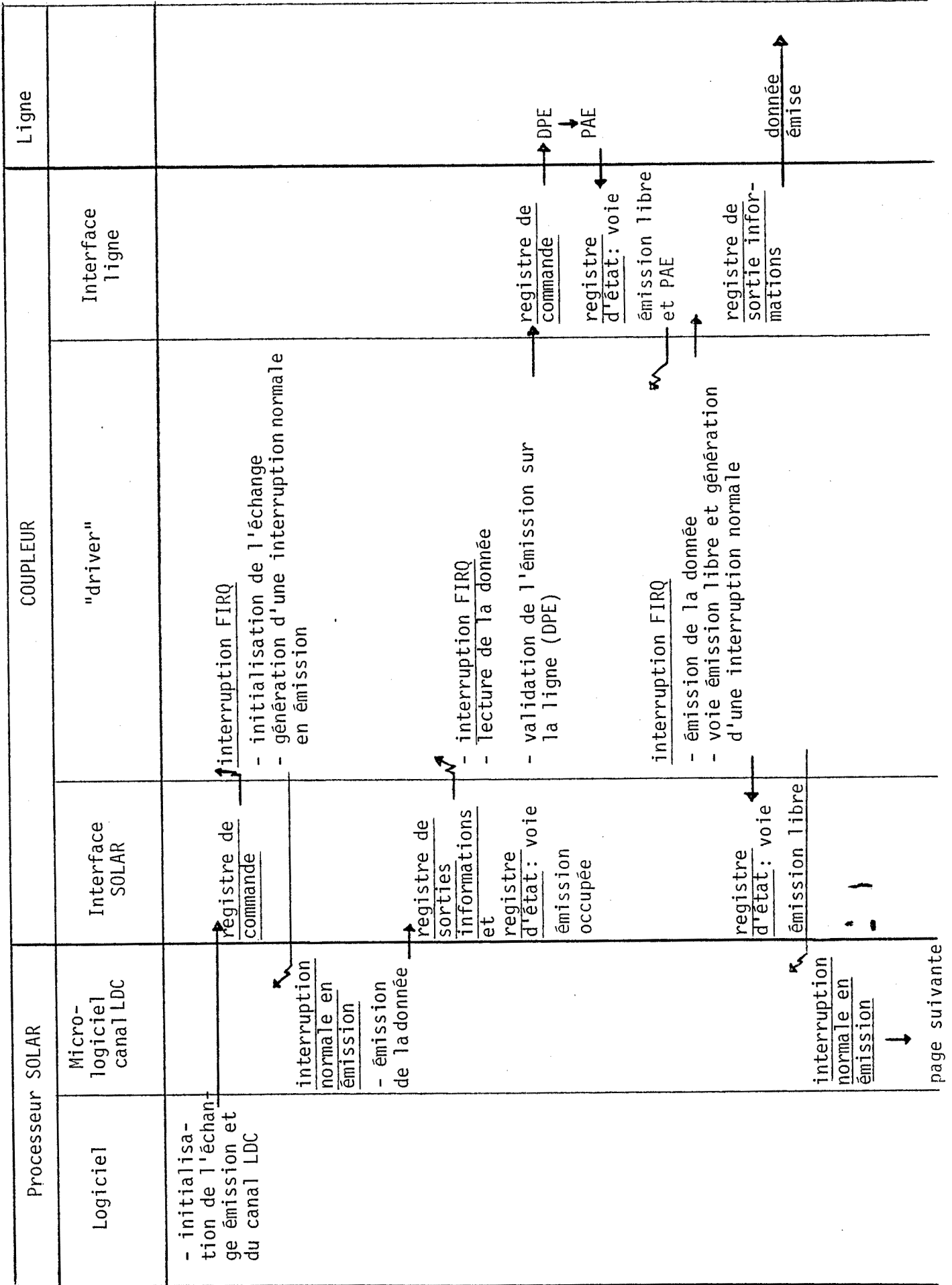


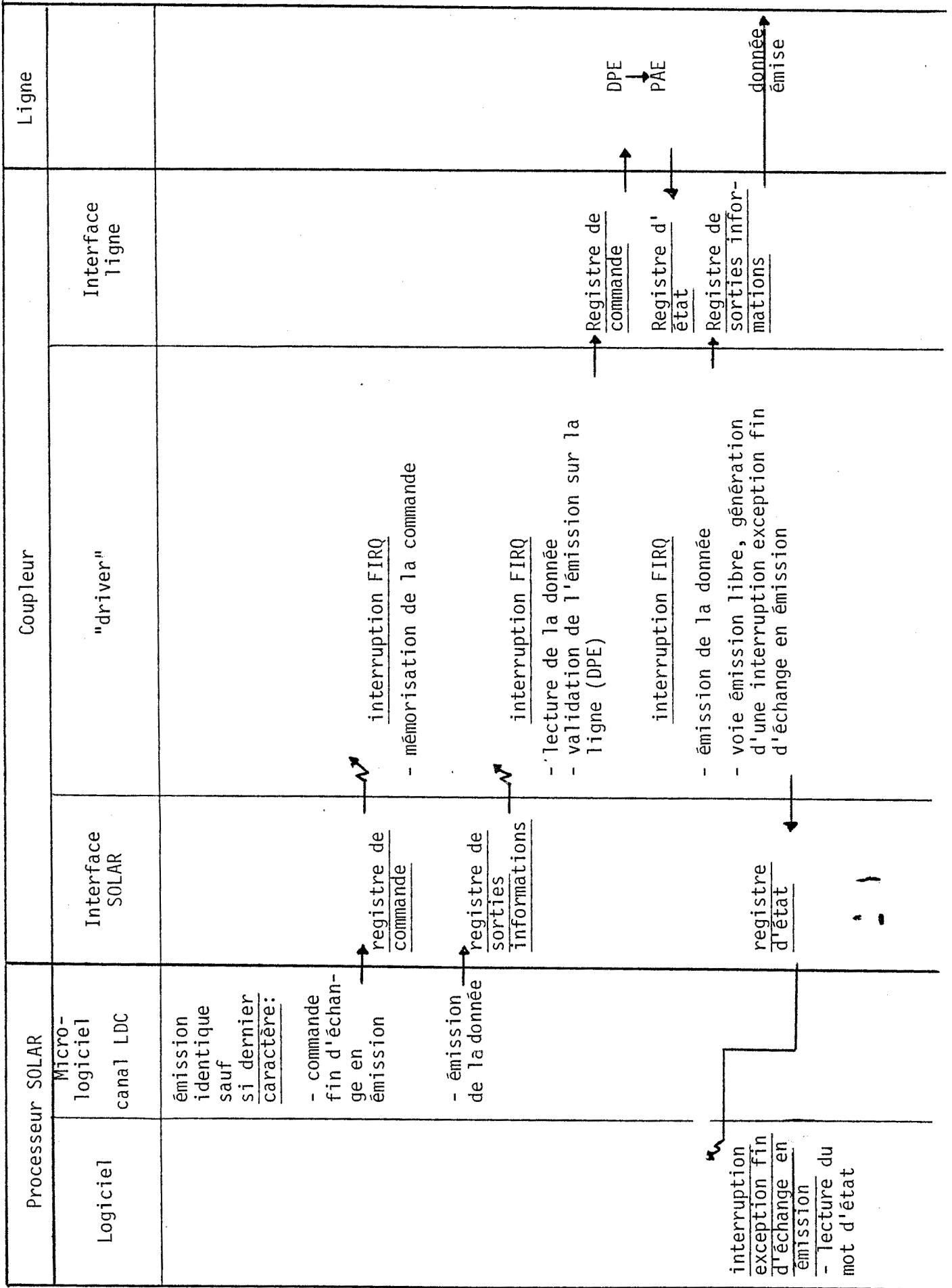
1.1.2. Description des échanges émission et réception en mode canal,
sans traitement d'informations

Les signaux modem sont gérés sur la ligne 2.

1.1.2. a) Echange en émission

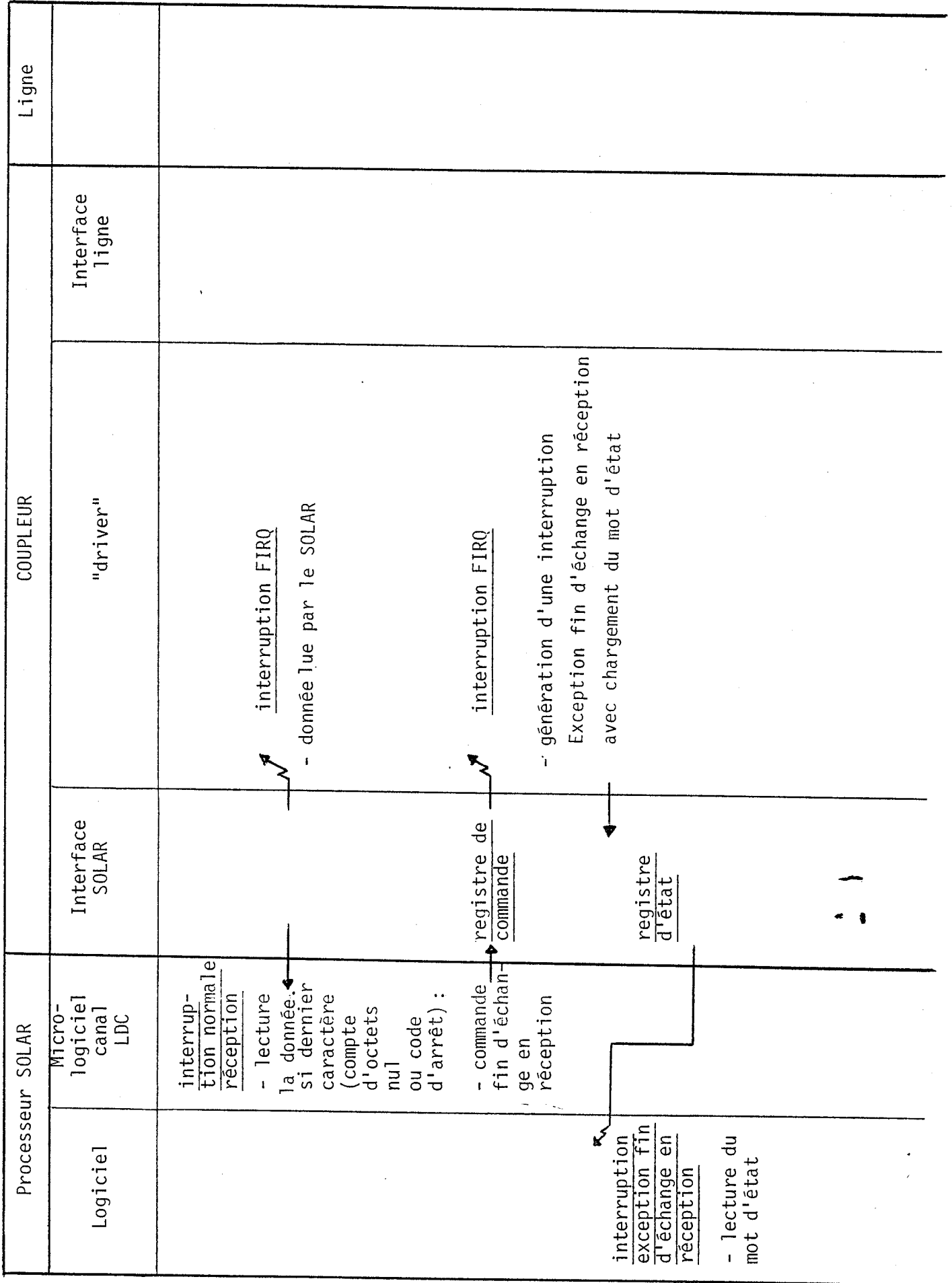
Tableau 3.3. : Emission en canal





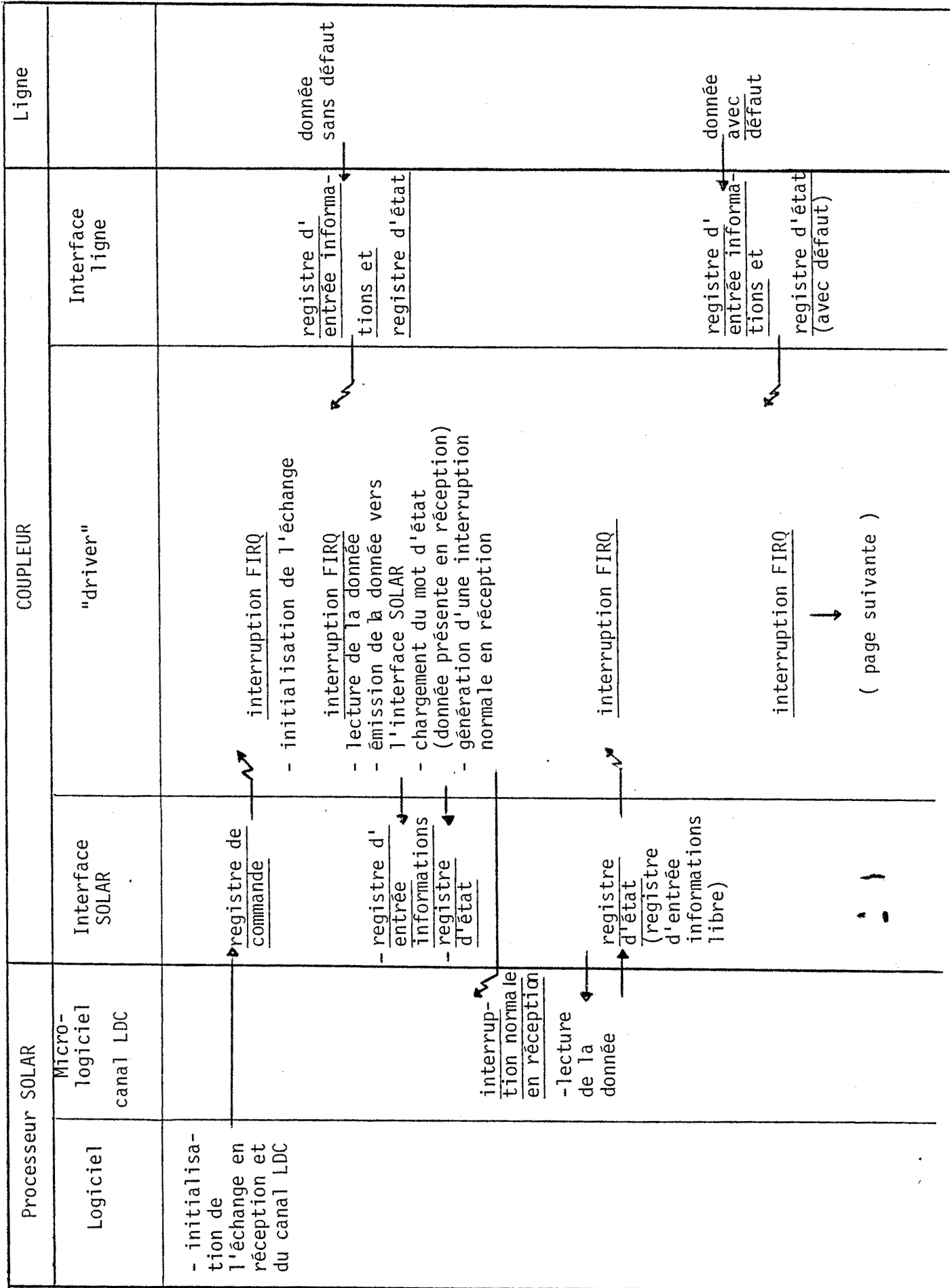
Processeur SOLAR		COUPLEUR			Ligne
Logiciel	Micro-logiciel canal LDC	Interface SOLAR	"driver"	Interface Ligne	
		(registre d'entrée informations libre)	<p><u>interruption FIRQ</u></p> <ul style="list-style-type: none"> - sert à gérer une éventuelle erreur de cadence logique dans le cas où le SOLAR ne lit pas les informations reçues à la vitesse de la ligne 		
	<p><u>interruption normale en réception</u></p> <p>- lecture de la donnée</p>		<p><u>interruption FIRQ</u></p>		donnée sans défaut
		<p>registre d'entrée informations</p> <p>registre d'état</p>	<p>le traitement est identique jusqu'à la réception du dernier caractère</p> <p><u>interruption FIRQ</u></p> <ul style="list-style-type: none"> - lecture de la donnée - chargement de la donnée dans le registre d'entrée informations - chargement du mot d'état et - génération d'une interruption normale réception 	<p>registre d'entrée informations</p> <p>et</p> <p>registre d'état</p>	donnée sans défaut

page suivante



1.1.3. Réception d'une donnée avec défaut

Tableau 3.5 : réception avec défaut



Processeur SOLAR		COUPLEUR		Ligne
Logiciel	Micro-logiciel canal LDC	Interface SOLAR	"driver"	Interface ligne
<ul style="list-style-type: none"> - interruption exception - lecture du mot d'état - lecture de la donnée - traitement du défaut 	<ul style="list-style-type: none"> - registre d'entrées informations - registre d'état (avec défaut) 	<ul style="list-style-type: none"> - lecture de la donnée - émission de la donnée vers l'interface SOLAR - chargement du mot d'état avec défaut - génération d'une interruption exception de défaut (l'échange est arrêté) 		

1.1.4. Réception avec "bufferisation"

Cette fonction est validée par micro-interrupteur sur le coupleur et peut ainsi être activée avec des logiciels existants.

Le "driver" mémorise les données reçues de la ligne dans un tampon circulaire de 256 octets. Sur réception de la commande d'initialisation de l'échange en réception, il vide ce tampon vers le SOLAR. La notion d'erreur de cadence intervient en cas de saturation de ce tampon.

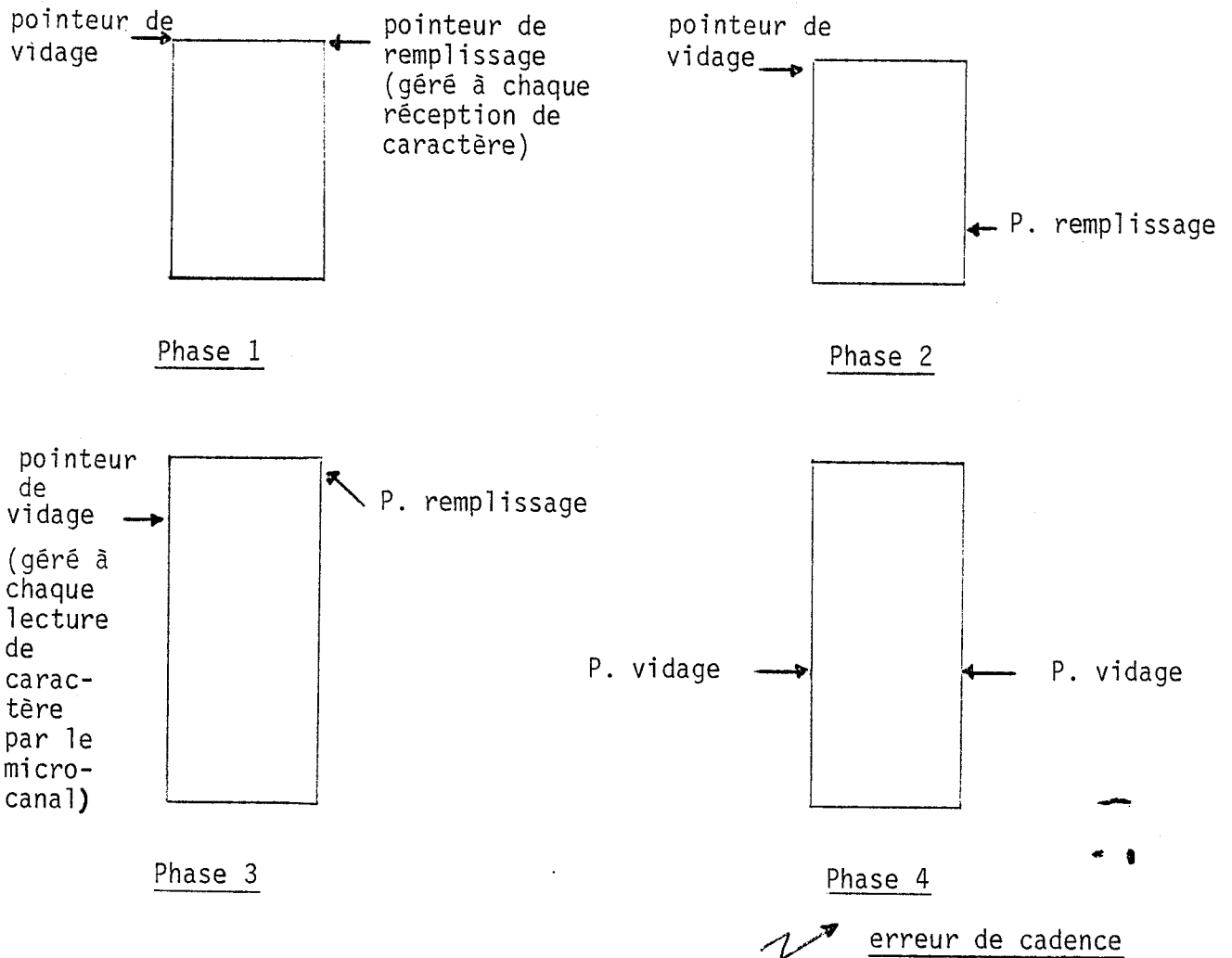
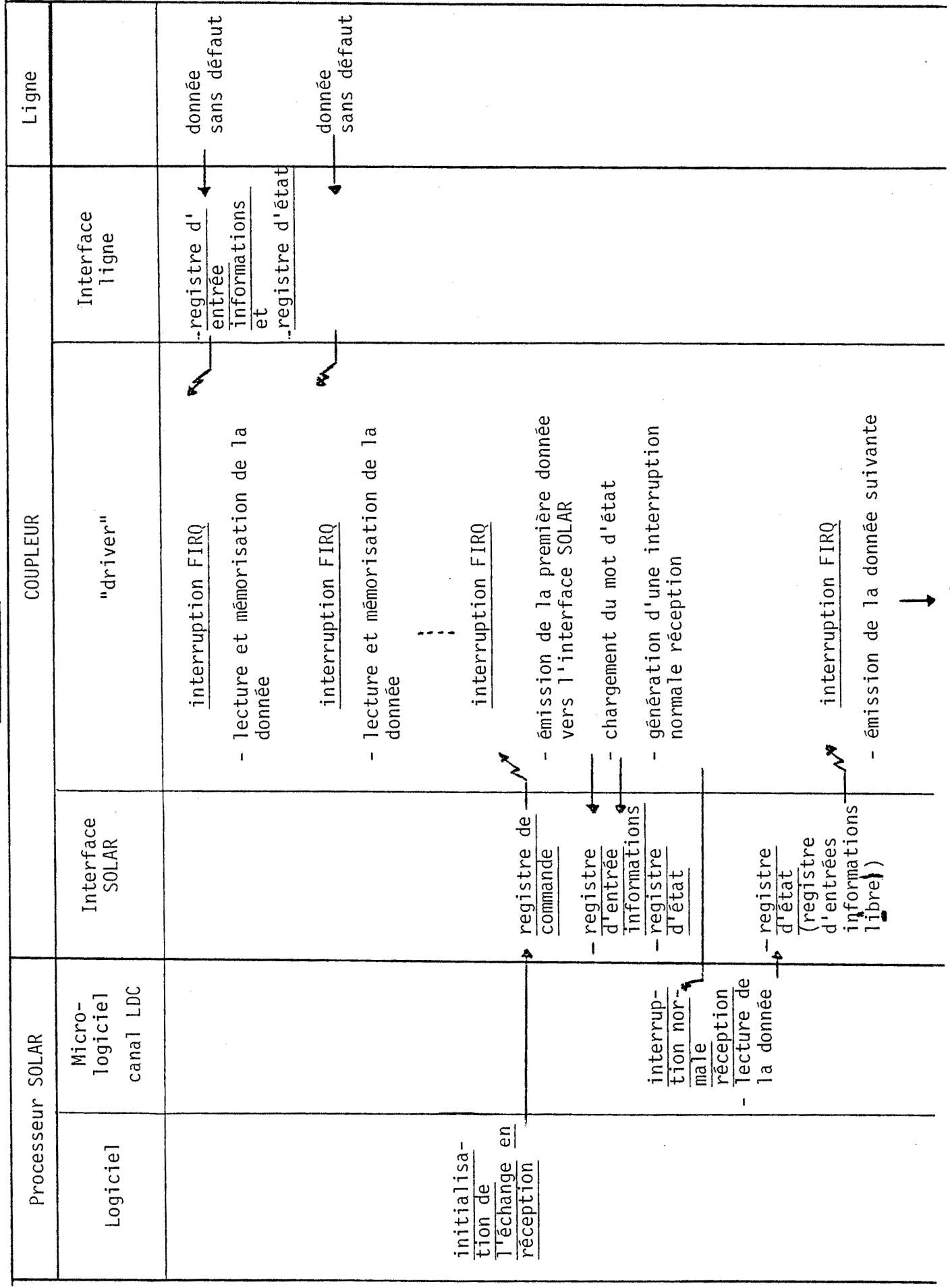
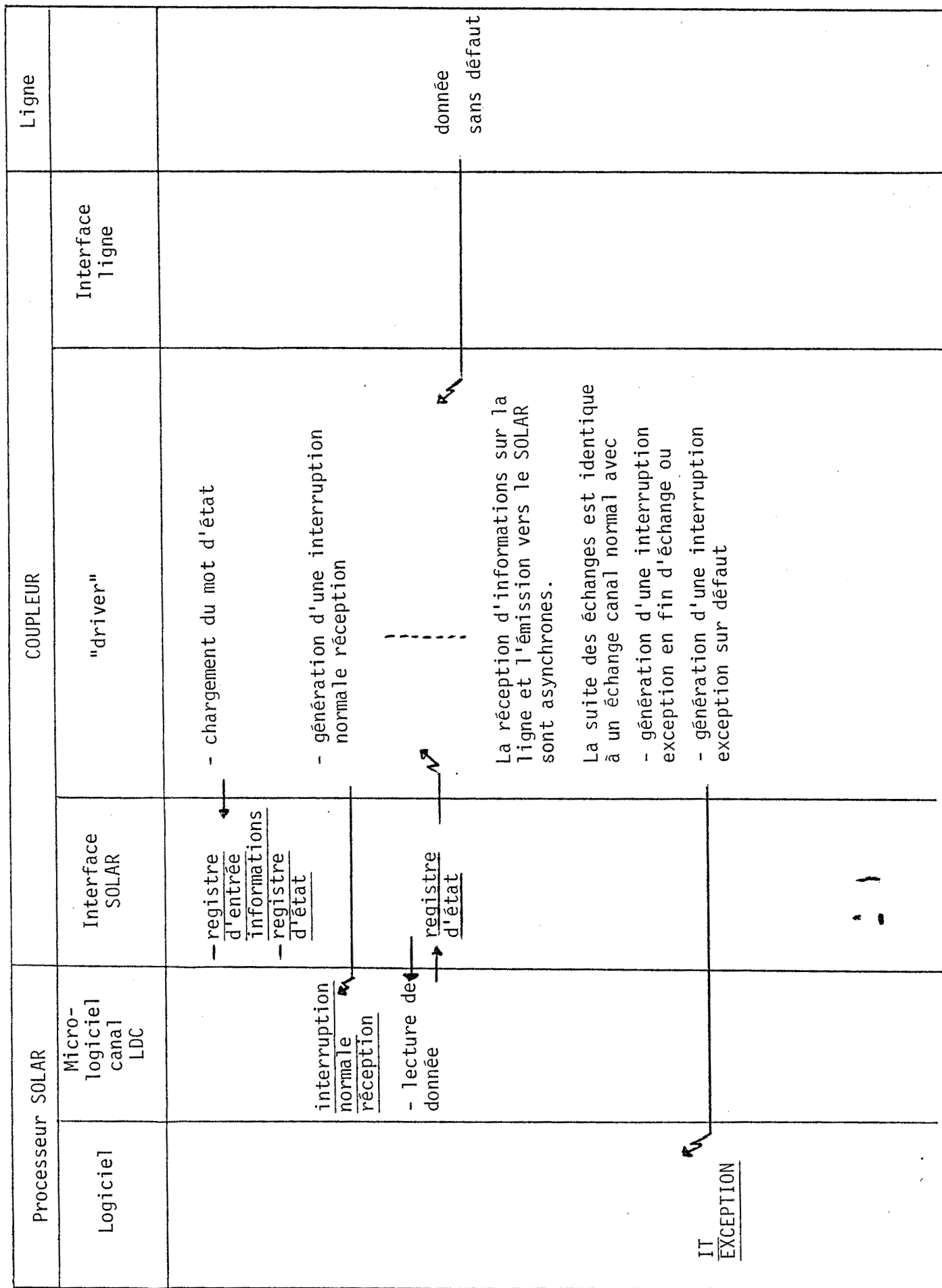


Figure 3.1. : détection de l'erreur de cadence dans le cas de la réception avec "bufferisation"

Cette fonction permet ainsi de mémoriser le début d'un message reçu avant l'initialisation de la réception par le logiciel SOLAR et de régler les problèmes de temps prohibitifs de "retournement" émission sur réception dans des applications en environnement half duplex.

