Familles ...MAC & ...SIMPA
Note d'application
Liaison calculateur : protocoles et syntaxe

Date : 17.07.08

Réf. : mi_v0_an03_fr.pdf
Réf. MI : BLN740876.DOC

Révision : 0

Auteur : B.LOPEZ

SOMMAIRE

I – GENERALITES