Tableau 3.6. : réception avec "bufferisation"





1.1.5. La gestion des signaux modem

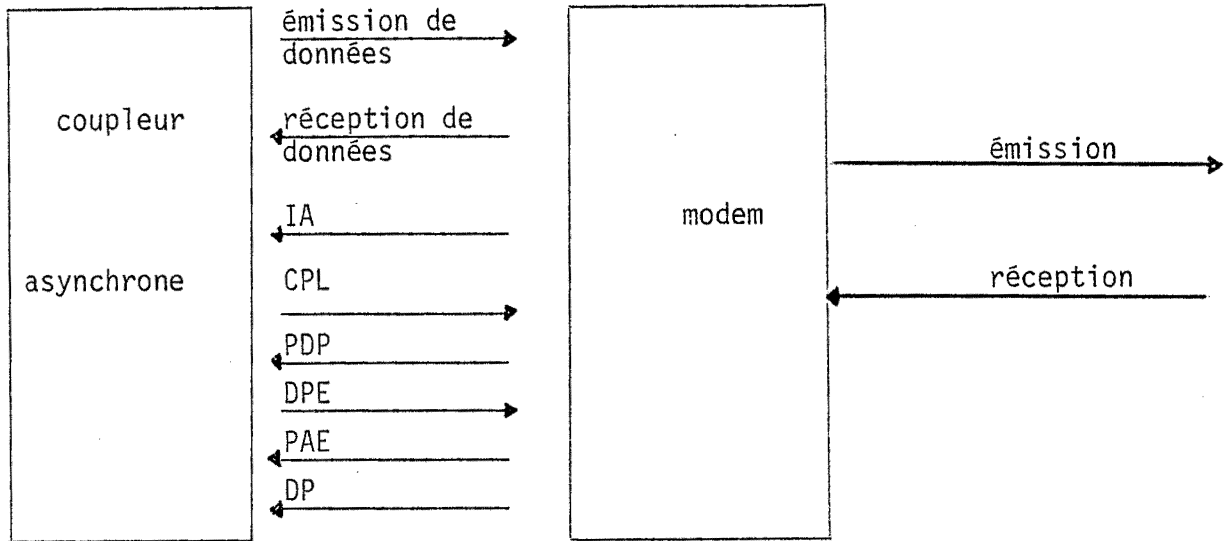


Figure 3.2. : les signaux modem

A l'état initial, l'ensemble coupleur asynchrone et modem est déconnecté de la ligne. La commande CPL permet la connexion à la ligne. Elle peut être réalisée de 2 façons :

a) le coupleur est raccordé au réseau commuté. La demande de connexion du correspondant est détectée par le modem qui positionne l'indicateur d'appel IA. Celui-ci est vu par le "driver" qui génère une interruption exception au logiciel SOLAR ; le logiciel SOLAR répond par la commande CPL. Quand l'échange est terminé, le SOLAR envoie la commande de déconnexion $\overline{\text{CPL}}$.

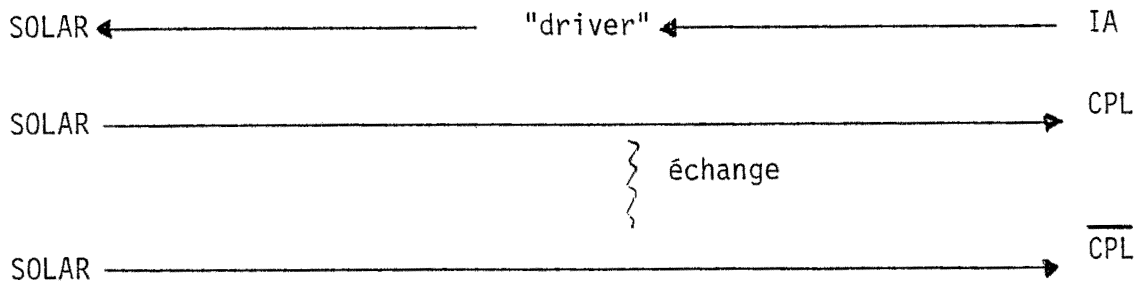


Figure 3.3. : gestion du CPL

b) le coupleur n'est pas raccordé au réseau commuté. La connexion à la ligne est faite par le logiciel SOLAR, systématiquement à la mise sous tension et reste permanente.

Les autres signaux modem (PDP, PAE, DP) et la commande DPE sont gérés par le "driver". Le logiciel SOLAR est totalement déchargé de cette gestion mais peut connaître l'état des signaux modem par lecture du mot d'état.

Le "driver" ne fait aucun échange tant que le signal PDP (poste de données prêt) n'est pas positionné. PDP représente l'état de la connexion à la ligne c'est-à-dire la réponse à la commande CPL réalisée par le logiciel SOLAR ; le "driver" ne prend pas l'initiative de réaliser la connexion.

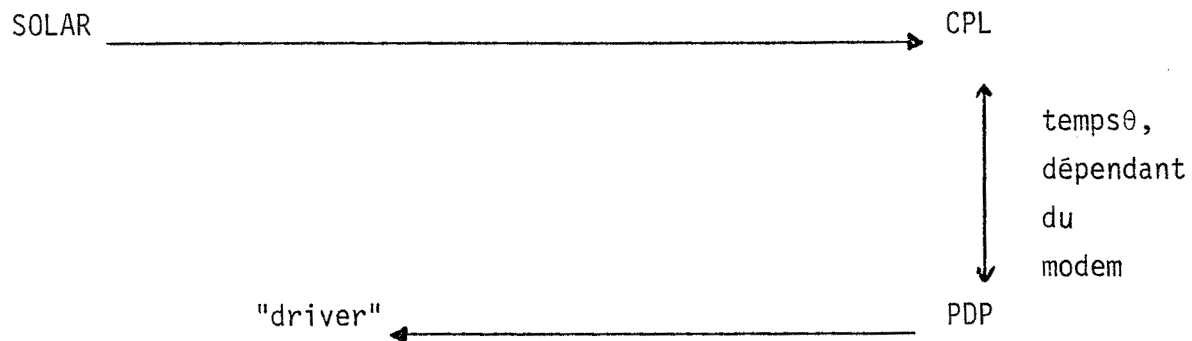


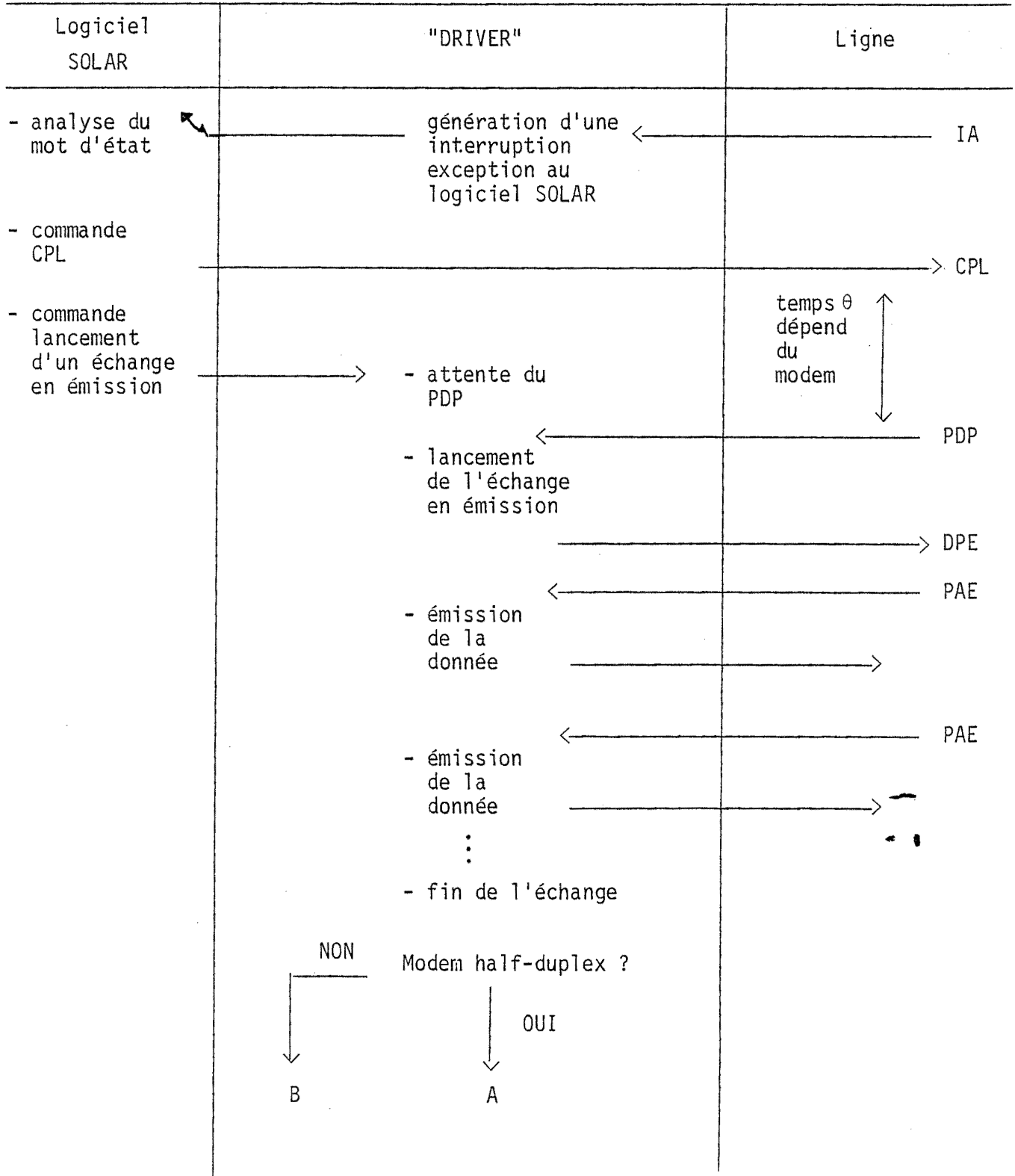
Figure : 3.4 : CPL et PDP

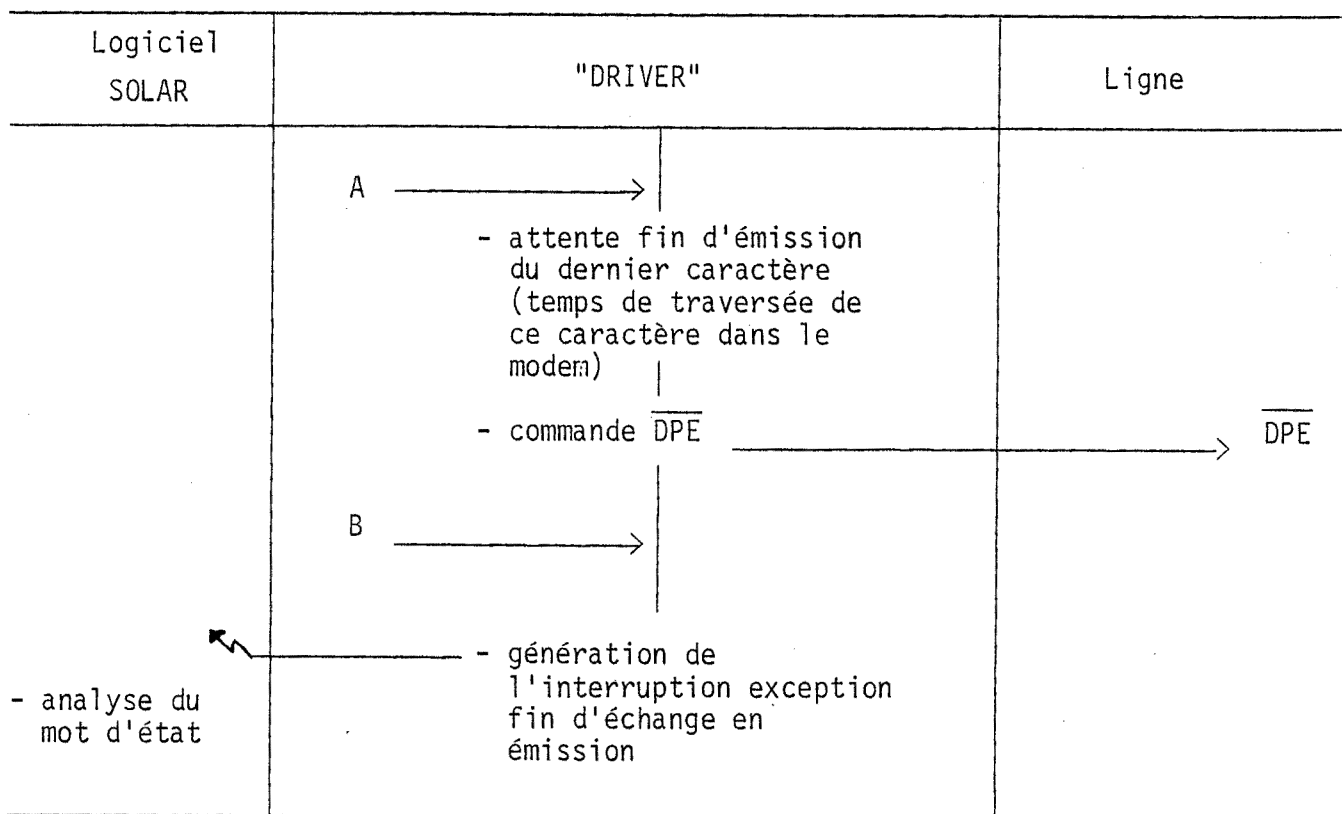
Echange en émission

Sur réception de la commande "lancement d'un échange en émission", le "driver" positionne DPE (demande pour émettre) et attend le signale PAE (prêt à émet-
tre) pour émettre un caractère.

La commande DPE est maintenue pendant toute la phase d'émission dans le cas d'un modem half-duplex ou reste permanente dans le cas d'un modem full duplex.

Tableau 3.7 : Gestion des signaux modem en émission.





Signal Détection de Porteuse (DP) : "Porteuse présente (DP = 1)" signifie qu'une réception est en cours.

La chute de porteuse peut être interprétée :

- = soit comme une fin d'échange,
- = soit comme un défaut si la porteuse chute en cours d'échange.

A la configuration du CMF, le logiciel peut choisir entre trois possibilités de gestion du bit DP, disponible dans le mot d'état SOLAR :

- = Il suit l'état de la porteuse et la chute de porteuse ne génère pas d'interruption exception ; c'est l'option prise par défaut par le CMF.
- = Il suit l'état de la porteuse et la chute de porteuse génère une interruption exception ; ce choix est fait par modification des paramètres du coupleur (cf. III.1.2.4).
- = Dans le cas d'une gestion de procédure, il n'a plus la même signification : il signale qu'un message procédure est à prendre en compte par le logiciel (cf. III.2.1.5).

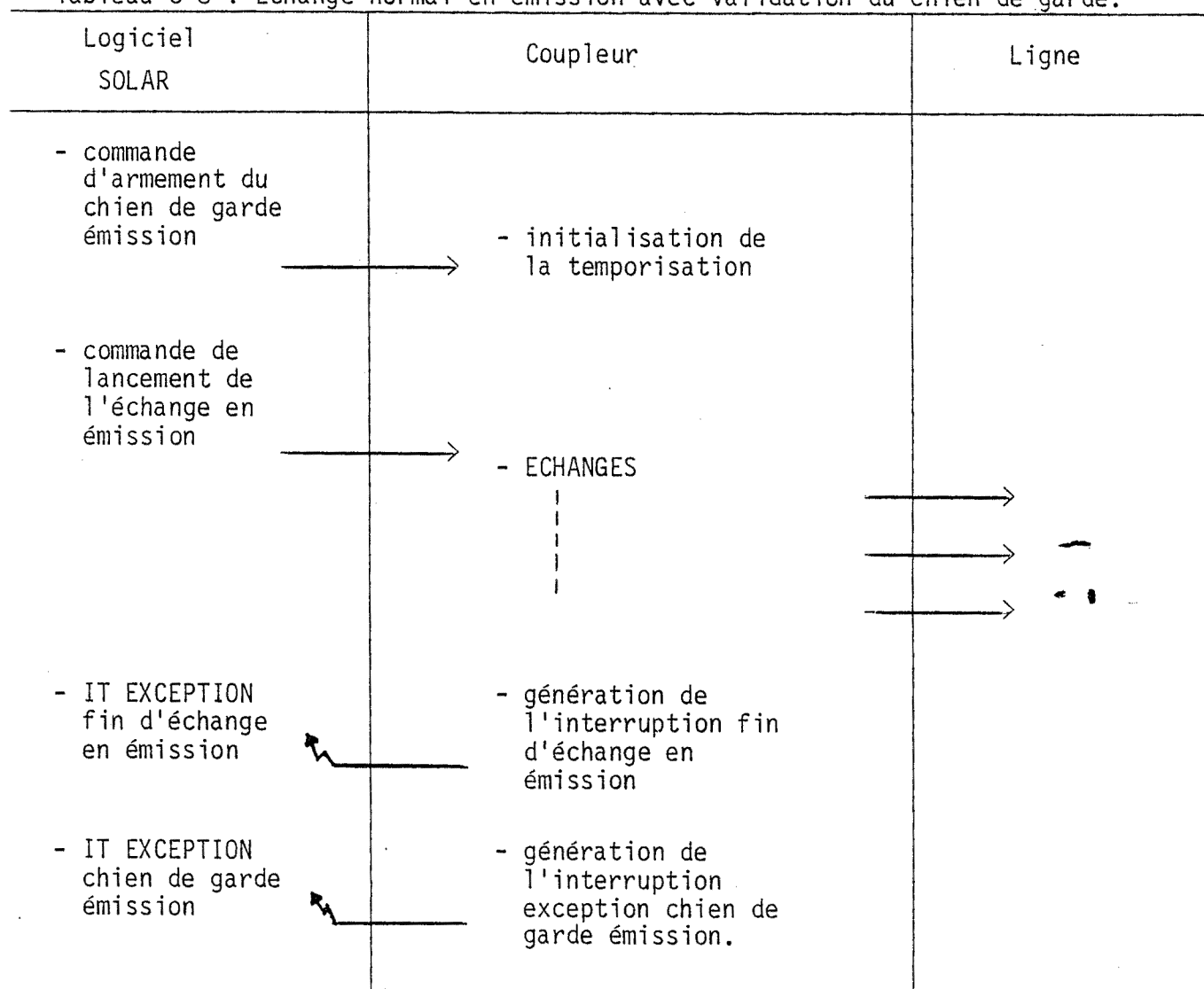
1.1.6. Les chiens de garde logiciels

L'utilisateur peut associer un délai maximum de réalisation à une séquence d'émission ou de réception. A cet effet, il dispose d'une temporisation définie en standard sur le CMF ou programmable par une commande d'initialisation (cf. III.1.2.4). L'arrivée à échéance de cette temporisation provoque une interruption exception au logiciel SOLAR pour lui signaler que le temps maximum imparti à cet échange s'est écoulé et donc que l'échange ne s'est pas déroulé normalement.

L'utilisateur dispose d'un chien de garde émission et d'un chien de garde réception par ligne.

a. Echange NORMAL

Tableau 3-8 : Echange normal en émission avec validation du chien de garde.



b. Echange avec défaut

Tableau 3-9 : Echange anormal en émission avec validation du chien de garde

Logiciel SOLAR	Coupleur	Ligne
- commande d'armement du chien de garde émission	→ - initialisation de la temporisation	
- commande de lancement de l'échange en émission	- ECHANGES	→
- interruption EXCEPTION chien de garde émission	↙ - génération de l'interruption exception chien de garde émission	

c. La gestion de cette temporisation est réalisée par la tâche de traitement du top 10 ms.

Sur l'arrivée à échéance de la temporisation (valeur nulle), le micrologiciel génère une interruption exception pour réveiller le logiciel SOLAR.

1.2. Les traitements d'informations

1.2.1. Les codes d'arrêt

Cette option n'est prise en compte que dans le cas d'un échange canal. Le canal micro-programmé du SOLAR fonctionne, en réception, dans l'un des modes suivants :

= arrêt sur compte d'octets nul : le logiciel SOLAR initialise l'échange en réception en donnant au canal le nombre d'octets à recevoir.

= arrêt sur code d'arrêt : le logiciel initialise l'échange en réception en précisant le code qui doit stopper l'échange. Pour le logiciel d'application, l'existence d'un code d'arrêt unique est souvent insuffisante ; pour cette seule raison, l'emploi du mode canal est rendu impossible et le fonctionnement en mode programmé prioritaire (réception avec interruption, caractère par caractère) s'impose pour pouvoir analyser tous les caractères reçus et les comparer aux codes d'arrêt.

Le coupleur CMF permet de supprimer cette contrainte. Il peut détecter, sur la voie réception de chaque ligne, un parmi 10 codes d'arrêts différents.

1.2.1. a. Validation et invalidation de la fonction "codes d'arrêt multiples"

La fonction de détection de codes d'arrêt multiples est inhibée à la mise sous tension. Elle doit être initialisée, pour une ligne de transmission donnée, par une commande (cf. II.3.1.b) précisant le nombre de codes d'arrêt puis par N autres commandes donnant les codes d'arrêt. Le premier code doit être le code d'arrêt reconnu par le canal micro-programmé. Il est possible, à tout moment, de modifier ces codes, toujours par une séquence d'initialisation.

EXEMPLE : Initialisation de 3 codes d'arrêt sur la ligne 1.

LAI '103 SIØ CØMMANDL1	< type de commande 2 pour le driver (0100) et 3 codes d'arrêt à initialiser.
LAI '18D SIØ CØMMANDL1	< commande 2 + 1 ^e code d'arrêt = 8D
LAI '10A SIØ CØMMANDL1	< commande 2 + 2 ^e code d'arrêt = 0A
LAI '1A0 SIØ CØMMANDL1	< commande 2 + 3 ^e code d'arrêt = A0

La fonction de détection de codes d'arrêt multiples est désactivée sur réception d'une commande avec nombre de codes nul ou par la mise hors tension du système.

La détection de la fin d'échange n'invalide pas la fonction car l'application peut enchaîner plusieurs réceptions avec codes d'arrêt identiques.

1.2.1. b. Le principe de fonctionnement

Les échanges en réception sont similaires à ceux décrits au chapitre III.1.2;

- = le canal est chargé de détecter le code d'arrêt de base,
- = le coupleur est chargé de détecter le code d'arrêt (parmi les N) qui a effectivement stoppé l'échange. Lorsqu'un code d'arrêt est détecté, le "driver" transmet au canal micro-programmé non pas le caractère reçu, mais le code d'arrêt standard pour que le canal puisse arrêter l'échange ; puis il charge dans le registre d'entrée informations le code qui a effectivement arrêté l'échange.

Cela implique que le logiciel d'application, sur réception de l'interruption exception de fin d'échange en réception, lise le registre d'entrée informations pour connaître le code qui a effectivement arrêté l'échange.

1.2.2. Les codes spécifiques

Certains périphériques possèdent un buffer en réception qui leur permet de visualiser ou d'imprimer les données reçues du calculateur à leur vitesse propre.

En cas de saturation ou de désaturation de ce buffer, le périphérique asservit le calculateur par l'émission de 2 caractères spéciaux débanalisés!

Pour gérer ce type de périphérique, le logiciel SOLAR travaille en réception caractère par caractère pour analyser chaque caractère.

Cette fonction est maintenant assurée par le CMF ; le logiciel est déchargé de la gestion de ces codes de saturation, désaturation et peut donc réaliser ses échanges réception en mode canal.

a. Initialisation de la fonction

Cette fonction est inhibée à la mise sous tension. Elle est implicitement activée, pour une ligne donnée, par la réception d'une commande d'initialisation des codes spéciaux de saturation, désaturation (cf. III.1.2.4) et ne sera désactivée qu'à la mise hors tension du système.

b. Principe de fonctionnement

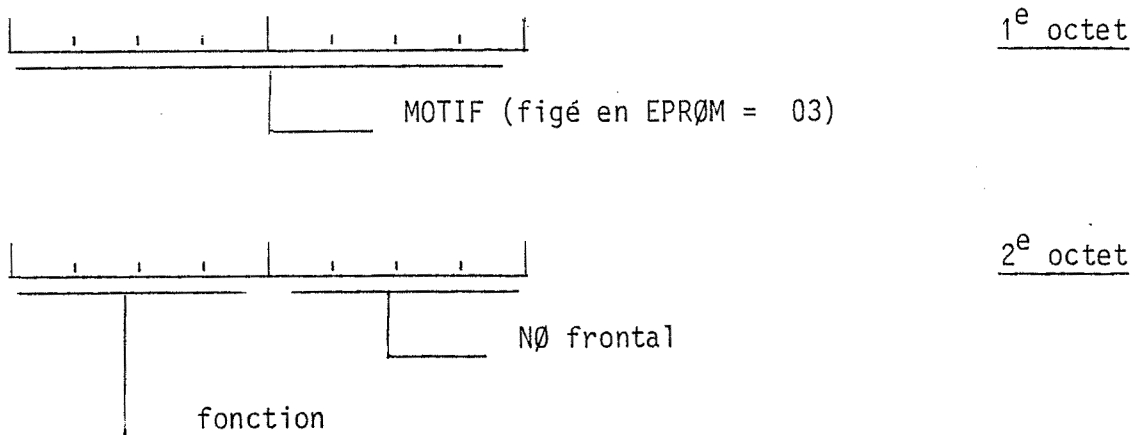
Sur réception du code de saturation, le "driver" bloque l'émission de données vers le périphérique. Il revalide l'émission de données vers le périphérique sur réception du code de désaturation.

Les 2 codes ne sont pas transmis au logiciel SOLAR.

1.2.3. Les fonctions de téléchargement et de télécommande

a. La fonction de télécommande sur ligne asynchrone (fonction TIP)

Si l'état TIP est actif (cf. II.4.2.2), tout caractère reçu sur une des lignes de transmission est analysé comme un éventuel caractère d'un message de télécommande. Les messages TIP sont construits de telle façon qu'il n'y ait pas de confusion possible entre une trame de télécommande et une trame d'informations. Une trame de télécommande est composée de 256 octets qui sont 128 fois la répétition des 2 octets.



Le module TIP du CMF piège et analyse toutes les informations transitant en réception sur la ligne. Si le premier octet reçu est formé d'un code préalablement défini appelé MOTIF et si le deuxième octet est formé du numéro de frontal (cf. II.4.1.2) affecté au CMF et d'une fonction codée, alors un compteur logiciel est incrémenté. Quand ce compteur passe par une valeur prédéterminée, le "driver" prend l'action précisée dans le message reçu pour la transmettre au processeur SOLAR.

Il est possible de coder 16 fonctions ; en fait, seulement les 4 commandes du PUC sont utilisées pour le moment ; les autres pourront être définies ultérieurement :

- ARRET
- INI
- LOAD le numéro de périphérique est prédéfini sur le coupleur
- RUN.

b. La fonction de télécommande sur ligne synchrone X25

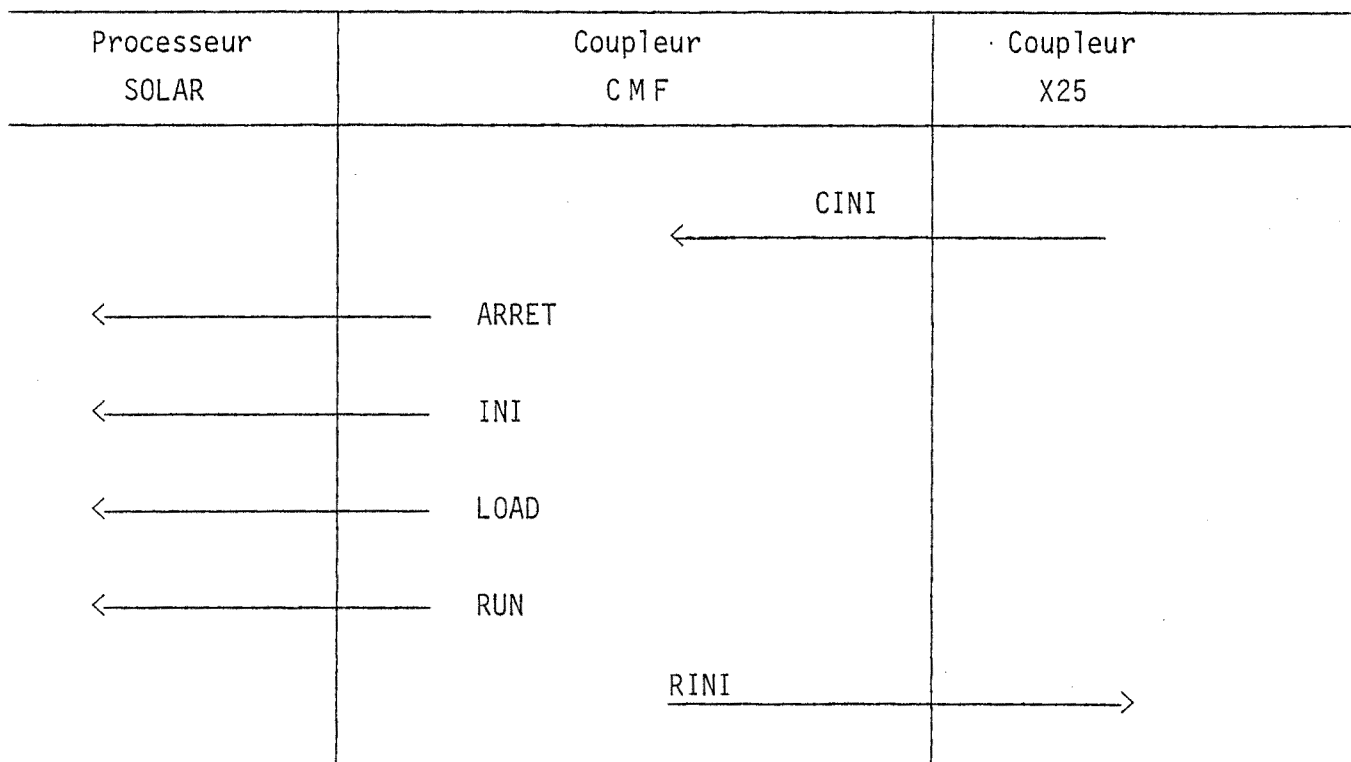
Le coupleur synchrone X25 SOLAR a la possibilité, par l'intermédiaire du CMF, de réinitialiser son SOLAR et de recharger l'application à partir d'un périphérique bootstrapable quelconque. Pour cela, le coupleur X25 envoie un signal (CINI) au coupleur CMF qui, sur détection de ce signal (dans sa tâche de fond), envoie les commandes :

- ARRET
- INI
- LOAD le numéro de périphérique ou de bootstrap est prédéfini sur le coupleur.
- RUN

au processeur en respectant les délais nécessaires entre chaque commande.

En fin de séquence, le coupleur CMF émet vers le coupleur X25 un signal (RINI) de fin d'exécution de la séquence d'initialisation.

Tableau 3-10 : Télécommande du coupleur X25



c. La fonction TRANSPAC

Dans certaines configurations, un SOLAR relié au réseau TRANSPAC doit pouvoir se connecter au réseau automatiquement, à la mise sous tension.

Si cette fonction est validée par un micro-interrupteur sur le coupleur CMF, le CMF envoie, à la mise sous tension, les commandes

ARRET

INI

LOAD bootstrap N° 5 (PRØM)

RUN

au processeur.

La PRØM TRANSPAC est intégrée à la PRØM de l'utilisateur et est limitée à 4 K octets ; elle contient un chargeur de système à travers le réseau.

Le micro-logiciel doit être capable de différencier une mise sous tension d'une initialisation du système. Il utilise pour cela une zone de 4 mémoires dont l'état est aléatoire à la mise sous tension car la mémoire vive du micro-processeur de gestion des fonctions de transmission n'est pas sauvegardée. A la mise sous tension, ces mémoires sont chargées avec des valeurs choisies en fonction de la probabilité quasiment nulle de les trouver en tant qu'état aléatoire.

1.2.4. La modification des paramètres du coupleur

Un des objectifs du CMF est d'assurer la compatibilité avec les logiciels existants ; ce qui amène des contraintes certaines au niveau de la configuration du coupleur. La nécessité de prendre en compte certaines fonctions nouvelles nous a amenés à proposer à l'utilisateur la commande de modification et de configuration des paramètres du coupleur ; cette commande existe sur les 2 lignes.

a. Nous trouvons sur les 2 lignes :

- la possibilité de charger du micro-logiciel par le SOLAR,
- la définition des codes spécifiques de saturation, désaturation de buffer de périphériques (cf. II.1.2.2),
- la modification des chiens de garde ; en standard, les chiens de garde émission et réception sont des valeurs moyennes des chiens de garde du coupleur ASX ; l'utilisateur a la possibilité, grâce à cette commande, de modifier ces temporisations.

b. Nous trouvons sur la ligne 1, l'initialisation de l'heure du système, — service offert à des procédures gérant la ligne 1 ou la ligne 2 (cf. III.2.4.10) pour la datation des trames reçues.

c. Sur la ligne 2, l'utilisateur peut préciser le mode de fonctionnement de la chute de porteuse. En standard, la chute de porteuse ne génère pas d'interruption exception au SOLAR, et l'utilisateur trouve l'état de la

porteuse dans le mot d'état. Par cette commande de modification de paramètres, il peut demander au "driver" de générer une interruption exception sur la chute de porteuse.

d. Des extensions fonctionnelles sont possibles :

La modification ou la configuration des paramètres du CMF se fait en mode canal. Le logiciel SOLAR exécute une commande de configuration (cf. II.3.1) après avoir initialisé la table d'échange en émission pour le canal avec un ou plusieurs messages de configuration.

Le message de configuration comprend :

- un octet précisant le type de configuration.
- les octets suivants donnent la configuration :
 - . Pour le chargement de logiciel :
 - octets 1,2 : adresse de chargement
 - octets 3,4 : nombre d'octets à charger
 - octets 5 à n : le micro-logiciel à charger.
 - . Pour les codes spécifiques :
 - octet 1 : code de saturation
 - octet 2 : code de désaturation
 - . Pour les chiens de garde logiciels :
 - octets 1,2 : chien de garde émission
 - octets 3,4 : chien de garde réception

(Les chiens de garde sont exprimés en top de 10 ms).
 - . Pour l'heure :
 - octets 1,2,3 : heure de système (en top de 10 ms)
 - . Pour la gestion de la porteuse :
 - octet 1 : mode de fonctionnement de la porteuse.

2. L'interface avec la procédure de l'utilisateur

Pour décharger le SOLAR, l'utilisateur peut intégrer dans le CMF, sur l'une ou sur les deux lignes de transmission, une procédure gérant tout ou partie des fonctionnalités prévues dans les niveaux 2, 3 ou 4 de l'ISØ :

- = gestion de l'émission/réception de trames de service ou de données,
- = traitement des blocs d'informations. Exemples :

- interprétation des champs de contrôle, traitement des caractères de synchronisation de la transmission...

Nous retrouverons l'organisation suivante :

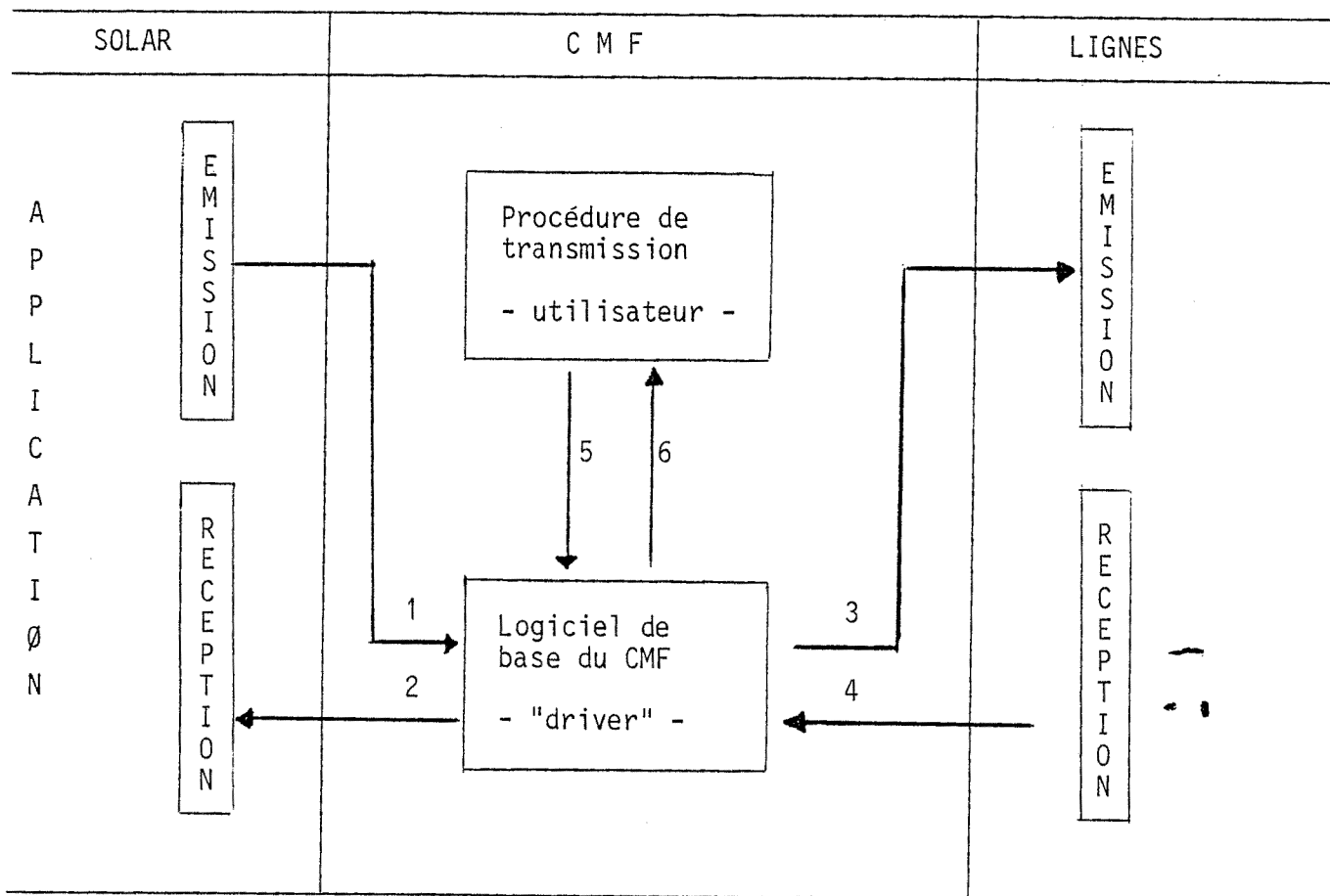


Figure 3-5 : Les chemins de données.

La procédure n'a pas accès aux interfaces matérielles SOLAR ou ligne ; c'est le "driver" qui gère ces interfaces ; tout événement ou toute demande de service transite par le "driver".

L'option "procédure de traitement" est validée par micro-interrupteur sur le CMF.

L'utilisateur peut implanter sa procédure dans une mémoire morte (EPROM), ou peut la charger en mémoire vive par la commande de chargement de logiciel (cf. III.1.2.4).

2.1. Les services offerts par le "driver"

Le travail d'une procédure consiste :

= à émettre ou à recevoir des données du SOLAR ou de la ligne avec contrôle du bon déroulement de ces transferts.

= à organiser en trames les données à émettre ou à interpréter les trames reçues.

Le "driver" offre à la procédure un certain nombre de services lui permettant de réaliser le premier travail.

La procédure peut ainsi :

- initialiser une voie émission ou une voie réception,
- recevoir des données de la ligne,
- émettre des données sur la ligne,
- recevoir des données du SOLAR,
- émettre des données vers le SOLAR,
- réveiller le SOLAR,
- gérer les signaux modem,
- armer ou désarmer un chien de garde logiciel,
- disposer d'une horloge temps réel,
- connaître l'heure du système.

Pour la réalisation de tous ces services, la procédure établit, avec le "driver", un dialogue précisant tous les paramètres du service demandé.

2.1.1. La requête d'initialisation d'une voie émission ou réception peut être demandée :

- = par la procédure qui doit préciser la voie à initialiser (émission ou réception),
- = par le logiciel SOLAR.

Dans les deux cas, le "driver" initialise la voie puis prévient la procédure de la fin de l'initialisation.

2.1.2. Réception de données de la ligne

Deux cas peuvent se produire :

- = la procédure attend une trame et initialise la réception avant que le premier caractère soit pris en compte par le driver.
- = la procédure n'est pas en attente de trame ; sur réception du premier caractère, le "driver" prévient la procédure pour qu'elle initialise la réception ; il continue à recevoir les données transitant sur la ligne.

L'interprétation des données est faite par la procédure lorsqu'une trame complète est reçue. La reconnaissance de fin de trame est propre à chaque procédure. Ce peut être :

- la réception d'une nième donnée,
- la réception d'un code particulier,
- la chute de porteuse,
- etc...

Pour recevoir des données d'une ligne, le "driver" doit :

- = éventuellement réveiller la procédure pour initialiser la réception,
- = stocker les données reçues sur la ligne et tester si la réception est correcte,
- = réveiller la procédure pour lui donner le compte rendu de fin de réception.

2.1.3. Emission de données sur la ligne

Pour émettre une trame sur la ligne, la procédure précise au "driver" :

- = l'adresse du "buffer" à émettre,
- = le nombre d'octets à émettre.

Après émission des données, le "driver" réveille la procédure pour lui donner un compte rendu de fin d'émission dans lequel l'utilisateur trouvera le nombre d'octets restant à émettre (nul si l'émission s'est déroulée correctement).

2.1.4. Réception de données du SOLAR

Le "driver" mémorise les données provenant du SOLAR. Sur réception de la commande de fin d'échange en émission, il prévient la procédure de la réception de données en provenance du SOLAR et lui fournit les paramètres qui lui permettront d'exploiter ce message.

Le "driver" n'accepte de nouvelle émission SOLAR qu'après traitement de l'émission courante par la procédure.

2.1.5. Emission de données vers le SOLAR

La procédure transmet au "driver"

- des paramètres décrivant le message à rendre au SOLAR : nombre d'octets et adresse du "buffer" à émettre,
- des paramètres permettant au logiciel d'application d'initialiser sa voie réception : ce peut être un numéro de tâche à réveiller et à initialiser...

Avant d'émettre les données, le "driver" réveille le SOLAR (avec l'interruption exception "message procédure disponible") pour lui préciser le nombre d'octets à recevoir et éventuellement d'autres informations permettant au logiciel d'application d'initialiser sa tâche réception. Le "driver" commence l'émission vers le SOLAR sur réception de la commande "lancement d'un échange en réception". Après l'émission des données, il rend un compte-rendu à la procédure dans lequel elle trouvera le nombre d'octets restant à émettre (nul si l'émission s'est déroulée correctement).

2.1.6. Pour réveiller le SOLAR, la procédure dispose du service "rendre un mot d'état au SOLAR". Elle peut, à un instant donné, avoir besoin de réveiller le SOLAR pour lui signaler l'arrivée d'un événement, d'une trame, d'une anomalie...

Le "driver" génère une interruption exception de réveil puis émet le ou les comptes-rendus, paramètres donnés par la procédure, par l'intermédiaire du registre d'entrée informations.

2.1.7. Gestion des signaux modem sur la ligne 2

A l'état initial, l'ensemble coupleur asynchrone et modem est déconnecté de la ligne. La commande CPL permet la connexion à la ligne ; elle peut être réalisée de deux façons :

a) Le coupleur est raccordé au réseau commuté. La demande de connexion du correspondant est détectée par le modem qui positionne l'indicateur d'appel IA. Celui-ci est vu par le "driver" qui réveille la procédure ; la procédure répond par une requête CPL adressée au "driver" et exécutée par le "driver". Quand l'échange est terminé, la procédure demande au "driver" la requête CPL de déconnexion.

b) Le coupleur n'est pas raccordé au réseau commuté ; la connexion à la ligne est demandée par la procédure, à la mise sous tension du système et reste permanente.

Les autres signaux modem (PDP, PAE, DP) et la commande DPE sont gérés par le "driver". L'état des signaux PDP, PAE ou DP peut être demandé par la procédure, à tout moment. De plus, sur disparition de la porteuse en réception, le "driver" réveille la procédure qui peut ainsi encadrer une émission/réception de trames par coupure de porteuse (procédure du type X25 asynchrone).

2.1.8. La fonction d'armement ou de désarmement d'un chien de garde permet à la procédure d'encadrer toute requête d'émission ou de réception pour en contrôler le bon déroulement. La procédure communique au "driver" la durée

du chien de garde définie en multiple de 10 ms. La durée nulle permet de désarmer le chien de garde. La procédure peut, à tout moment, interroger le "driver" pour connaître le temps restant avant échéance du chien de garde ; sur échéance, le "driver" réveille la procédure.

2.1.9. La fonction horloge temps réel permet à la procédure de calibrer un délai t pour gérer des actions exécutables à l'échéance de ce délai. Périodiquement (toutes les 20 ms), le "driver" incrémente un compteur et réveille la procédure. Ce compteur est réinitialisé par la procédure.

2.1.10. Heure du système

Le "driver" gère une heure système, initialisée par le logiciel d'application (cf. III.1.2.4) qui permet à la procédure de dater les trames ou les événements reçus.

2.2. Le dialogue entre la procédure et le driver

Une zone de mémoires débanalisées, réservée dans la mémoire du "driver", permet d'établir le dialogue entre la procédure et le "driver".

Dans cette zone mémoire baptisée BCR (blocs de contrôle des requêtes),

- le driver trouvera tous les paramètres nécessaires à l'initialisation et à l'exécution de la requête demandée par l'utilisateur,
- la procédure trouvera le compte-rendu de la requête.

Pour le "driver", une requête peut être :

- à exécuter.

Pour la procédure, une requête peut être :

- en cours d'exécution,
- exécutée ; dans ce cas, la procédure analyse le compte-rendu donné par le "driver".

Pour faciliter la recherche et éviter l'analyse systématique de tous les BCR, 3 mots d'états sont réservés pour chaque ligne et mémorisés, pour chaque requête :

- soit l'état requête à exécuter,
- soit l'état requête en cours d'exécution,
- soit l'état requête exécutée.

Chaque bit du mot d'état "requête à exécuter" est :

- positionné par l'utilisateur pour demander l'exécution de la requête,
- analysé par le "driver" pour exécuter la requête,
- remis à 0 par l'utilisateur après analyse du compte-rendu donné par le "driver" sur fin d'exécution de la requête.

Chaque bit du mot d'état "requête en cours d'exécution" est :

- positionné par le "driver" avant d'exécuter la requête,
- analysé par le "driver" pour lancer l'exécution de la requête (le "driver" analyse "requête à exécuter" et "requête en cours d'exécution" pour savoir s'il doit initialiser et exécuter la requête) et par l'utilisateur avant de demander l'exécution de la requête,
- remis à 0 par l'utilisateur après l'analyse du compte-rendu.

Chaque bit du mot d'état "requête exécutée" est :

- positionné par le "driver" sur fin d'exécution de la requête,
- analysé par l'utilisateur,
- remis à 0 par l'utilisateur après l'analyse du compte-rendu.

L'utilisateur doit encadrer ses phases de lecture ou d'écriture des mots d'états "requête en cours d'exécution" et "requête exécutée" par les instructions de masquage et de démasquage des interruptions IRQ et FIRQ pour ne pas être interrompu par le "driver"

ANDCC	#	\ AF	< Masquage des ITS FIRQ et IRQ
LDD		ETAL12	< lecture du M.E.2 de la ligne 1
QRCC	#	\ 50	< démasquage des ITS FIRQ et IRQ.

A la mise sous tension du système ou sur initialisation du coupleur, les blocs de contrôle des requêtes et les mots d'états sont remis à 0 par le "driver".

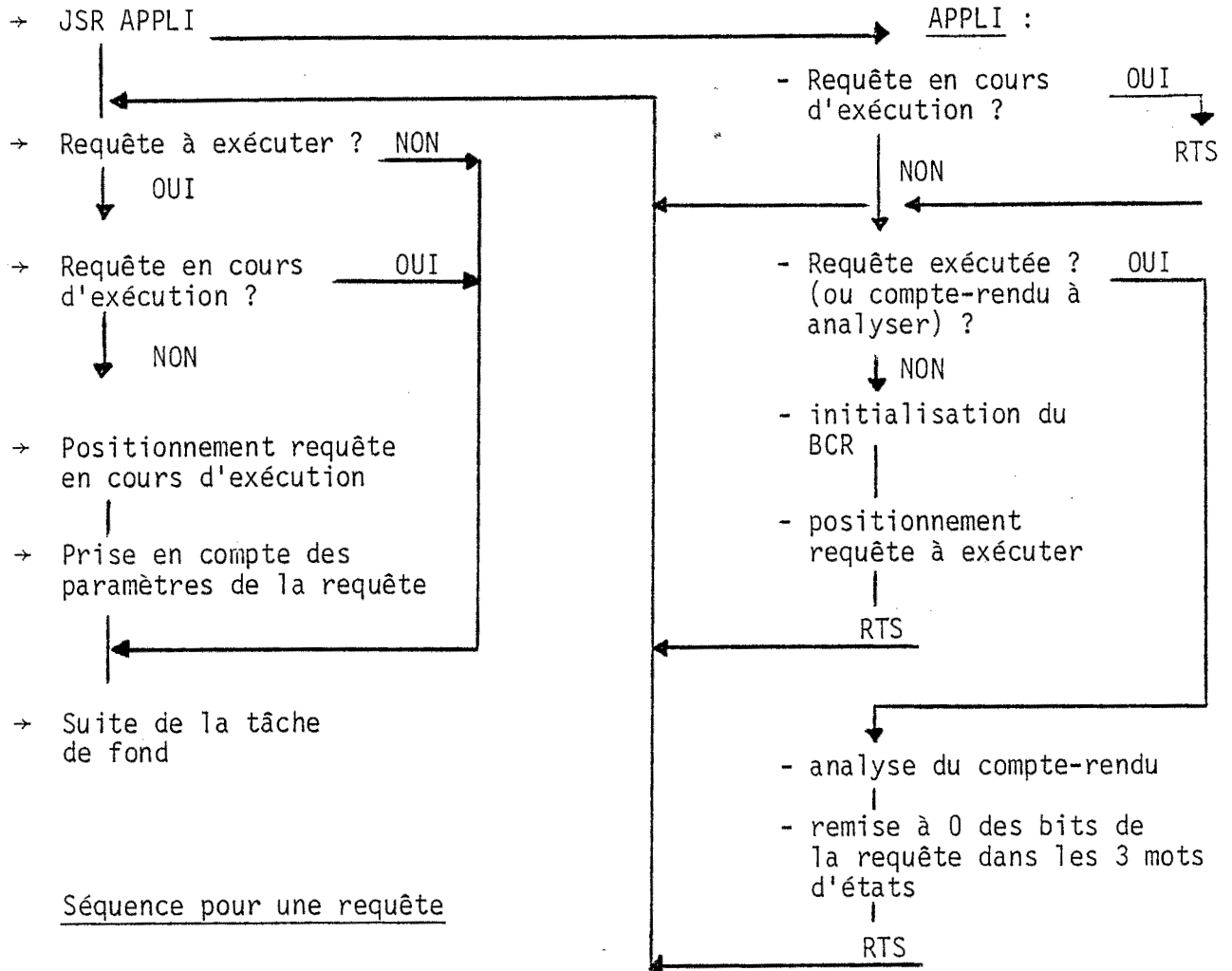
Le "driver" considère les programmes utilisateurs comme des sous-programmes. Il donne la main à l'utilisateur dans sa tâche de fond. L'utilisateur peut à ce moment-là initialiser une ou plusieurs requêtes et analyser un ou plusieurs comptes-rendus. Dans tous les cas, la séquence utilisateur doit finir par un RTS - fin de sous-programme - et cela afin de rendre la main au "driver" qui doit initialiser les mémoires de travail nécessaires à la prise en compte de la requête.

C'est ensuite dans les tâches de traitement des interruptions FIRQ et IRQ que le "driver" exécute les requêtes demandées. Sur fin d'exécution d'une requête, il rend à l'application un compte-rendu, passé par l'intermédiaire du BCR.

TACHE DE FOND
("DRIVER")

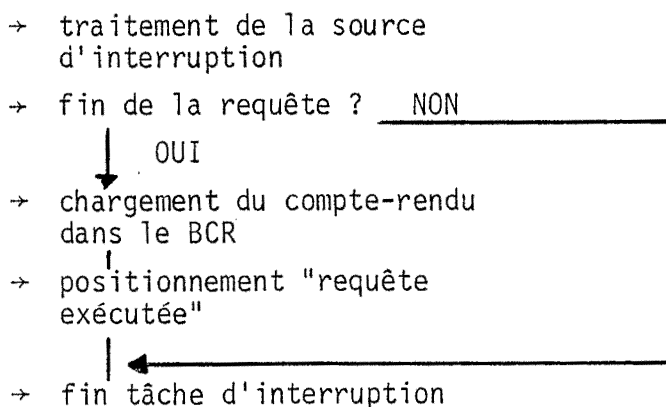
UTILISATEUR

TFOND :



TACHE FIRQ ou IRQ
("DRIVER")

Séquence pour une requête



L'utilisateur sera réveillé dans la tâche de fond (JSR APPLI) pour analyser ce compte-rendu.

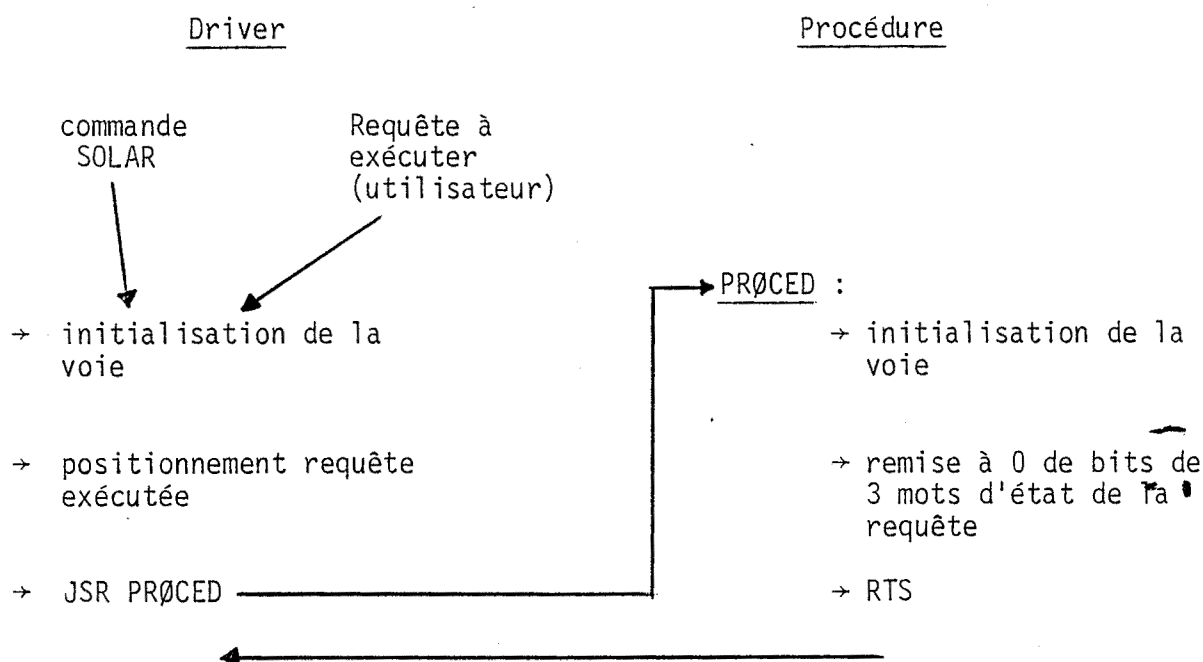
2.3. Description des blocs de contrôle

2.3.1. Initialisation des voies

Aucun bloc de contrôle n'est affecté aux 4 requêtes d'initialisation des voies émission ou réception sur la ligne 1 ou la ligne 2 car elles ne nécessitent pas le passage de paramètres au "driver". Ces requêtes peuvent être activées :

- par le logiciel SOLAR sur une commande d'initialisation de la voie émission ou réception ou sur une commande d'initialisation de la ligne,
- par la procédure.

Dans les deux cas, le "driver" initialise sa voie émission ou réception (ou les 2 dans le cas de l'initialisation ligne) puis signale à la procédure que la requête est exécutée ; à son tour, la procédure réalise l'initialisation de la voie émission ou réception et remet à 0 les bits des 3 mots d'états de la requête.



2.3.2. Gestion de l'heure

Un BCR de 3 octets contient l'heure du système exprimée en top de 20 ms. Cette heure est initialisée par le logiciel SOLAR (avec la commande de chargement des paramètres) et peut être lue à tout moment par la procédure, après masquage des interruptions IRQ (le top 10 ms génère une interruption IRQ au "driver" qui, dans sa tâche d'interruption, gère entre autres, l'heure). C'est pourquoi aucun bit n'est réservé dans les 3 mots d'état de requête.

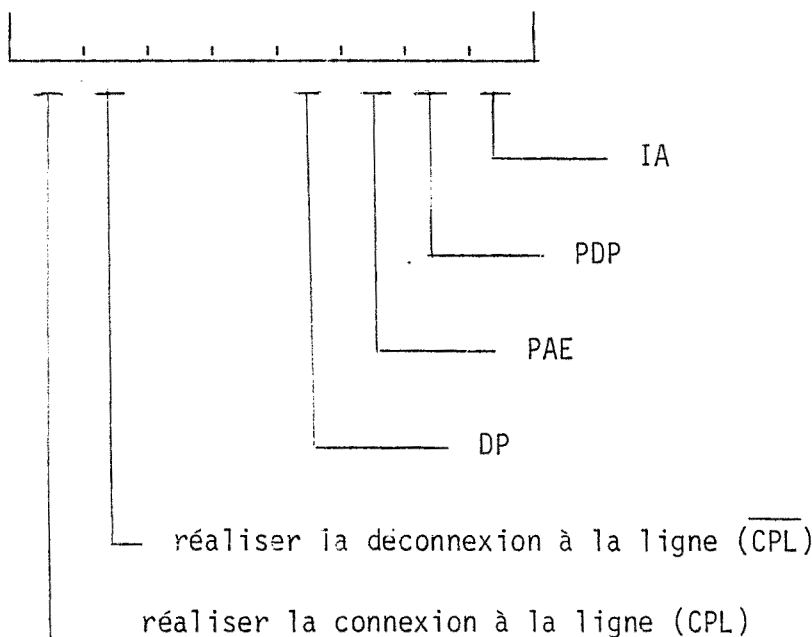
2.3.3. Horloge temps réel

Le BCR est en fait un indicateur donnant à la procédure le nombre de tops d'horloge temps réel en attente de traitement. Cette mémoire est réinitialisée par la procédure, après masquage des interruptions, dans sa phase de traitement (phase de prise en compte du compte-rendu HTR).

2.3.4. Signaux modem

Le "driver" rend, dans le BCR, l'état des signaux modem :

- sur demande de la procédure,
- sur disparition de la porteuse,
- sur apparition de l'indicateur d'appel pour prévenir la procédure qui active la requête CPL.



2.3.5. Chiens de garde logiciels

Pour armer le chien de garde émission ou le chien de garde réception, la procédure initialise les 2 octets, réservés par chien de garde dans la zone de BCR, avec une durée exprimée en unités de base de 10 ms.

Pour désarmer le chien de garde, il suffit que la procédure recharge ces 2 octets à 0.

Dans les deux cas, la procédure doit se protéger des interruptions IRQ et FIRQ en encadrant les séquences de chargement du chien de garde par les instructions de masquage, démasquage des interruptions.

L'état "requête exécutée" avertit la procédure que le chien de garde est arrivé à échéance.

2.3.6. Réception sur la ligne

Deux cas sont prévus :

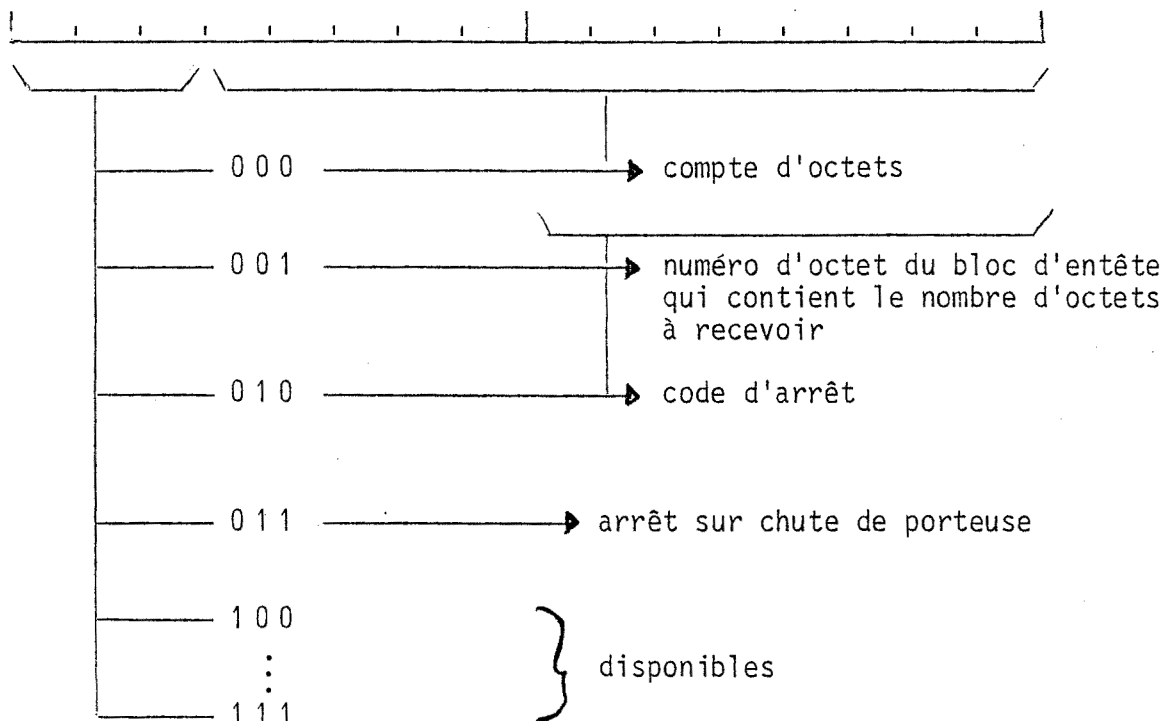
= la procédure n'a pas encore passé les paramètres de la réception au "driver". Dès réception d'un caractère, le "driver" mémorise le caractère et réveille la procédure pour qu'elle initialise la réception.

= la procédure initialise la réception par l'intermédiaire du BCR réception ligne.

Format du BCR

- octets 0,1 : pointeur de remplissage du "buffer" réception, initialisé par la procédure dans la phase d'initialisation du CMF ; ce "buffer" est géré en "buffer" circulaire par le "driver" ; ceci permet au "driver" de mémoriser les données reçues même si la réception n'a pas été initialisée.
- octets 2,3 : pointeur de vidage du même "buffer" initialisé dans la phase d'initialisation du système et incrémenté par la procédure. Ce pointeur permet au "driver" de gérer une éventuelle erreur de cadence logicielle en cas de saturation du "buffer" réception.

- octets 4,5 : mode de fonctionnement : arrêt de la réception sur
 - compte d'octets
 - code d'arrêt
 - chute de porteuse
 - ...



Le compte-rendu de l'échange est donné par le "driver" dans le même BCR :

octets 0,1 : pointeur de remplissage du "buffer" réception : il est incrémenté du nombre de caractères reçus.

octets 2,3 : pointeur de vidage du "buffer" ; il est inchangé. Le "driver" l'utilise en lecture seule.

octets 4,5 : mode de fonctionnement - inchangé.

octets 6,7 : nombre d'octets reçus.

octet 8 : compte-rendu d'échange ; le bit de poids forts est positionné à 1 si une erreur a été détectée ; dans ce cas, le type de l'erreur (parité, format, erreur de cadence matérielle ou logicielle) apparaît dans les autres bits. Le caractère erroné est rangé dans le "buffer" réception.

2.3.7. Réception de l'interface SOLAR

La procédure initialise, dans la phase d'initialisation du système, l'adresse du "buffer" réception.

Dès réception de la commande "lancement d'un échange en émission, le "driver" mémorise les données reçues dans ce "buffer".

Sur réception de la commande "fin d'échange" en émission et du dernier caractère, il transmet à la procédure un compte-rendu dans lequel elle trouvera :

- l'adresse du "buffer" réception
- le nombre d'octets reçus.

2.3.8. Emettre sur la ligne

Dans les octets 0,1 du BCR, la procédure donne l'adresse du "buffer" émission.

Dans les octets 2,3 du BCR, elle précise le nombre d'octets à émettre.

Le compte-rendu de l'échange est donné dans le même BCR :

- octets 0,1 : adresse du dernier caractère émis.
- octets 2,3 : nombre d'octets restant à émettre : nul si l'émission s'est déroulée correctement ; sinon c'est l'événement chien de garde qui doit réveiller la procédure.

2.3.9. Emettre vers l'interface SOLAR

Format du BCR :

octets 0,1 : adresse du "buffer" à émettre.

octets 2,3 : nombre d'octets à émettre.

octet 4 : nombre d'octets d'informations supplémentaires à rendre au SOLAR.

octets 5,... 9 : éventuellement les informations supplémentaires telles que numéro de tâche à réveiller ...

Sur détection de cette demande, le "driver" génère une interruption exception au logiciel SOLAR pour lui dire "MESSAGE PROCEDURE" (par l'intermédiaire du mot d'état SOLAR). Les éventuelles informations, telles que le nombre d'octets à recevoir ou le numéro de tâche à réveiller, sont transmises par le registre d'entrées informations.

Le logiciel SOLAR, sur réception de ces octets de paramétrage de l'échange, peut initialiser sa réception : préparation du canal, commande lancement de l'échange en réception. C'est sur réception de cette commande que le "driver" exécute l'échange avec le SOLAR.

Sur fin d'échange (commande réalisée par le canal), le "driver" passe un compte-rendu à la procédure, dans le BCR :

- octets 0,1 : adresse du dernier caractère émis.
- octets 2,3 : nombre d'octets restant à émettre : 0, si l'échange s'est déroulé correctement, sinon c'est le chien de garde émission qui doit réveiller la procédure.
- octets suivants : inchangés.

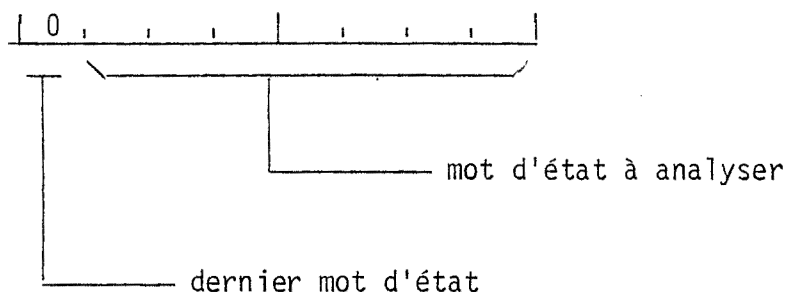
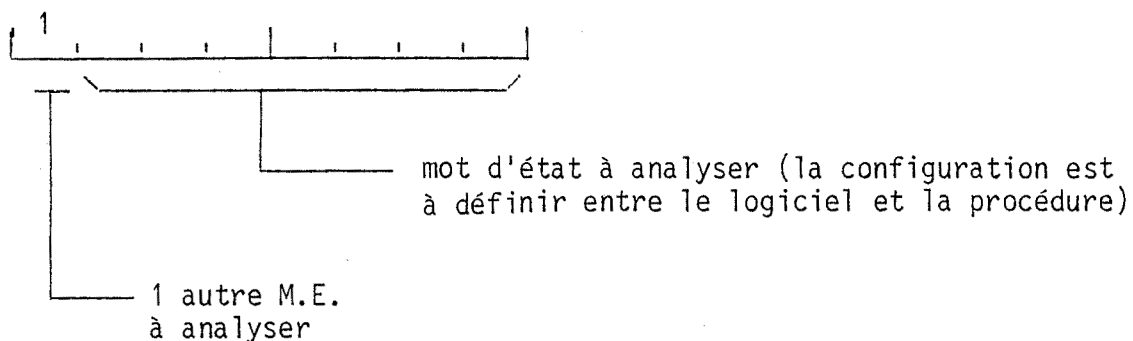
2.3.10. Envoyer un mot d'état au SOLAR

Format du BCR :

octet 0 : nombre d'octets d'état à rendre au logiciel SOLAR.

octets 1,... 9 : les octets d'état.

Le "driver" génère une interruption exception au SOLAR pour lui dire "MOT D'ETAT procédure" à analyser ; le ou les mots d'état sont transmis par le registre d'information sous la forme :



En fin d'exécution de requête, la procédure trouvera dans le BCR :

- octet 0 : le nombre d'octets d'état restant à émettre ; nul si l'échange s'est correctement déroulé, sinon c'est le chien de garde émission qui doit réveiller la procédure.
- autres octets : inchangés.

2.4. Traitements particuliers

2.4.1. Les codes spécifiques

Dans le cas où les codes spécifiques ont été au préalable définis, c'est le "driver" qui s'occupe de cette gestion : sur réception du code "buffer" plein, le "driver" bloque l'émission sur la ligne. L'émission est revalidée sur réception du code de désaturation.

2.4.2. Le traitement des codes d'arrêt

Cette notion disparaît car pour rendre des informations au SOLAR, le "driver" réveille le logiciel pour lui donner le nombre d'octets à attendre. Le canal réception, dans ce cas, fonctionne en arrêt sur compte d'octet nul.

CHAPITRE IV

LES FONCTIONS PUPITRE

	<u>PAGES</u>
IV - <u>LES FONCTIONS PUPITRE</u>	
1 - <u>LE DIALOGUE</u>	98
2 - <u>LES FONCTIONS PROPOSEES</u>	100
3 - <u>LA PRISE EN COMPTE DES INFORMATIONS RECUES SUR LA LIGNE</u>	102
4 - <u>LES CHEMINS DE DONNEES ET LES INTERFACES</u>	104
4.1 <u>Chemin de données entre la console de service et le micro-processeur</u>	104
4.2 <u>Chemin de données entre le micro-processeur et le processeur SOLAR</u>	104
4.3 <u>Récapitulatif</u>	105

IV - LES FONCTIONS PUPITRE

Actuellement, les fonctions pupitre sont réalisées par l'intermédiaire d'un pupitre avec clés d'accès à la machine interne du SOLAR ; les fonctions possibles sont des fonctions de très bas niveau d'adressage de mémoire ou de registre.

Il s'est avéré indispensable d'offrir à l'utilisateur des moyens de mise au point et de maintenance plus sophistiqués.

Ainsi, avec le C M F, les fonctions pupitre sont demandées par conversationnel sur un organe de dialogue connecté sur l'une des deux lignes asynchrones. Toutes les fonctions activables avec le pupitre POP sont reprises et le C M F assure, de plus, des fonctions plus sophistiquées telles que l'édition de zones système du moniteur d'entrées - sorties (IOCS) ...

A l'initialisation du système, les fonctions pupitre sont activables, par défaut, sur la ligne 1.

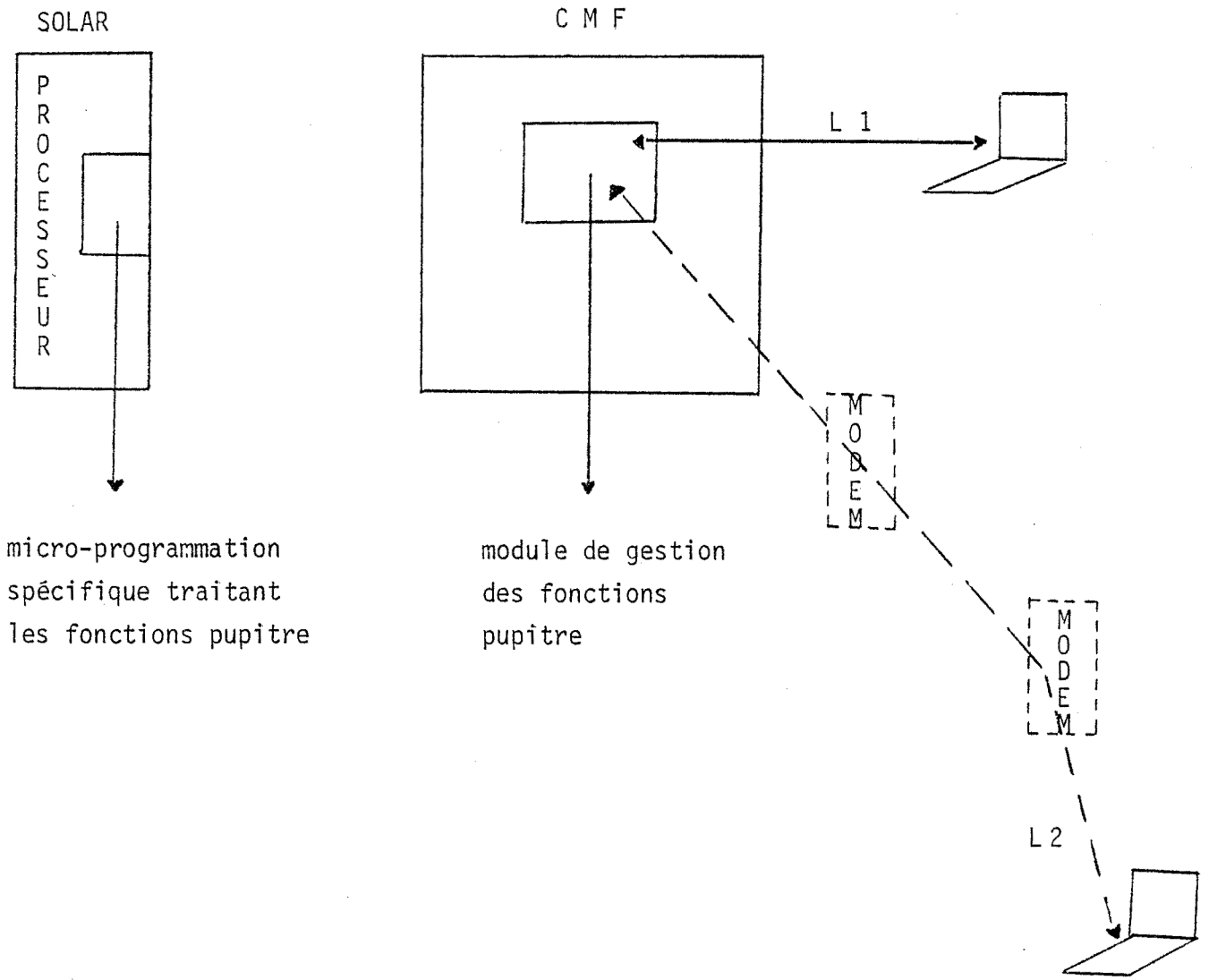


Figure 4.1 : Environnement des fonctions pupitre.

La ligne 2 est une ligne asynchrone munie d'une interface V24 complète et permet l'exécution de fonctions de télémaintenance.

Actuellement, la maintenance est assurée par SEMS par intervention directe sur la machine : cela nécessite le déplacement de l'inspecteur de maintenance sur le site client et pénalise le client en délai et en coût d'intervention.

SEMS s'attache à résoudre ce problème et veut assurer le déport des fonctions de maintenance dans des centres de maintenance. Le CMF permet d'envisager, grâce aux fonctions pupitre accessibles par ligne asynchrone connectée à un modem, une maintenance à partir d'un site de maintenance SEMS dans des conditions optimales.

1) LE DIALOGUE

Il respecte les règles de base des dialogues SOLAR et permet à l'opérateur de demander toutes les fonctions réalisées avec le pupitre opérateur POP, ainsi que des fonctions plus élaborées (telles que édition d'une zone mémoire ...) dans lesquelles le micro-logiciel du coupleur se charge de toutes les incrémentsations, décrémentsations, indirections sur les adresses mémoire ou sur les numéros de registre.

Dès réception de la première commande, le micro-logiciel mémorise le fait qu'une console est connectée sur la ligne ; cela afin de pouvoir éditer, ultérieurement, des messages d'intérêt général.

Le langage utilise les caractères disponibles, sur un clavier de machine à écrire, c'est-à-dire :

- les 26 lettres de l'alphabet
- les 10 chiffres
- quelques caractères spéciaux

Il respecte les règles des dialogues SOLAR :

- toute commande se termine par le code (RC)
- le caractère (←) annule une phrase en entrée de commande
- le caractère (↑) annule le caractère précédent
- le "break" arrête la commande en cours d'exécution

Une différence intervient, cependant, au niveau de la représentation des nombres, en entrée de commande. En effet, tout nombre décimal doit être précédé des signes + ou - ; sinon il est interprété comme un nombre hexadécimal ; tout nombre hexadécimal est précédé ou non du caractère (').

Toute phrase est composée :

- = d'un mot clé : c'est une chaîne de caractères reconnue par le micro-logiciel de gestion du pupitre et qui spécifie le type de fonction à réaliser.

= éventuellement, d'un

ou plusieurs nombres : ce sont les paramètres de la fonction qui précisent une adresse mémoire, un contenu de mémoire ou de registre.

Le caractère séparateur des différents éléments d'une phrase est le caractère espace.

Le micro-logiciel de gestion des fonctions pupitre analyse la demande de l'opérateur, en vérifie la validité (il édite un éventuel message d'erreur si une erreur est détectée), dialogue avec le processeur SOLAR puis édite les résultats de cette fonction sur l'organe de visualisation.

2) LES FONCTIONS PROPOSEES

L'opérateur peut :

- lire ou modifier le contenu d'une mémoire avec passage ou non à la mémoire précédente ou à la mémoire suivante.
- lire ou modifier le contenu d'un registre avec passage ou non au registre précédent ou au registre suivant.
- éditer une zone mémoire dont les adresses de début et de fin de zone sont données en paramètres avec la clé.
- translater des données d'une zone mémoire vers une autre zone mémoire ; les adresses source, destination et le nombre d'octets sont donnés en paramètres de la clé.
- positionner un ou plusieurs points d'arrêt
- identifier les points d'arrêt positionnés
- supprimer un ou plusieurs points d'arrêt
- supprimer tous les points d'arrêt
- réaliser les fonctions du pupitre de commande :
 - lancement du programme en continu
 - arrêt programme
 - initialisation du SOLAR
 - chargement du bootstrap à partir d'un périphérique Bootstrapable dont le numéro est donné en paramètre.
- exécuter un ou plusieurs pas de programme avec édition, à chaque pas de programme, du contenu de tous les registres.
- éditer des zones mémoire système telles que :
 - les contextes des tâches immédiates (PSTH), ces tâches sont les tâches associées aux interruptions de programme, aux alarmes machine et au défaut secteur
 - les contextes des tâches différées (PSTS)

- les tables d'échange canal (CCB) ; ces tables permettent au logiciel de passer les paramètres de l'échange au canal
- les tables d'unité physique (TUP) associés aux périphériques vus par le moniteur d'entrées-sorties (IOCS).
- les fils du SCHEDULER, option micro-programmée du processeur SOLAR permettant de gérer de 1 à 128 tâches logicielles

ASTF : file de 8 mots de 16 bits dont chaque bit i mis à 1 indique que la tâche logicielle de priorité i a été demandée par l'instruction ARM

ESTF : file de 8 mots de 16 bits dont chaque bit i mis à 0 indique que la tâche logicielle de numéro i est masquée

RSTF : file de 8 mots de 16 bits dont chaque bit i mis à 0 indique que la tâche logicielle de numéro i nécessite une action du moniteur (par ex : un chargement en mémoire de cette tâche)

- éditer l'indice d'évolution de la mémoire morte du coupleur multi-fonctions - ceci afin de vérifier la compatibilité matériel, micro-logiciel.
- dialoguer avec le processeur 0 ou le processeur 1, dans le cas d'une configuration bi-processeur.
- activer les fonctions pupitre sur la ligne 1 ou sur la ligne 2. A l'initialisation du système, les fonctions pupitres sont activables sur la ligne 1.
- exécuter les opérations d'addition et de soustraction pour faciliter la tâche du programmeur dans le calcul d'adresses, par exemple.

3) LA PRISE EN COMPTE DES INFORMATIONS RECUES SUR LA LIGNE

Les informations qui transitent sur la ligne (en émission ou en réception) sont soit des informations système, soit des informations pupitre : c'est le micro-logiciel qui se charge d'aiguiller ces données vers le logiciel SOLAR ou vers le module de gestion du pupitre.

Les données sont aiguillées vers le module pupitre si les 4 conditions ci-dessous sont réalisées :

- Le processeur est en mode ARRET et le pupitre de commande n'est pas verrouillé (ceci pour des raisons de sécurité). La possibilité (validation matérielle de l'option "POP/RUN") de réaliser les fonctions pupitre avec un processeur en mode "MARCHE" est cependant laissée à l'opérateur et lui permet d'exécuter des fonctions en temps réel. Des indicateurs matériels précisent au micro-logiciel l'état du processeur et du pupitre de commande.
- La fonction pupitre est validée sur la ligne. A la mise sous tension, elle est validée sur la ligne 1. Cette option peut être modifiée par la fonction pupitre : " choix de la ligne de dialogue pour la fonction pupitre ".
- La voie émission de la ligne n'est pas occupée par une émission canal. Le module pupitre utilise la voie émission de la ligne pour réaliser le mode écho des caractères frappés par l'opérateur. En conséquence, il ignore les caractères reçus sur la ligne quand une émission en mode canal est en cours (sinon il y aurait un mélange entre le texte en cours d'émission canal et l'écho des caractères reçus).
- Une fonction pupitre n'est pas déjà en cours de traitement.

paramètres du système ↓ aiguillage de la donnée	ETAT du pupitre de commande		émission canal en cours (sur la ligne)	fonction pupitre activée sur la ligne	fonction pupitre en cours d'exécution	option " POP RUN "
	LOCK	RUN				
aiguillage de la donnée reçue vers le logiciel SOLAR (ou vers la procédure de gestion de ligne asynchrone éventuellement intégrée sur le coupleur)	Y	Y	Ø	Ø	Ø	Ø
	Y	N	Ø	Ø	Ø	Ø
	N	Y	Ø	Ø	Ø	N
103						
aiguillage de la donnée vers le module "PUPITRE"	N	N	N	Y	N	Ø
	N	Y	N	Y	N	Y
la donnée est ignorée	N	N	Y	N	Y	Ø
	N	N				Ø
	N	N				Ø
	N	Y				Y
	N	Y				Y
	N	Y				Y

Ø : état indifférent

Tableau 4.1 : aiguillage d'une donnée reçue en fonction de l'état du système

4) LES CHEMINS DE DONNEES ET LES INTERFACES

Les caractères frappés par l'opérateur sont interprétés par le module "PUPITRE" qui, après recherche de la fonction, en demande l'exécution au processeur SOLAR. Le compte-rendu du processeur est mis en forme par le module "PUPITRE" pour être édité par l'organe de visualisation.

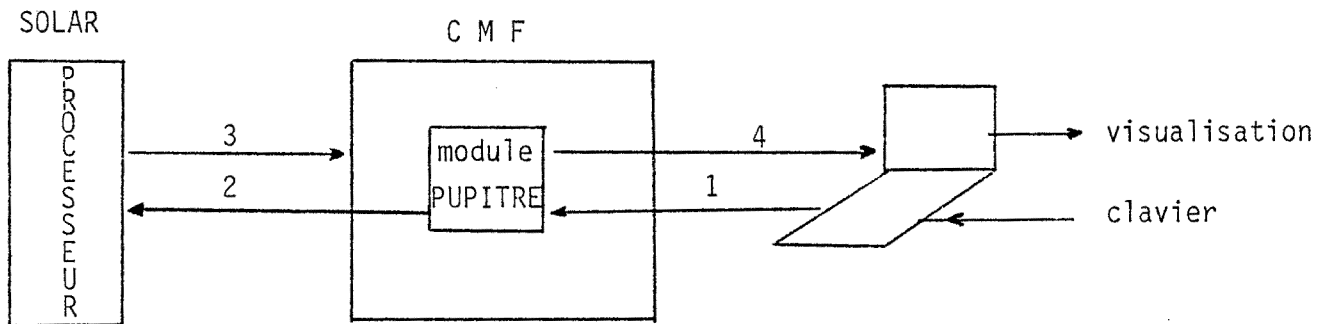


figure 4.2 Les chemins de données

4.1. Interface entre la console de service et le micro-processeur

Chemin 1 et 4 : le dialogue opérateur-micro-processeur transite par la partie émission/réception de l'ACIA de la ligne concernée et utilise les circuits émission/réception de cette ligne.

La partie émission est validée sans interruption pour

- réémettre le caractère frappé par l'opérateur
- éditer le listing des résultats.

4.2. Interface entre le micro-processeur et le processeur SOLAR

Le dialogue entre le micro-processeur et le processeur SOLAR est dépendant du fonctionnement du processeur SOLAR dans la phase de traitement des fonctions du pupitre POP. L'interface existante a été respectée pour éviter toute modification du micro-programme gérant les fonctions pupitre sur les processeurs SOLAR :

5 registres sont vus par le micro-programme du processeur SOLAR :

- a) → registre d'entrées informations dans lequel le micro-programme SOLAR trouve l'adresse de la mémoire ou le numéro de registre ou la valeur à charger dans la mémoire ou le registre concerné par l'échange.
- b) → registre d'entrée mot d'état. Chaque bit de ce registre indique au micro-programme SOLAR la fonction à réaliser.
- c) → registre de sortie informations par lequel le micro-programme SOLAR rend au micro-logiciel le contenu de la mémoire ou du registre.
- d) → registre de sélection par lequel le micro-programme SOLAR rend au micro-logiciel l'adresse mémoire ou le numéro de registre.
- e) → le registre de commande en sortie dans lequel le processeur SOLAR précise son mode de fonctionnement (travail en adressage mémoire ou registre, en mode point d'arrêt ou pas).

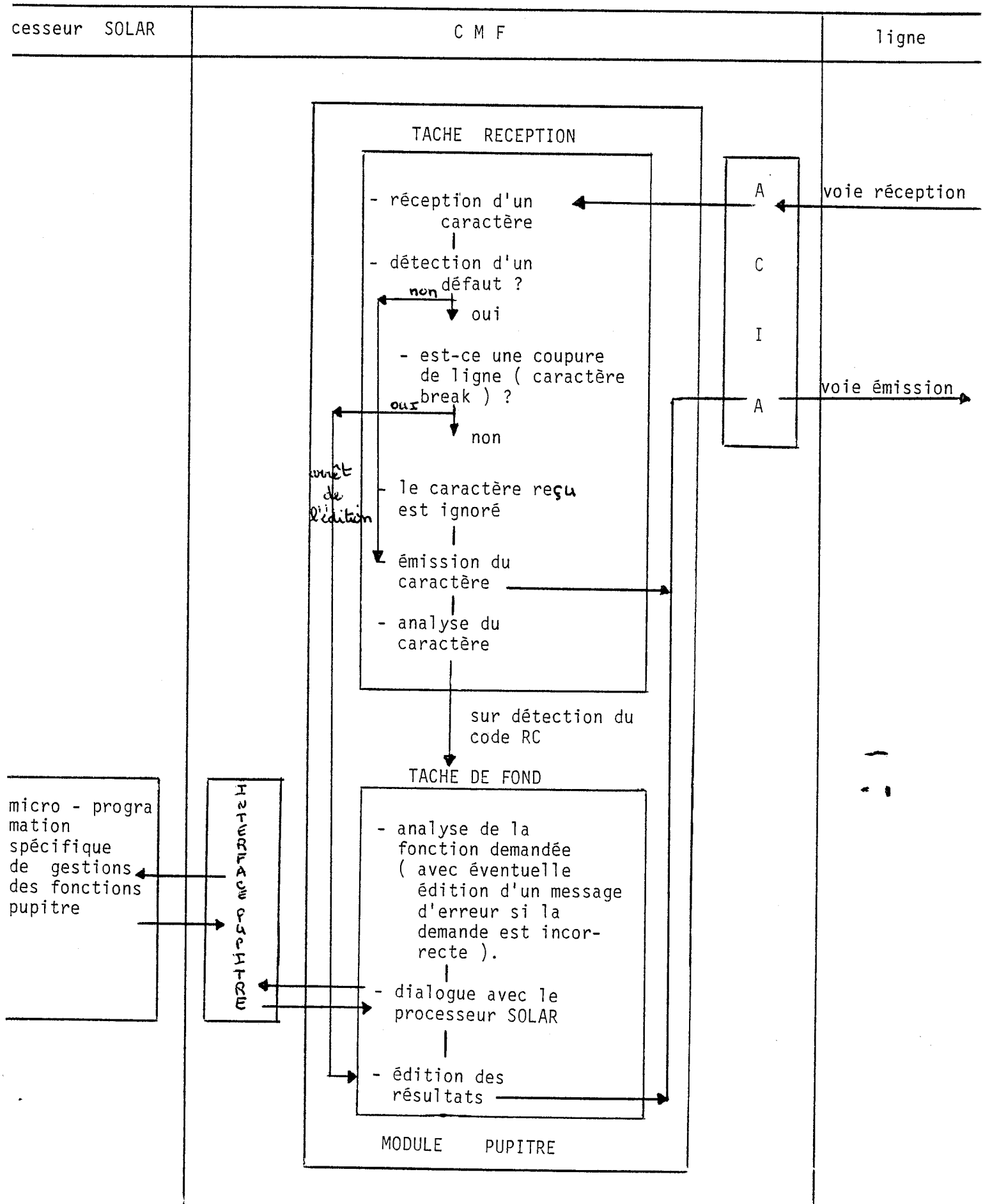
4.3. Récapitulatif

La figure IV.3 montre le cheminement et la prise en compte des informations reçues par le module pupitre.

CHAPITRE V

LES FONCTIONS TEMPS REEL

Figure IV.3 : Module pupitre



	<u>PAGES</u>
V - <u>LES FONCTIONS TEMPS REEL</u>	
1 - <u>LE CHIEN DE GARDE CALCULATEUR ET LES ENTREES TOR</u>	109
1.1. Initialisation du chien de garde	113
1.2. Les commandes d'armement et de désarmement	113
1.3. La gestion du chien de garde.	113
2 - <u>LE CALENDRIER SECOURU</u>	114
2.1. L'heure secourue	114
2.2. L'heure et la date maintenues	115
2.3. L'initialisation de la date et de l'heure	116
2.4. Gestion de la date et de l'heure en cas de coupure secteur par le logiciel SOLAR.	117
2.5. Le chargement de la batterie	117
3 - <u>LA SYNCHRONISATION INTER-CALCULATEUR</u>	119
3.1. Le principe de fonctionnement	121
3.2. La réalisation	122
4 - <u>LE BANDEAU DE VISUALISATION</u>	124
5 - <u>L'HORLOGE TEMPS REEL</u>	126
6 - <u>LES TEMPORISATIONS PROGRAMMABLES</u>	127
6.1. L'initialisation d'une ou plusieurs temporisations	128
6.2. Les commandes d'armement et de désarmement	130
7 - <u>LES HEURES DE REVEIL</u>	133
7.1. L'initialisation des heures de réveil	133
7.2. Le traitement des heures de réveil.	134

V - LES FONCTIONS TEMPS REEL

Dans toute application industrielle, les fonctions temps réel traitées par le CMF sont indispensables :

= pour des raisons de sécurité - voir le chien de garde calculateur, le calendrier secouru ou la synchronisation inter - calculateurs.

= pour alléger la gestion de ces fonctions au niveau des applications - voir l'horloge temps réel, les temporisations ou les heures de réveil.

Les fonctions temps réel peuvent être demandées simultanément par différentes tâches de l'application ; il a donc été nécessaire d'introduire la notion d'occupation du coupleur :

Une tâche demande l'exécution d'une fonction seulement si le coupleur n'est pas occupé.

Le logiciel temps réel est scindé en 4 parties dépendantes de l'organisation matérielle :

→ la phase d'initialisation dans laquelle il réalise toutes les initialisations matérielles et logicielles.

→ la tâche de traitement de la base de temps (10 ou 20 ms) dans laquelle sont gérées la plupart des fonctions (entre autres l'heure et la date).

Sur détection d'un défaut de cette base de temps, le micro-logiciel génère une interruption exception de défaut horloge pour alerter le SOLAR qu'aucune fonction temps réel ne peut être traitée.

→ la tâche de traitement des interruptions FIRQ (cf II.4.3.2)

→ la tâche de traitement des interruptions IRQ (cf. II.4.3.2)

1) LE CHIEN DE GARDE CALCULATEUR ET LES ENTREES TOUT OU RIEN

Le chien de garde a pour rôle de prévenir l'environnement du calculateur que l'équipement n'est plus en état de bon fonctionnement ou est hors tension. C'est un décompteur de temps programmable géré par le micro-processeur temps réel. Le logiciel peut armer, réarmer ou désarmer le chien de garde et programmer sa durée T. Si au bout du temps T, le logiciel n'a pas désarmé ou réarmé le chien de garde, le micro-logiciel commande un relais multi-contacts qui peut :

- activer un signal d'alarme (klaxon, voyant lumineux ...).
- avertir un autre calculateur par l'intermédiaire d'une des deux entrées tout ou rien disponibles sur le CMF et affectées à la surveillance - sur changement d'état d'une entrée tout ou rien, le coupleur génère une interruption exception au logiciel SOLAR.

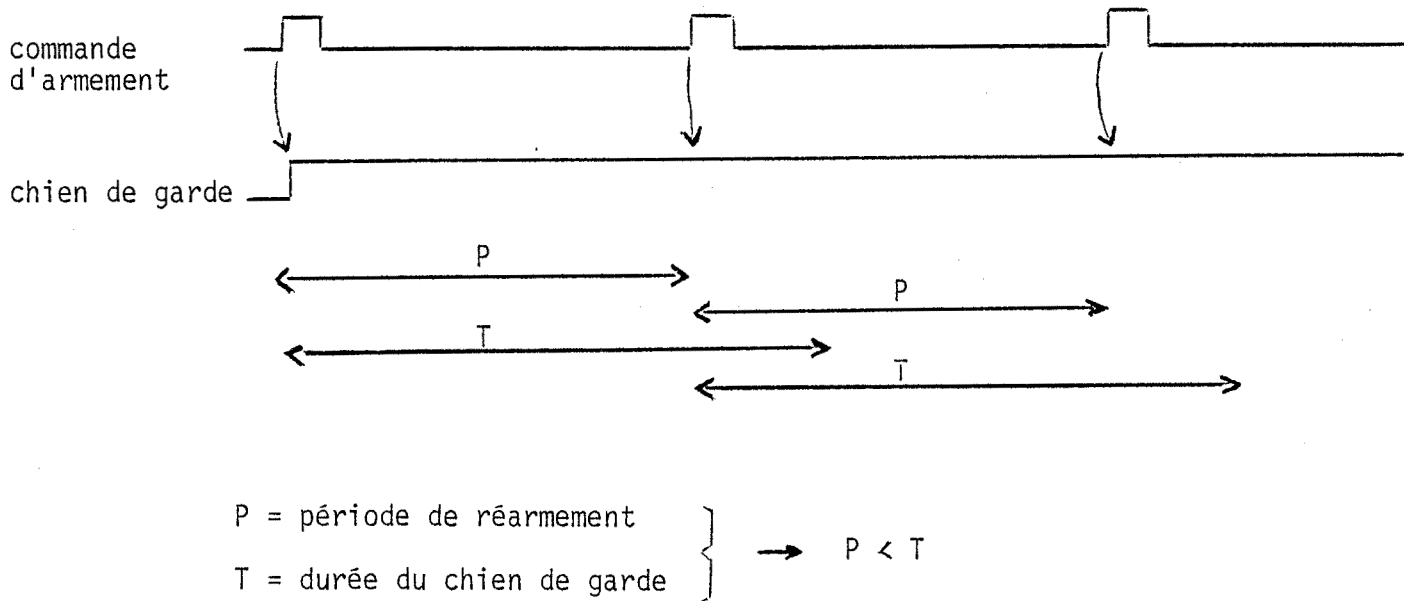


figure 5.1 : bon fonctionnement du calculateur

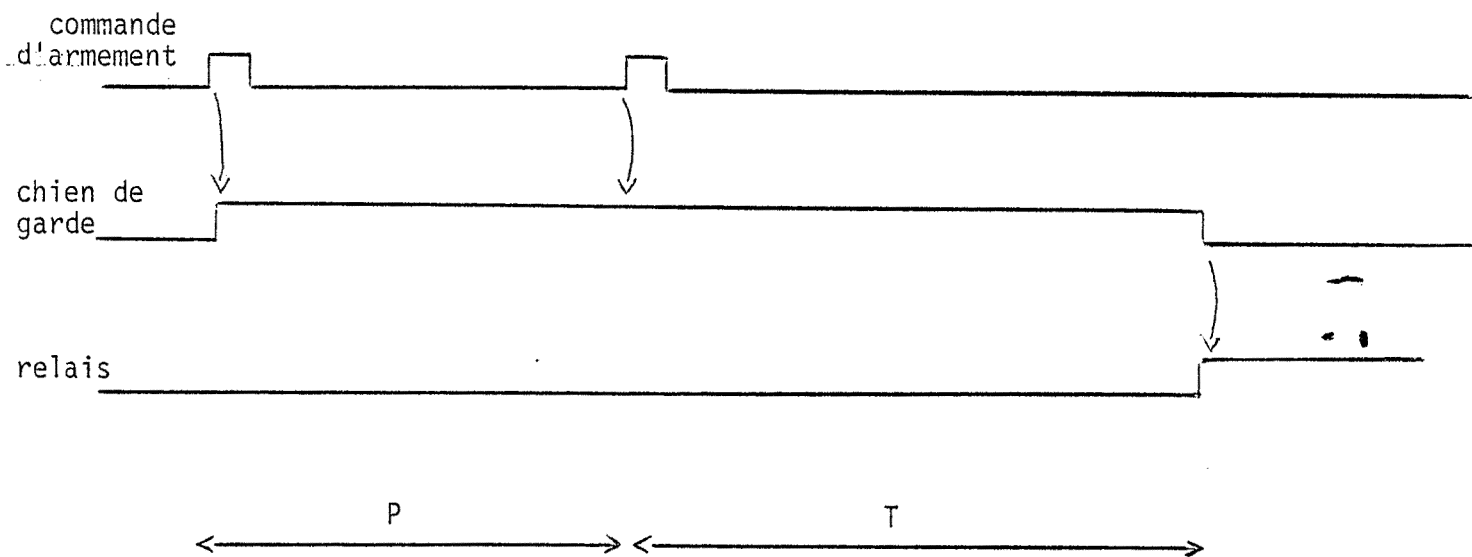


figure 5.2 : mauvais fonctionnement du calculateur

Le chien de garde ordinateur et les entrées tout ou rien permettent essentiellement de fabriquer des structures redondantes pour minimiser le temps d'indisponibilité d'un ordinateur en cas de panne.

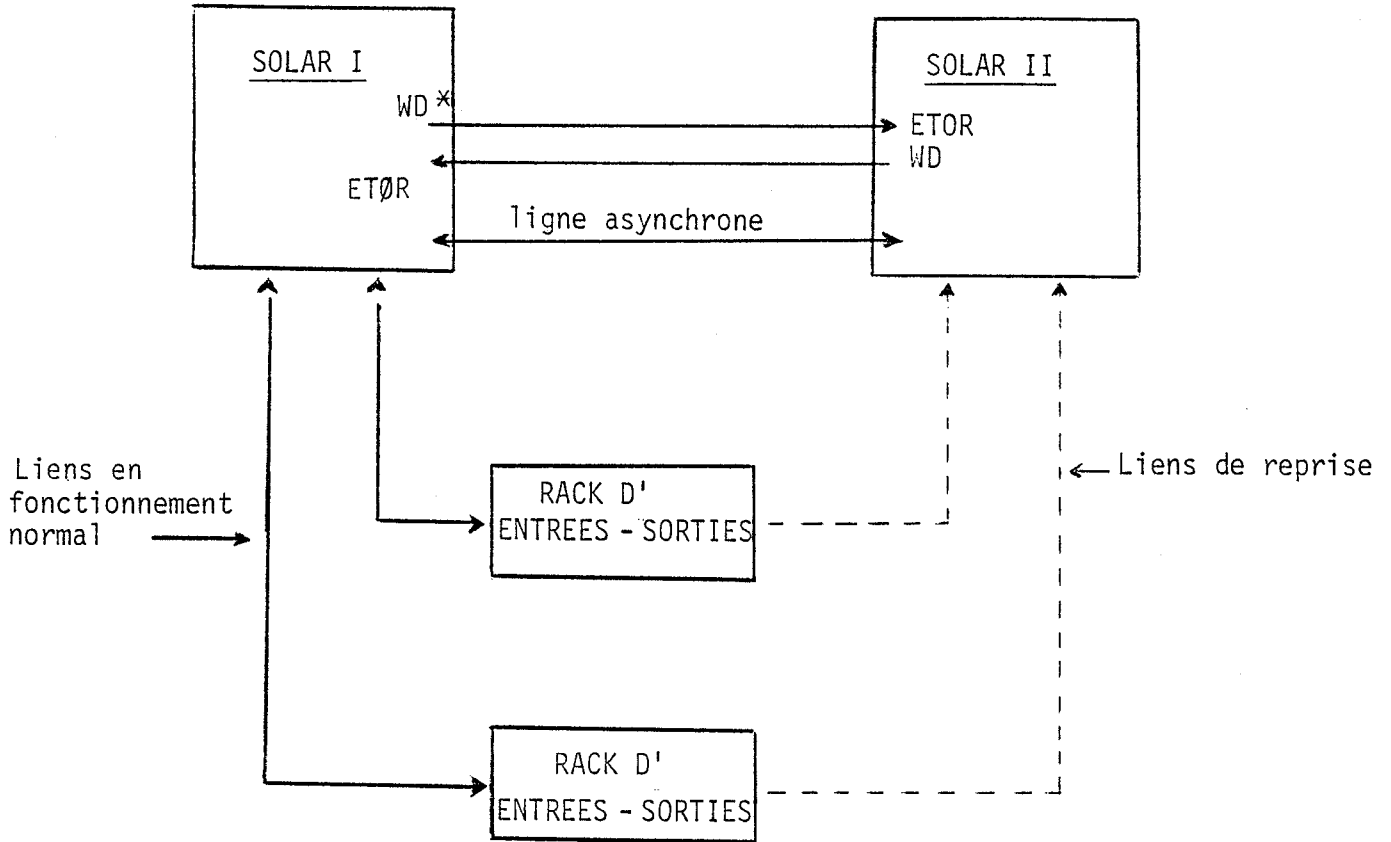


figure 5.3 : exemple de structure redondante de reprise en secours d'un processeur par un autre au repos

WD* : chien de garde (Watch - dog)

Un dialogue permanent de mise à jour s'exécute entre les deux unités centrales par l'intermédiaire d'une ligne asynchrone du CMF. En cas de panne du SOLAR I la reprise sera faite par l'unité centrale II, sans perte de temps.

Actuellement (sans CMF), ce type de structure nécessite dans une configuration SOLAR 3 cartes :

- une carte chien de garde,
- 1 coupleur asynchrone et
- 1 carte 16 appels externes

- 1.1. L'initialisation du chien de garde est réalisée par le logiciel SOLAR avec un échange canal de 2 octets dans lesquels il précise la durée (en "TOP" de base de 20 ms) et éventuellement la demande d'armement du chien de garde : cela évite de réaliser cette commande en plus de la séquence d'initialisation.

- 1.2. Les commandes d'armement et de désarmement sont exécutées par le logiciel par l'intermédiaire du registre de commande (cf. II.3.4).

Sur réception de la commande d'armement, le micro-logiciel recharge le décompteur de temps avec la valeur initiale.

Sur réception de la commande d'arrêt, il le réinitialise avec la valeur nulle.

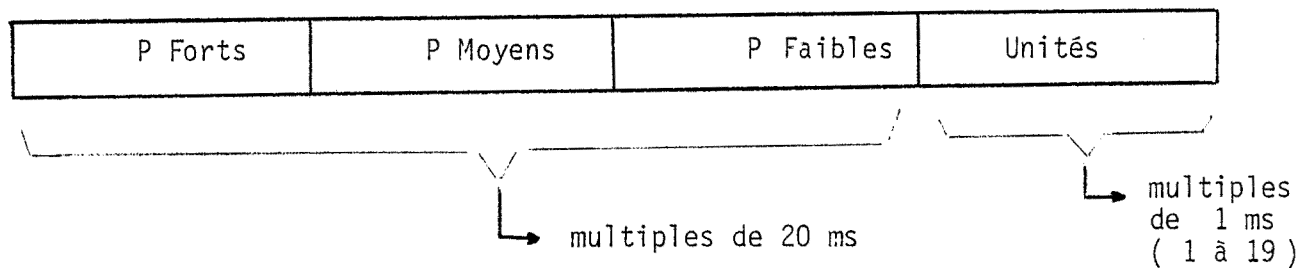
- 1.3. La gestion du chien de garde : le décompteur est géré, s'il n'est pas nul, dans la tâche de traitement du top 20 ms.

2) LE CALENDRIER SECOURU -

Actuellement, sur tout système SOLAR, après une coupure secteur, l'opérateur doit, au moment du redémarrage du système, recharger la date et l'heure, qui ne sont pas secourues. (Dans certaines configurations, le système est relié à une horloge externe par l'intermédiaire d'un coupleur universel, le coupleur GPI ; mais c'est une configuration peu utilisée car très coûteuse).

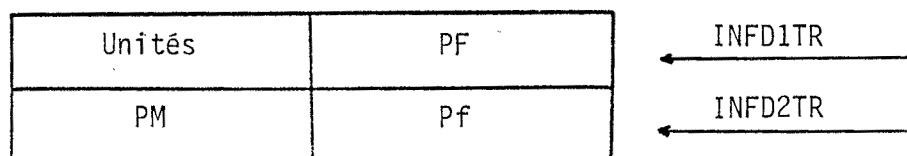
Le CMF élimine cette opération car il propose à l'utilisateur :

2.1. L'heure secourue exprimée en base de temps de 1 ms ; c'est une heure qui continue à évoluer tant que les batteries sont correctement chargées, soit environ pendant 72 heures. Elle est gérée par matériel avec 4 registres accessibles en lecture directe par le logiciel : cela permet à un utilisateur voulant lire l'heure fréquemment de ne pas être pénalisé par le temps de traitement d'échanges canaux.



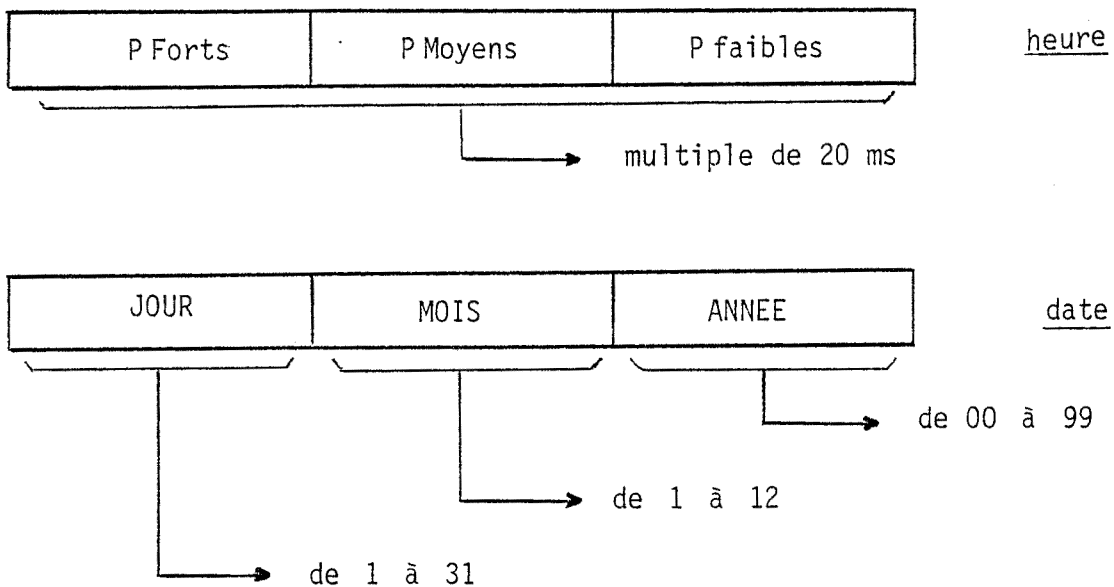
L'heure en multiple de 20 ms tient sur 3 octets ; le 4ème octet permet d'accéder aux multiples de 1 ms.

Le logiciel acquiert l'heure secourue par lecture directe des 4 registres INFD1RT et INFD2TR :



Cette séquence de lecture doit être ininterrompible pour éviter toute modification intermédiaire de ces registres.

2.2. L'heure, exprimée en base de temps de 20 ms et la date maintenues sont gérées par le micro-logiciel et mémorisées dans des circuits alimentés en 5 V secouru ; en cas de coupure secteur, elles sont sauvegardées mais ne continuent plus à évoluer : Elles donnent à l'utilisateur l'heure et la date de la coupure secteur.



L'utilisateur acquiert la date ou l'heure maintenues par un échange canal de 3 octets.

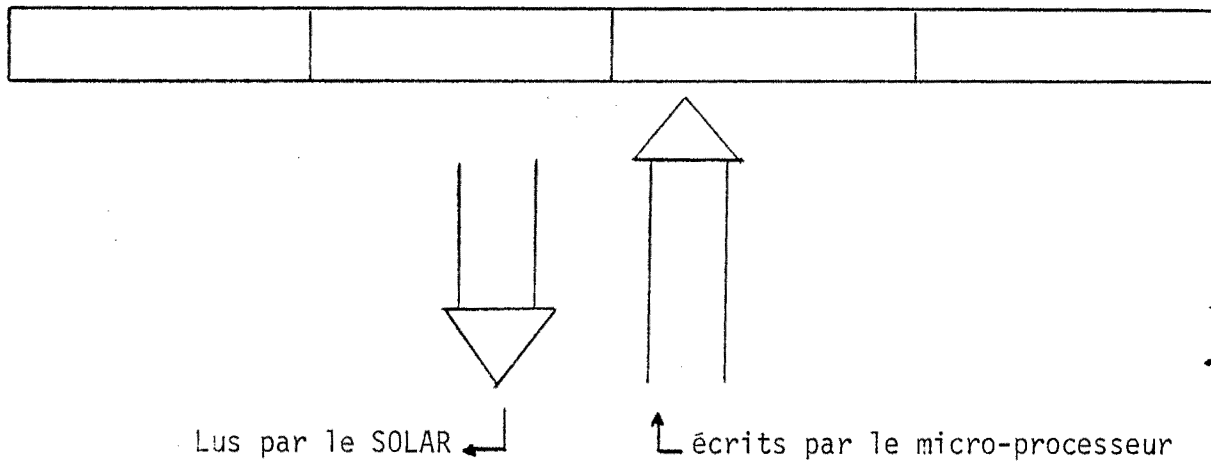
2.3. L'initialisation de la date et de l'heure est réalisée par le logiciel SOLAR avec un échange canal :

- de 3 octets pour la date dans lesquels il précise le JOUR, le MOIS et l'ANNEE.

- de 4 octets pour l'heure par lesquels il donne au micro-logiciel une indication lui précisant si c'est une initialisation de l'heure après coupure secteur ou pas (pour calculer le temps de charge de la batterie) et l'heure en multiple de 20 ms.

Sur réception de la commande d'initialisation de l'heure, le micro-logiciel initialise :

- l'heure maintenue
- et l'heure secourue avec remise à 0 des multiples de 1 ms ; le micro-logiciel n'a accès à l'heure secourue qu'en écriture ; cela, lié à la notion d'occupation du coupleur, élimine les conflits d'accès à ces registres entre SOLAR et micro-processeur.



L'heure et la date sont ensuite gérées dans la tâche de traitement du top 20 ms.

2.4. Gestion de la date et de l'heure par le logiciel SOLAR, en cas de coupure secteur

A la remise sous tension du système, si le 5 V secouru est encore correct (une interruption exception est générée par le micro-processeur sur défaut batterie), l'application doit :

- = gérer la reprise en fonction de la durée de la coupure secteur.
- = remettre à jour la date maintenue : il connaît la durée de la coupure secteur par lecture et comparaison de l'heure secourue et de l'heure maintenue.
- = remettre à jour l'heure maintenue en précisant au micro-logiciel qu'il s'agit d'une initialisation après coupure secteur pour que le micro-logiciel puisse assurer la charge de la batterie.

2.5. Le chargement de la batterie

La batterie incorporée au coupleur alimente la mémoire vive du micro-processeur et les circuits gérant l'heure secourue pendant 72 heures.

A la remise sous tension du système, cette batterie a besoin d'être chargée. Cela peut être réalisé :

- par le matériel, en appliquant un courant pendant un certain temps ; cette solution est coûteuse en nombre de composants.
- par le micro-logiciel, en fonction du temps de la coupure secteur ; cette solution est celle retenue car moins coûteuse du point de vue matériel. Le micro-processeur a, à sa disposition, tous les éléments nécessaires à la réalisation de cette fonction.

Lors de l'initialisation de l'heure maintenue, le SOLAR rend au micro-logiciel une indication lui précisant si c'est une initialisation après coupure secteur ou pas. Le micro-logiciel connaît alors le temps de la coupure secteur par comparaison de l'ancienne valeur de l'heure maintenue à la valeur courante donnée par le logiciel SOLAR.

Le temps de charge de la batterie dépend du temps de la décharge (durée de la coupure secteur) selon la formule (liée à la batterie choisie)

$$t_{\text{charge}} = 0,12 \times t_{\text{coupure}} - 0,5 \text{ heure}$$

D'après cette formule :

$$\text{si } t_{\text{coupure}} \leq 4 \text{ H, alors } t_{\text{charge}} = 0$$

(temps de charge et temps de décharge sont exprimés en heure).

La batterie sera en légère " sous-charge " (préférable pour la batterie à une surcharge) et sera rechargée lentement par un courant d'entretien.

Cette formule permet un taux d'utilisation moyen de la batterie de 1/10.

EXEMPLE : si la batterie s'est déchargée pendant 10 heures, il suffit de la recharger pendant 1 heure.

Pour charger la batterie, le micro-logiciel commande un relais périodiquement. La commande de ce relais enclenche un monostable qui provoque la fermeture et l'ouverture du relais de charge.

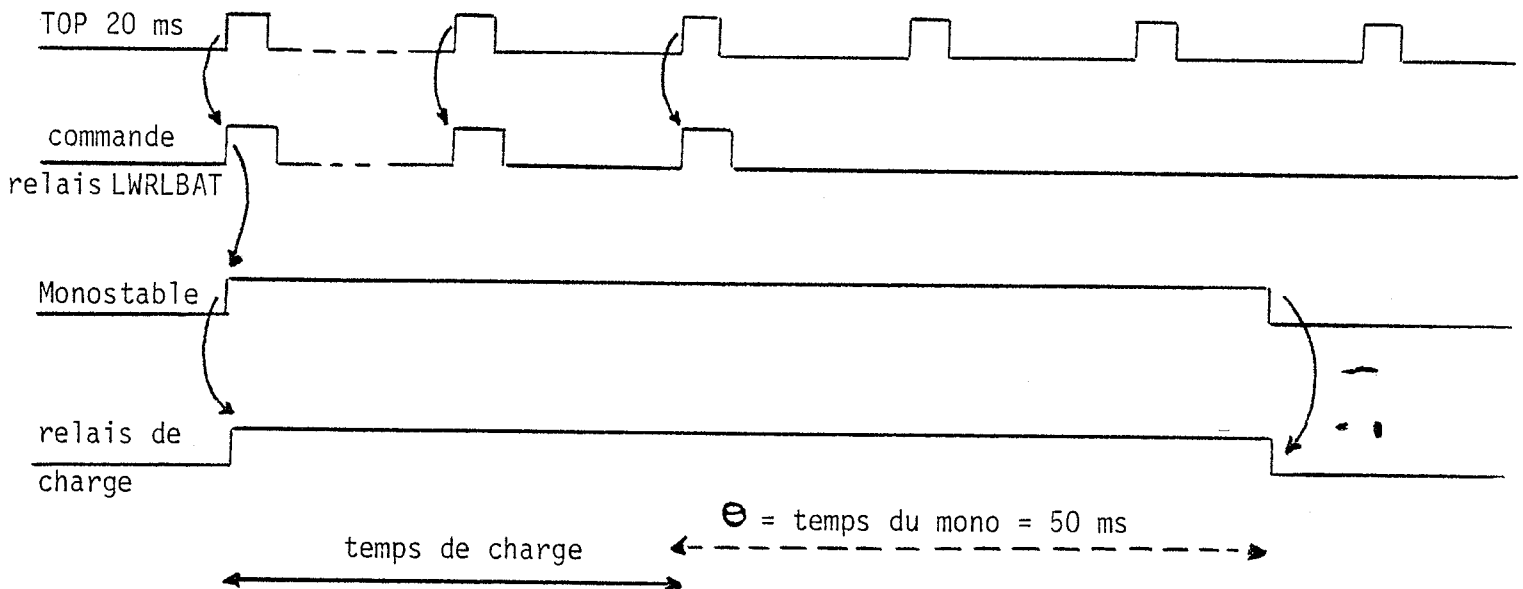


figure 5-4 : chargement de la batterie

Cette solution a été choisie pour des raisons de sécurité : si un défaut bloque le fonctionnement du micro-processeur, il faut que ce soit le matériel qui arrête la charge de la batterie pour en éviter la détérioration par surcharge.

3) LE SIGNAL DE SYNCHRONISATION

Dans un environnement décentralisé, le signal de synchronisation inter-calculateurs assure une base de temps identique sur tous les calculateurs.

Cette référence de temps précise est indispensable dans toute application industrielle.

EXEMPLE : dans le cas d'une structure hiérarchisée, un calculateur central est relié à un ou plusieurs frontaux et a un rôle hiérarchique vis à vis des frontaux : c'est lui qui se charge de regrouper les résultats, d'élaborer des bilans globaux, de classer les événements, de réaliser les historiques des informations ...

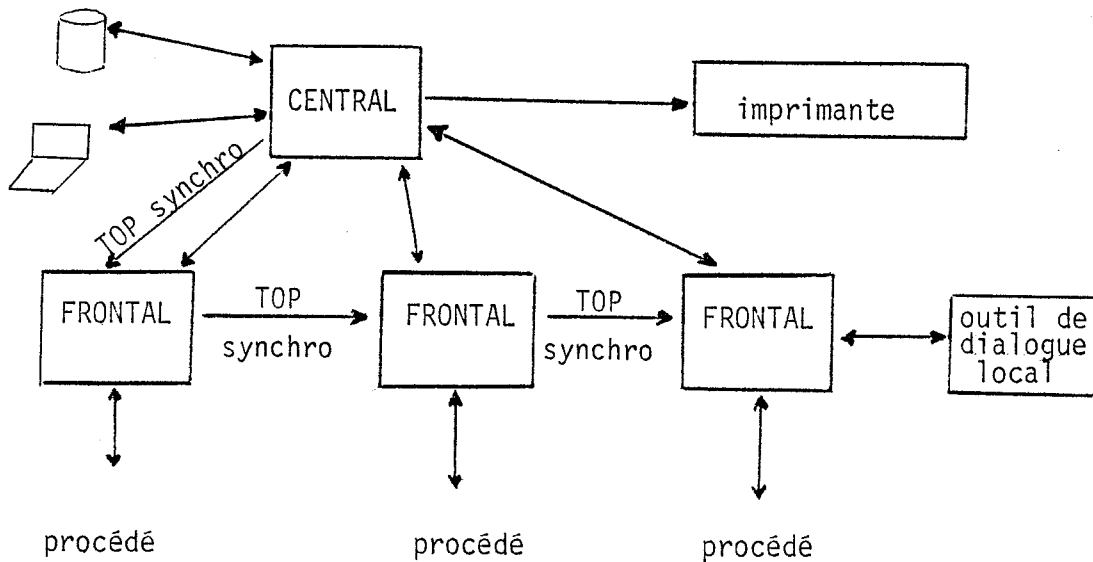


figure 5-5 : top synchro dans une structure hiérarchisée

Le central et les frontaux doivent posséder la même heure ; le central peut ainsi :

- réaliser le classement chronologique des événements reçus ;
- identifier et remonter aux sources d'évènements responsables d'une avalanche de défauts,

- faire une comparaison dans l'absolu entre différents éléments,
- demander l'exécution d'actions individuelles mais simultanées, ayant éventuellement des liens entre elles, par tous les frontaux, à un instant t
-

Actuellement dans les systèmes SOLAR, ce problème d'heure commune est difficilement résolu. Le coupleur multi-fonctions apporte une solution qui offre des perspectives intéressantes pour l'avenir.

3.1. Le principe de fonctionnement

Le calculateur maître génère périodiquement le signal de synchronisation aux calculateurs esclaves qui doivent, dès réception du signal, après un temps t de filtrage du signal, recadrer leur base de temps (c'est-à-dire l'heure maintenue et l'heure secourue).

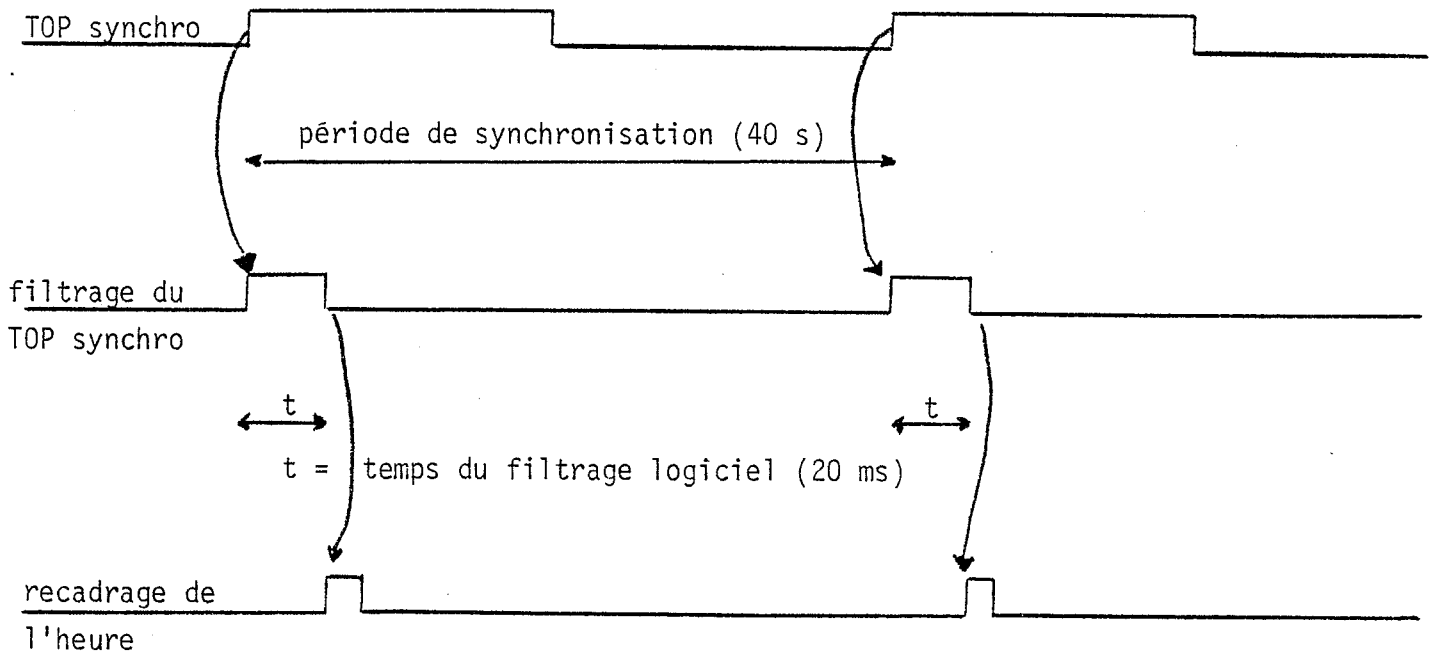


figure 5-6 : Principe de fonctionnement du top synchro.

3.2. La réalisation

Le calculateur maître génère périodiquement le signal de synchronisation aux calculateurs esclaves.

La période de synchronisation dépend de la dérive ($\Theta = 5$ ms) tolérée (d'un calculateur à un autre, l'heure ne doit pas dériver de $+ \text{ou} - \frac{\Theta}{2}$) et de la précision du quartz utilisé.

$$p = \frac{\text{dérive } \Theta}{\text{précision}}$$

En cas de coupure secteur, la synchronisation de l'heure (dérive $< \Theta$) sera assurée pendant un temps Δ dépendant de la période de recadrage et de la précision du quartz.

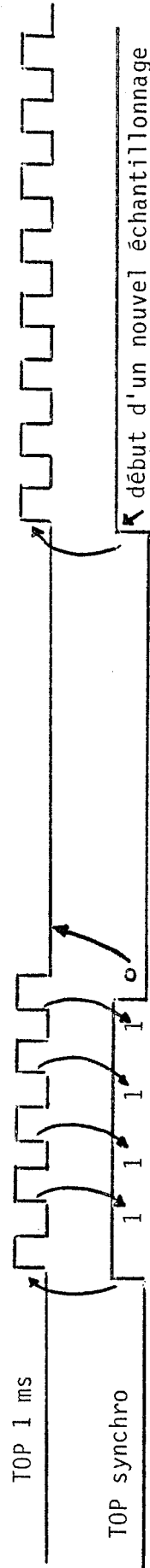
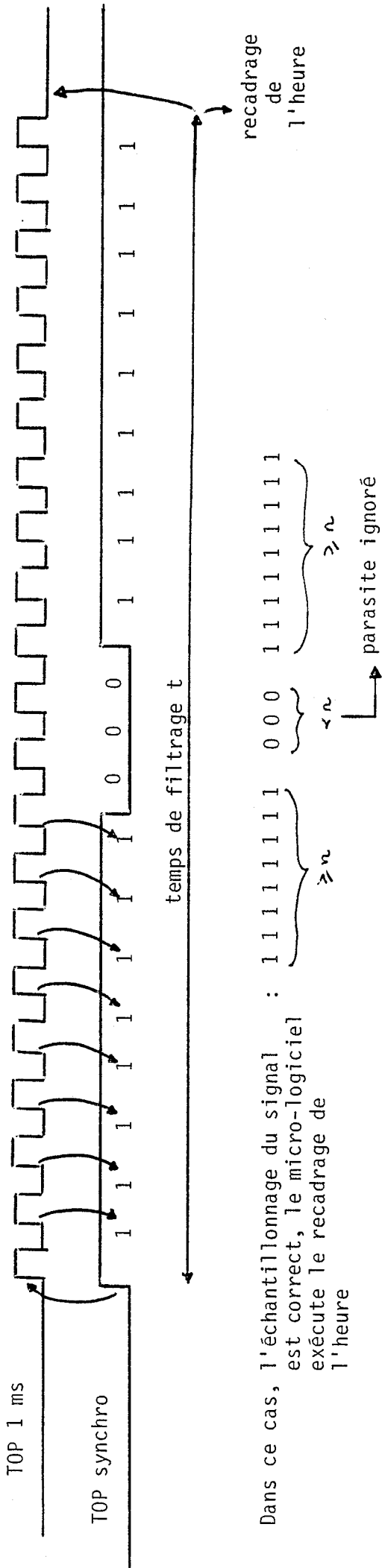
$$\Delta = \frac{\text{Période}}{\text{précision} \times 3600 \text{ S}}$$

Un filtrage minimum du signal est assuré par le matériel et est complété par un filtrage logiciel.

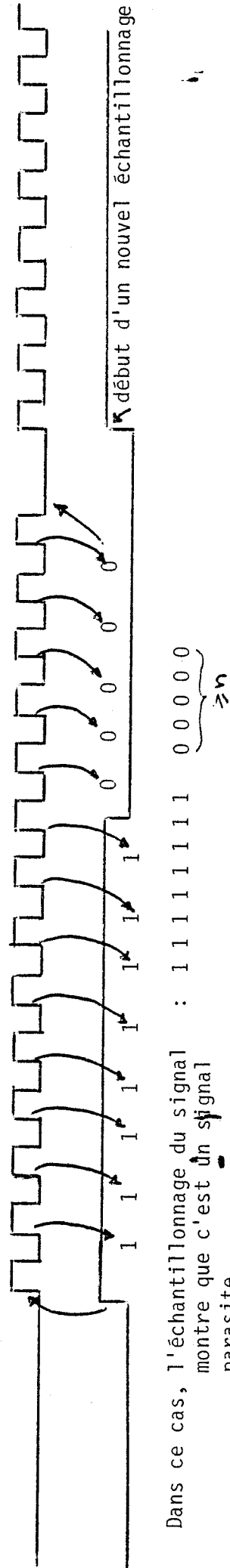
Principe du filtrage logiciel : Le micro-processeur reçoit périodiquement une interruption prioritaire sur le passage à 1 du signal "TOP SYNCHRO". Dès réception de cette interruption, le micro-logiciel échantillonne le signal top synchro toutes les ms pendant un temps de filtrage t.

- à l'apparition du signal, le signal doit être présent au moins n ms et pendant le reste du temps de filtrage t, il ne doit jamais être au repos plus de n ms, pour ne pas être déclaré parasite.

Le recadrage de l'heure n'est réalisé qu'à la fin de la séquence de filtrage et est fonction de la période P du signal de synchronisation et du temps t de filtrage.



Dans ce cas, l'échantillonnage du signal montre que c'est un signal parasite



Dans ce cas, l'échantillonnage du signal montre que c'est un signal parasite

figure 5-7 : EXEMPLES DE FILTRAGE

4) LE BANDEAU DE VISUALISATION

Il permet de visualiser :

- = la charge de traitement de l'unité centrale
- = l'état du chien de garde
- = l'heure secourue

Le bandeau de visualisation est un panneau métallique enclenchable sur la face avant du SOLAR et devant respecter des dimensions précises.

De ce fait, pour limiter la taille du module, nous avons dû diminuer le nombre de composants présents sur sa face arrière et nous avons choisi la solution à base de micro-processeur 6809.

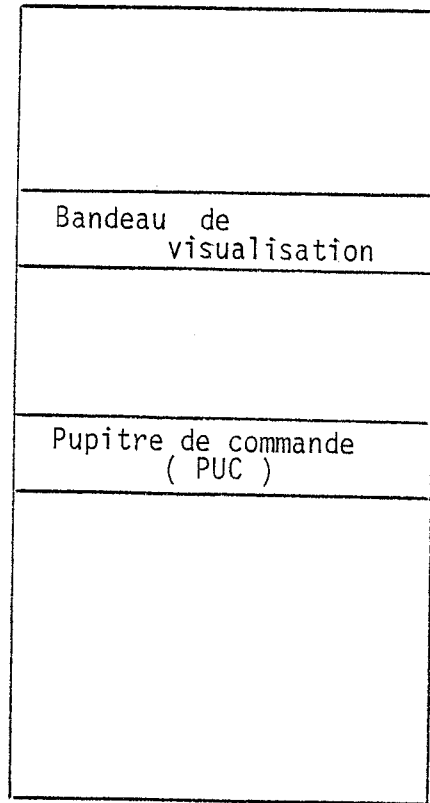


figure 5-8 : face avant du SOLAR

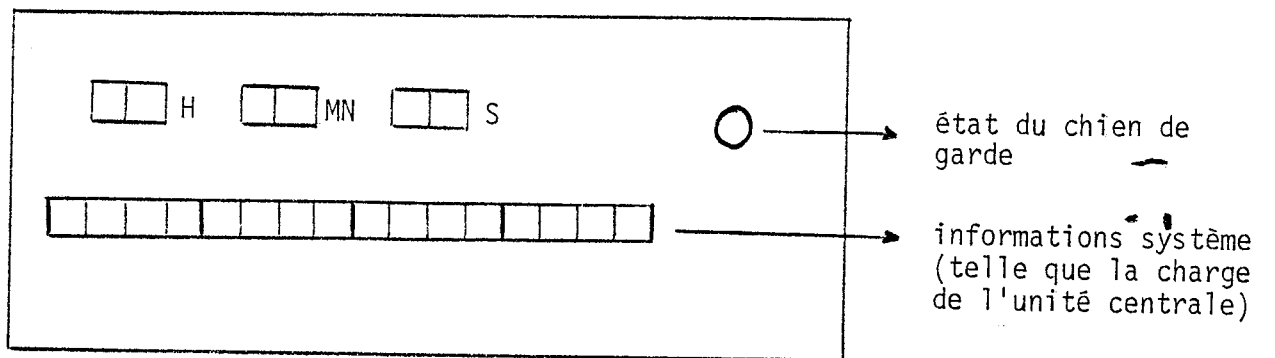


figure 5-9 : le bandeau de visualisation

5) L'HORLOGE TEMPS REEL est une horloge fournie périodiquement au logiciel d'application qui l'utilise actuellement pour gérer la date et l'heure ou des échéances particulières telle que l'échéance 24 Heures.

L'application peut valider ou dévalider cette HTR, par l'intermédiaire d'un registre de commande (cf. II-3-3). Si l'horloge temps réel est validée, le CMF génère, périodiquement (toutes les 20 ms), une interruption exception au logiciel SOLAR qui doit reconfirmer l'ordre de validation (sinon l'HTR est arrêtée).

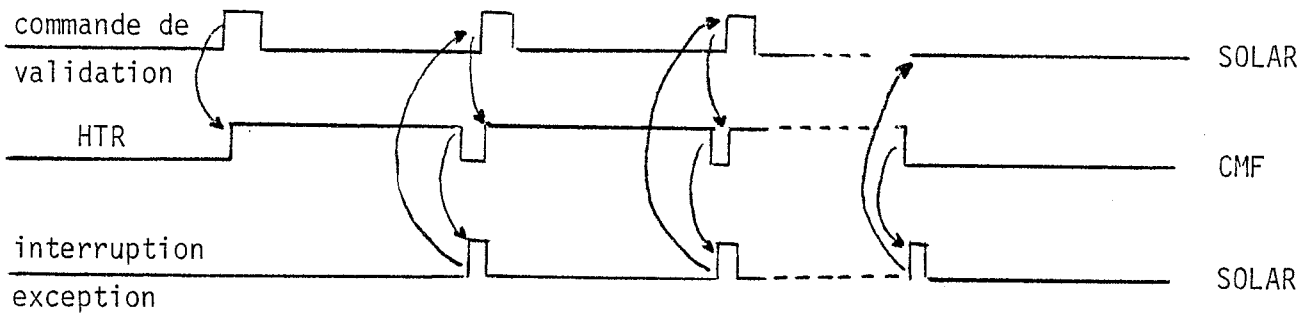


figure 5.10 : principe de fonctionnement de l'horloge temps réel

L'horloge temps réel est gérée par le matériel pour être compatible avec les logiciels existants.

6) LES TEMPORISATIONS PROGRAMMABLES sont indispensables dans toute application industrielle pour gérer des actions périodiques.

Une action périodique est une action réalisée à une période déterminée et indépendante de l'heure. L'application associe à chaque action une tâche qui peut réaliser une ou plusieurs actions.

A chaque action est associée une temporisation du CMF. Une plage de 32 temporisations est, à priori, suffisante pour les applications industrielles : c'est le cas pour MASC16, le progiciel de conduite de procédés et d'automatismes industriels développé par SEMS.

Le CMF propose 128 temporisations programmables :

- en unité : c'est une période de base qui évite au micro-logiciel de traiter toutes les temporisations à chaque "top" 10 ms.
- en valeur : nombre d'unités
- en répétitivité : nombre de réarmements automatiques après échéance. Ce nombre peut être déclaré "infini" pour réaliser des activations périodiques.

Sur l'arrivée à échéance d'une temporisation (valeur nulle), le CMF génère une interruption exception qui va activer la tâche associée et qui décharge ainsi le logiciel d'application de la gestion assez coûteuse de ces temporisations.

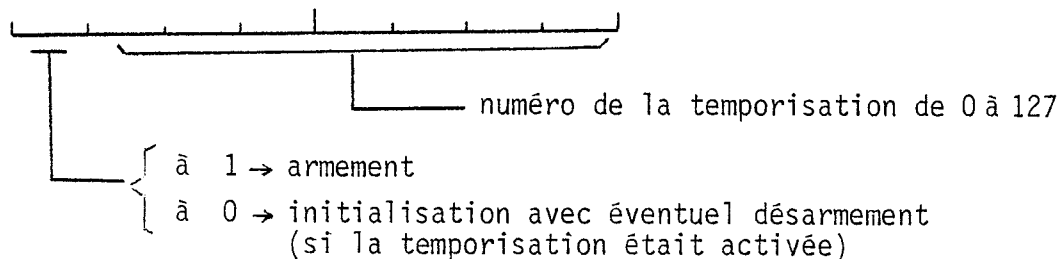
Le logiciel peut :

- initialiser et armer une ou plusieurs temporisations simultanément
- armer ou désarmer une temporisation.

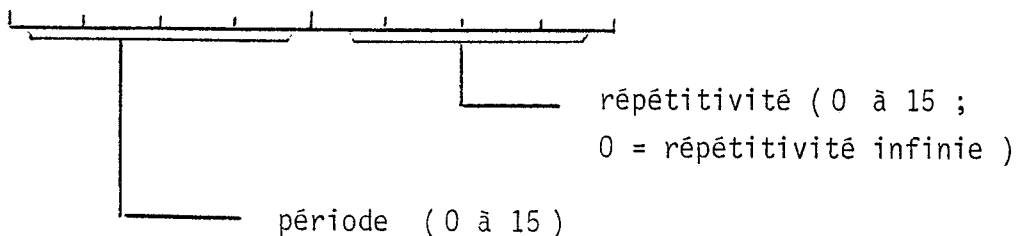
6.1. L'initialisation d'une ou plusieurs temporisations

est réalisée par le logiciel SOLAR avec un échange canal de 4 octets dans lesquels il précise :

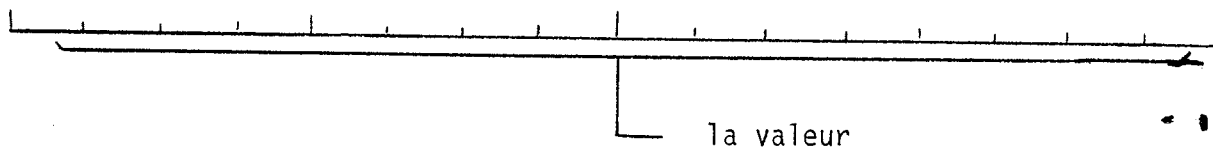
- le numéro de la temporisation et éventuellement la commande d'armement



- la période et la répétitivité



- la valeur



Le micro-logiciel mémorise les paramètres donnés par le logiciel SOLAR dans la table ITBTIM ;

Pour chaque temporisation, il sauvegarde :

- la période de traitement
- la durée
- la répétitivité.

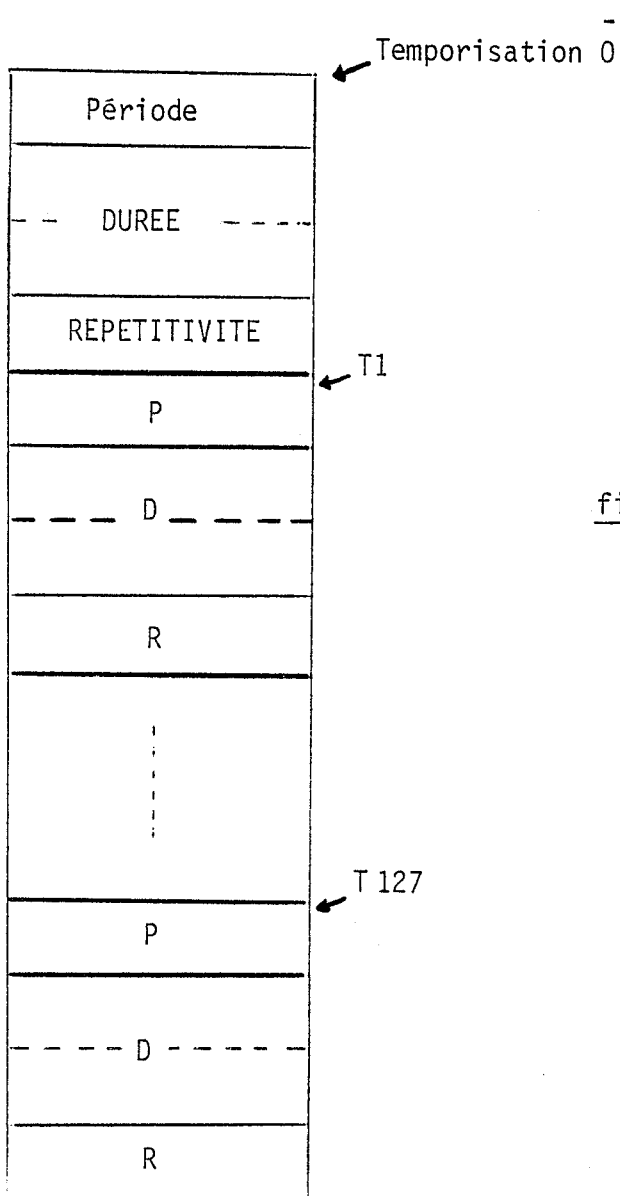


figure 5-11 : description de ITBTIM

La période rangée dans ITBTIM est fonction de la période codée par le logiciel SOLAR

Période codée	Période rangée	Période de traitement (en ms)
0	1	10
1	2	20
2	5	50
3	10	100
4	15	150
5	20	200
6	25	250
7	40	400
8	50	500
9	60	600
10	70	700
11	80	800
12	100	1000
13	150	1500
14	200	2000
15	250	2500

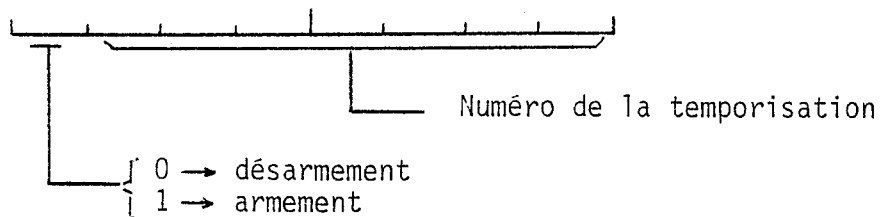
figure 5-12 : correspondance entre période SOLAR et période de traitement

6.2. Les commandes d'armement et de désarmement d'une temporisation

sont réalisées :

= soit à l'initialisation de la temporisation (cf. V-6-1)

= soit par la commande "timer" ; dans ce cas, le logiciel envoie au CMF la commande "TIMER" et une donnée dans laquelle il précise le numéro de la temporisation et le type de commande



6.2.a) Sur réception de la commande d'armement, le micro-logiciel initialise deux tables de travail, TBTIM et TBTRT, en fonction des valeurs initiales stockées dans ITBTIM. Ces tables sont ensuite gérées par la tâche de traitement du top 10 ms.

Dans TBTIM sont mémorisés la période de traitement et le numéro de la temporisation.

- si la période est nulle : la temporisation n'est plus activée.
- la période 'FF' joue le rôle d'indicateur de fin de table.

Dans TBTRT sont mémorisées la valeur et la répétitivité de la temporisation

6.2.b) Sur réception de la commande de désarmement, le micro-logiciel force à 0 la période de la temporisation stockée dans la table TBTIM ; cette période est utilisée comme période de traitement mais aussi comme indicateur d'armement de la temporisation.

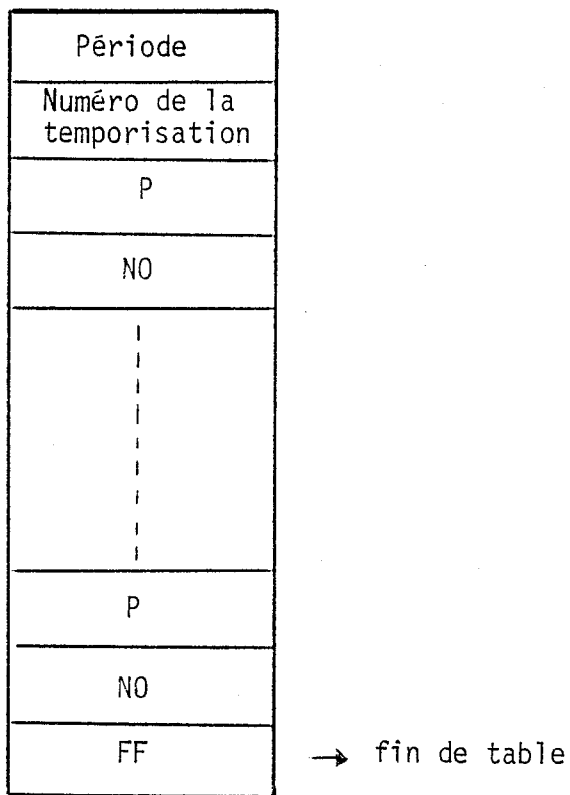


figure 5-13 : description de TBTIM

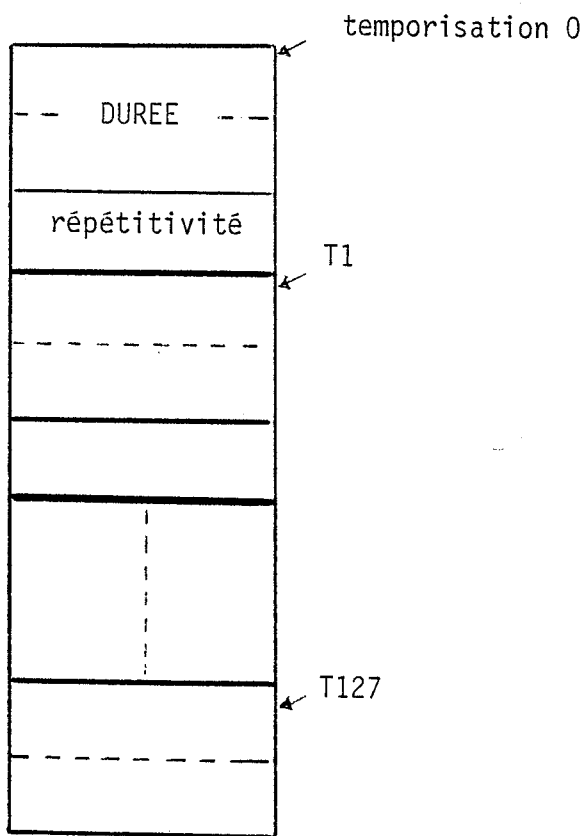


figure 5-14 : description de TBTRT

Tâche de Traitement du TOP 10 ms

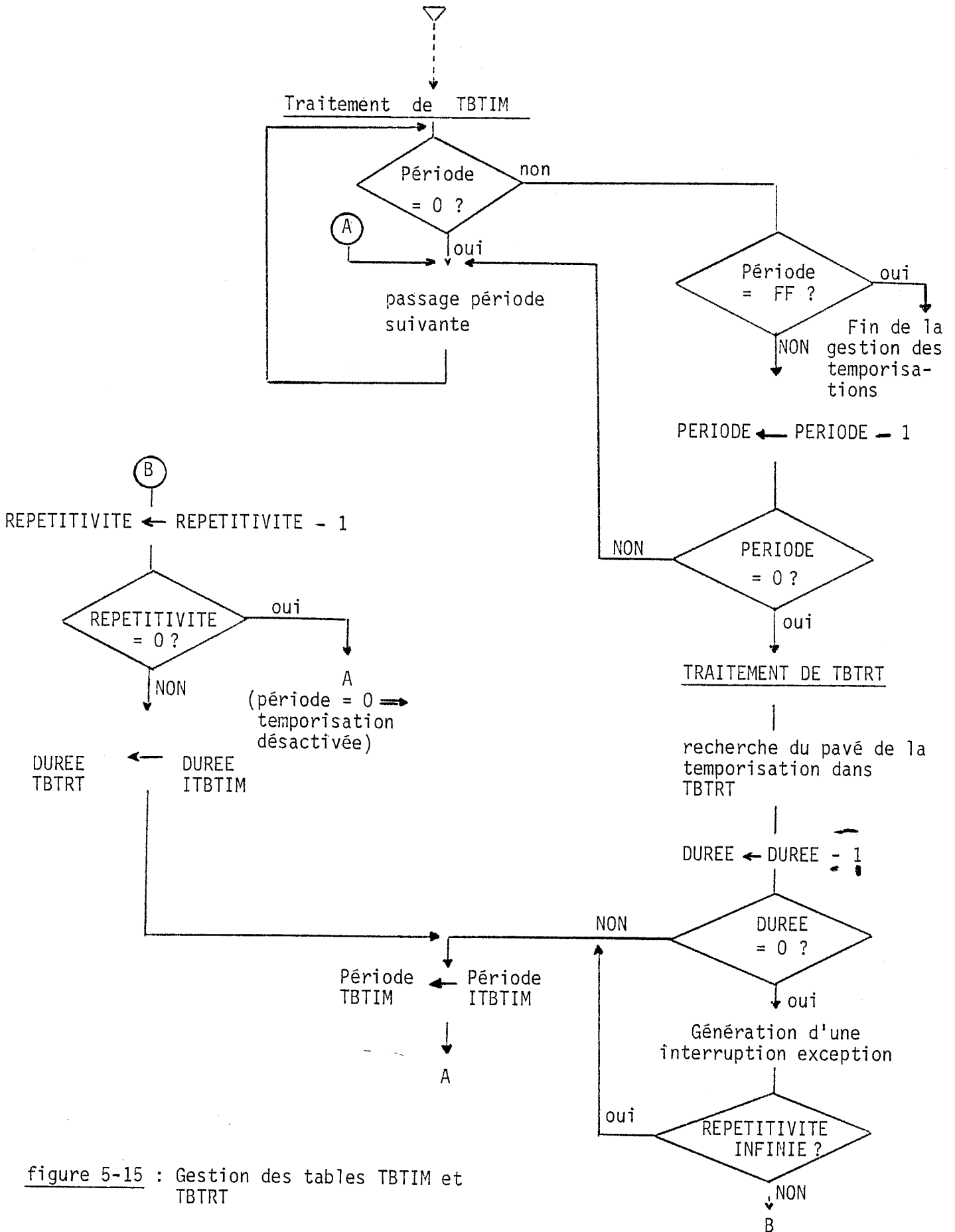


figure 5-15 : Gestion des tables TBTIM et TBTRT

7) LES HEURES DE REVEIL sont des échéances de temps dépendantes de l'heure du CMF, et permettent de gérer des actions non périodiques, c'est-à-dire des actions à exécuter à une heure donnée ou après un délai spécifié.

Ces actions sont, par exemple :

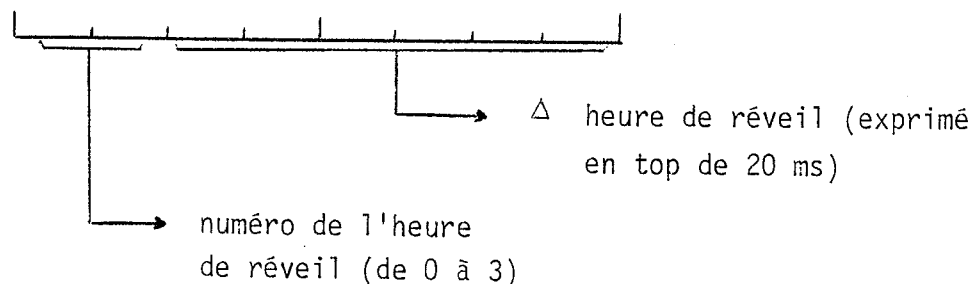
- = l'édition à heure fixe d'un journal de bord, de feuilles de quart ...
- = l'édition du bilan de la journée
- = ...

L'application gère une chaîne horloge dans laquelle, à chaque action, sont associés un ou deux pavés qui précisent le type d'action à réaliser et la priorité de la tâche à activer ainsi que des paramètres propres au type d'action. Les pavés sont chaînés par ordre d'échéance croissante. Actuellement, c'est l'application qui gère les heures d'échéance, par comparaison à l'heure absolue. Avec le CMF, l'application est déchargée de cette gestion car, sur l'arrivée à échéance d'une heure de réveil, elle est réveillée par l'interruption exception générée par le coupleur.

7.1. L'initialisation des heures de réveil

Le CMF propose 8 heures de réveil initialisées par le logiciel SOLAR :

- soit avec un échange canal de 4 octets dans lesquels il précise :
 - le numéro de réveil (de 0 à 7)
 - l'heure de réveil exprimée en top de 20 ms (comme l'heure maintenue)
- soit avec la commande d'initialisation : " DELTA HEURE DE REVEIL " ; dans ce cas, le logiciel envoie au CMF la commande et une donnée qui précise la différence entre l'heure d'échéance et l'heure courante (inférieure à 1,2 seconde) et le numéro de réveil (de 0 à 3)



Les échéances 4 à 7 ne peuvent pas être initialisées par cette commande ; ce type de communication de l'heure d'échéance est offert à l'application car il évite la traversée du moniteur d'entrées - sorties (IOCS), pénalisante, en temps de traitement, dans le cas d'échéances très courtes.

7.2. Traitement des heures de réveil

7.2.a) A l'initialisation d'une échéance, le micro-logiciel mémorise cette échéance (éventuellement incrémentée de 24 Heures si elle est inférieure à l'heure courante) et positionne un indicateur précisant que l'heure de réveil X (X:de 0 à 7) est à tester.

7.2.b) Les heures de réveil sont ensuite gérées par le CMF :

= dans la tâche de traitement de l'heure maintenue où s'effectue la comparaison de l'échéance à l'heure courante (avec génération éventuelle d'une interruption exception pour réveiller le logiciel SOLAR).

= dans la tâche de traitement de la date (ou pavé minuit) où les heures d'échéances initialisées sont décrémentees de 24 Heures (pour recadrer le cas des heures de réveil < heure courante).

CHAPITRE VI

UTILISATION DU CMF DANS LES APPLICATIONS INDUSTRIELLES

	<u>PAGES</u>
VI - <u>UTILISATION DU CMF DANS LES APPLICATIONS INDUSTRIELLES</u>	136
1 - <u>LES CONFIGURATIONS AUTONOMES</u>	137
1.1. Sans CMF	137
1.2. Avec CMF	137
2 - <u>LES CONFIGURATIONS HIERARCHISEES</u>	140
2.1. Sans CMF	140
2.2. Avec CMF	141

VI - UTILISATION DU CMF DANS LES APPLICATIONS INDUSTRIELLES

Vu l'intérêt des fonctions assurées par le coupleur multi-fonctions, il équipera toute nouvelle configuration SOLAR.

Nous développerons, dans ce chapitre, quelques exemples de configuration avec ou sans CMF.

1) LES CONFIGURATIONS AUTONOMES

1.1. Sans coupleur CMF, une configuration autonome nécessite :

- = l'unité de traitement et de gestion des entrées - sorties
- = la mémoire associée
- = éventuellement, l'option flottant câblé (carte format 1/1) ou l'option flottant micro-programmé (extension du processeur)
- = un outil de dialogue permettant d'utiliser le système et connecté au coupleur MFI.
- = une mémoire de masse et un outil de chargement de système (du type disquette WINCHESTER ou disque dur) et son coupleur associé (carte format 1/2 si disquette, format 1/1 sinon)
- = une imprimante pour éditer les résultats du procédé et connectée à un coupleur ASX.
- = le pupitre POP qui permet de réaliser les fonctions de mise au point et de maintenance du système.

1.2. Avec le coupleur CMF

SEMS développe, parallèlement au coupleur CMF, un processeur de traitement avec mémoire intégrée : le processeur 16/30 ; nous trouverons, ainsi, dans une configuration autonome :

- = le processeur 16/30
- = éventuellement l'option flottant câblé ou l'option flottant micro-programmé
- = un outil de dialogue permettant d'utiliser le système et connecté sur l'une des lignes asynchrones du CMF.
- = une mémoire de masse et un outil de chargement du système et son coupleur associé.
- = une imprimante connectée à l'autre ligne asynchrone du CMF.

Dans ce type de configuration, le CMF offre à l'utilisateur des fonctions inutilisées telles que le téléchargement ou la synchronisation inter-calculateurs.

Par contre, il permet à l'utilisateur :

= de décharger son unité centrale en déportant le traitement des codes d'arrêt multiples ou des codes de saturation, désaturation.

= d'exploiter toutes les fonctions temps réel (le calendrier secouru ; les temporisations et les heures de réveil ; le chien de garde pour alerter l'opérateur (ou autre) de son mauvais fonctionnement ; les 2 entrées tout ou rien pour piéger éventuellement des évènements du procédé).

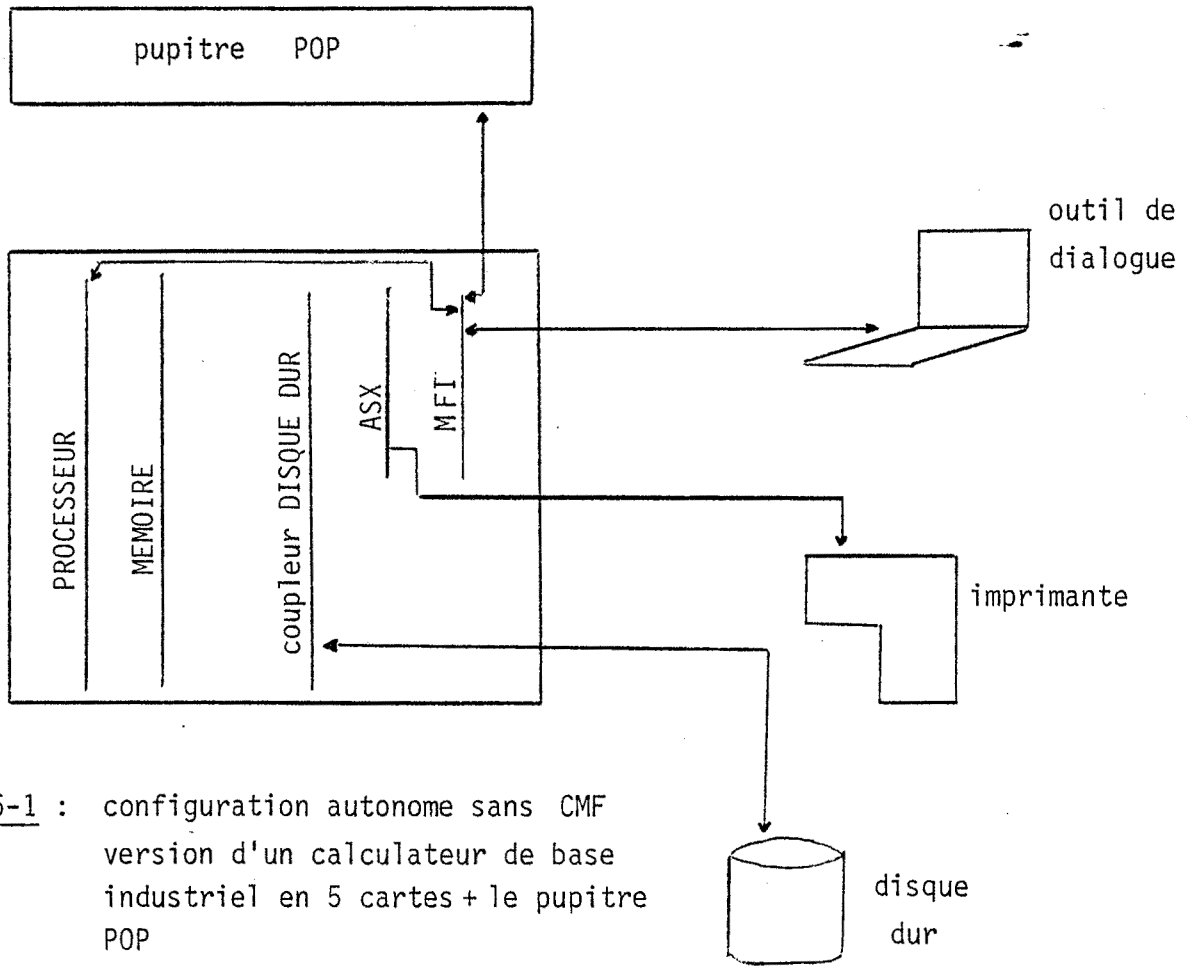


figure 6-1 : configuration autonome sans CMF
version d'un ordinateur de base
industriel en 5 cartes + le pupitre
POP

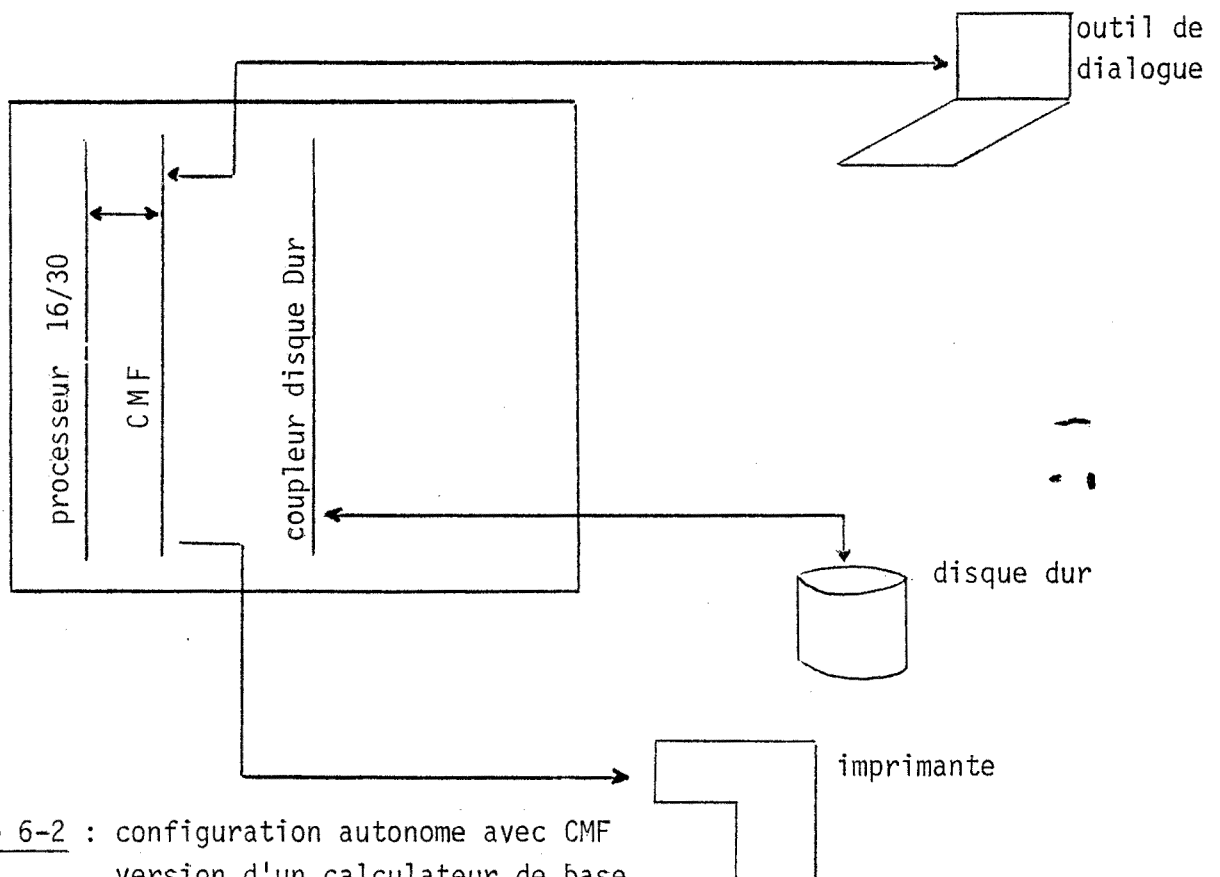


figure 6-2 : configuration autonome avec CMF
version d'un ordinateur de base
industriel en 3 cartes ; les fonctions pupitre sont activables
par l'outil de dialogue.

2) LES CONFIGURATIONS HIERARCHISEES

Elles sont organisées autour d'un calculateur central et de n calculateurs frontaux. Les frontaux sont souvent des calculateurs "aveugles" (sans pupitre POP de maintenance et de mise au point) et vides de tout logiciel ; ils sont téléchargés par le calculateur central.

2.1. Sans coupleur CMF

2.1.1. Un frontal doit être équipé :

- = d'un processeur de traitement
- = de la mémoire associée
- = d'une ligne de transmission asynchrone pour dialoguer avec le calculateur central.
- = d'un coupleur TIP pour assurer le téléchargement du frontal, par l'intermédiaire de la ligne asynchrone.
- = éventuellement d'un outil de dialogue local connecté sur une ligne asynchrone (coupleur ASX)
- = d'un chien de garde calculateur et d'entrées tout ou rien, dans le cas de structure doublée.
- = d'une autre ligne asynchrone pour le dialogue de mise à jour avec le calculateur de secours, dans le cas d'une structure doublée.

2.1.2. Le central doit être équipé :

- = d'un processeur de traitement
- = de mémoire
- = de plusieurs lignes de transmission asynchrones pour dialoguer avec les frontaux (coupleur MUX 4)
- = d'une imprimante connectée à un coupleur asynchrone (ASX)
- = d'un outil de dialogue graphique connecté au coupleur MFI
- = d'une mémoire de masse et de son coupleur associé
- = du pupitre POP de mise au point et de maintenance

Dans ce type de configuration, les fonctions de calendrier secouru et de synchronisation inter - calculateurs sont assurées par le logiciel d'application ainsi que la gestion des temporisations et des heures de réveil.

2.2. Avec le coupleur CMF

Nous associons, dans les calculateurs frontaux :

- = le processeur 16/30, développé parallèlement au CMF et comprenant une mémoire de 128 K
- = les 2 lignes du CMF sont réservées pour la liaison avec le calculateur central et le calculateur de secours ; si un outil de dialogue local est connecté au CMF, la liaison avec le calculateur de secours sera assurée par un autre coupleur ASX.

Le central doit être équipé :

- = d'un processeur de traitement
- = de mémoire
- = d'un coupleur asynchrone multiplexé (MUX 4) pour dialoguer avec les frontaux
- = du coupleur CMF sur lequel seront connectés l'outil de dialogue graphique et l'imprimante ; le pupitre POP disparaît de la configuration car les fonctions de mise au point et de maintenance sont activées par l'outil de dialogue
- = d'une mémoire de masse (disque dur) et de son coupleur associé.

Le CMF permet à l'utilisateur :

- = de décharger son unité centrale en déportant
 - le traitement des codes d'arrêt multiples et des codes de saturation, désaturation,
 - la gestion d'une partie ou de toute sa procédure de transmission,
 - les fonctions de calendrier secouru et de synchronisation inter-calculateurs
 - la gestion des temporisations et des heures de réveil
- = d'exploiter le chien de garde calculateur et les entrées TOR gérées par le CMF.
- = de supprimer le coupleur TIP de la configuration du frontal puisque cette fonction est assurée dans la gestion des 2 lignes asynchrones du CMF. Le coupleur MUX 4 proposé dans ce type de configuration est un coupleur développé parallèlement au CMF et assurant toutes les fonctions traitées par la partie gestion des fonctions de transmission (voir chapitre III) du CMF.

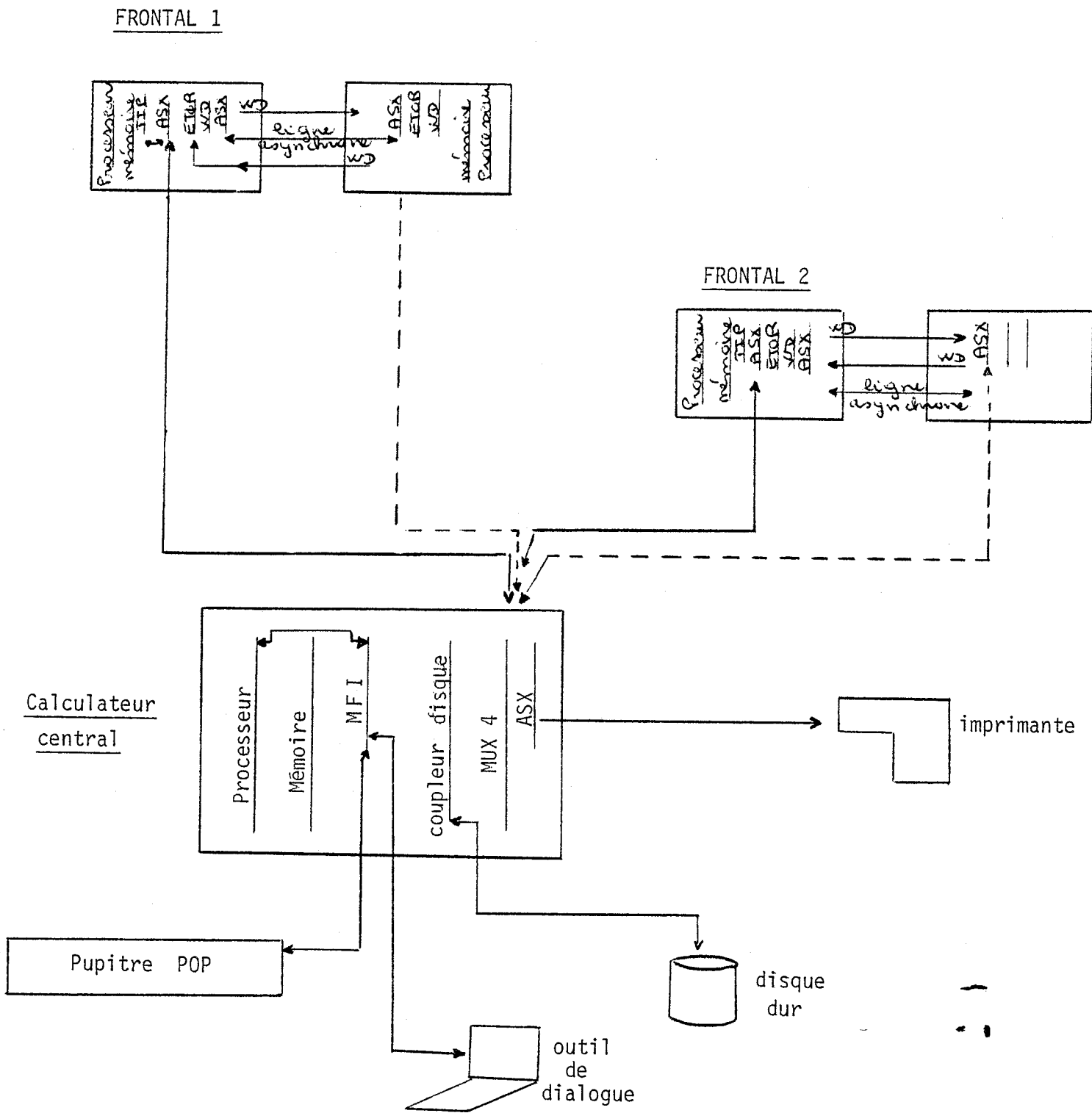


figure 6-3 : Structure hiérarchisée avec 2 frontaux en structure doublée sans CMF

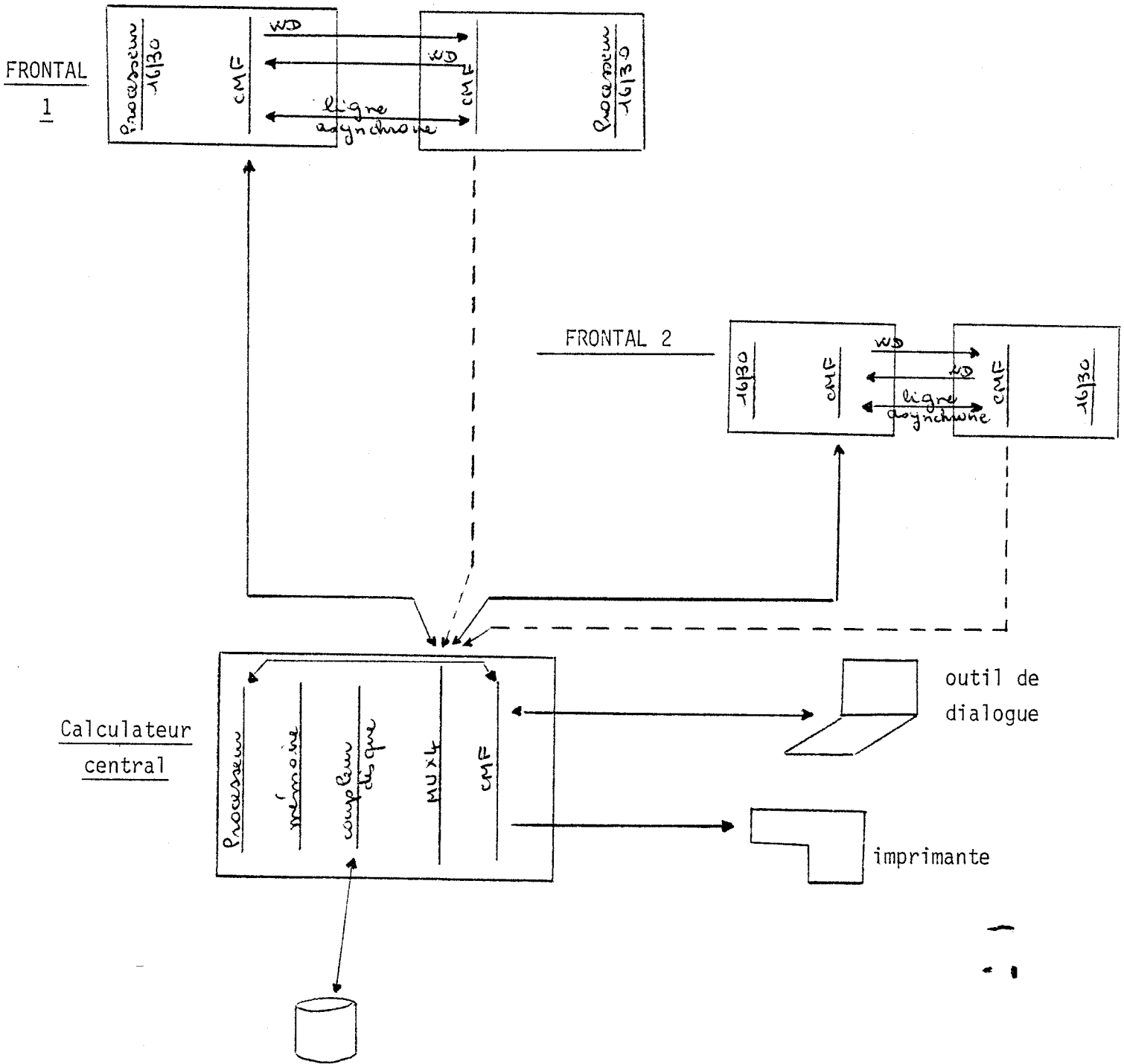


figure 6-4 : structure hiérarchisée avec CMF

CONCLUSION

Nous avons présenté, dans ce mémoire, le coupleur multi-fonctions : nous avons commencé par en expliquer les objectifs puis nous avons présenté l'organisation matérielle à base de 2 micro-processeurs 6809 et l'organisation du micro-logiciel des 2 types de fonctions traitées : fonctions de télétransmission et fonctions temps réel.

Nous avons concentré sur ce coupleur l'ensemble des traitements concernant les fonctions temps réel et gestion de ligne, fonctions jusqu'alors prises en compte par l'unité centrale. Cette solution permet d'alléger la charge du processeur SOLAR par sous - traitance de fonctions.

La décentralisation des traitements fait apparaître la nécessité d'une méthodologie de programmation très structurée et très affinée. Les frontières et les interfaces de communication entre logiciel décentralisé et logiciel SOLAR doivent être parfaitement définies; et cela, afin de faciliter l'écriture et la testabilité des logiciels.

Les fonctions de transmission et les fonctions temps réel sont totalement indépendants : ce choix a été fait pour porter l'architecture matérielle et logicielle des fonctions de transmission sur un coupleur multiplexé intelligent, gérant 4 lignes asynchrones : le MUX4, coupleur en cours de développement et suite logique du coupleur CMF.

Il est, d'autre part, important de rappeler que les choix sont essentiellement des choix économiques liés à une gamme existante et à des logiciels existants.

Je dois également souligner que l'étude et la réalisation de ce coupleur a été possible grâce au travail d'une équipe bénéficiant d'une part de la grande expérience de SEMS dans les systèmes industriels et d'autre part des résultats de recherches effectuées en ce domaine par les utilisateurs et les concepteurs.

Ma participation au développement de ce coupleur a enrichi mon expérience sur le plan humain (par un travail réalisé en équipe, de l'étude jusqu'à la réalisation) et sur le plan technique par l'analyse d'un projet complet. Ce projet englobe des problèmes de gestion

de procédure de transmission, de téléchargement et de télécommande à distance, de fonctions temps réel, sans négliger les problèmes de prix de revient du coupleur et de compatibilité avec une gamme existante, le SOLAR.

BIBLIOGRAPHIE

Tous les manuels compulsés sont des manuels techniques SEMS :

- manuel technique des entrées - sorties SOLAR
- spécification de réalisation du processeur 16/40
- manuel d'exploitation du coupleur ASX-01
- manuel d'exploitation du coupleur ASM-01
- manuel d'exploitation du coupleur MFI/A
- manuel d'exploitation du coupleur TIP
- manuel de référence ICS - TEL - TIP
- spécification de définition du "driver" CMF
- spécification de définition du coupleur CMF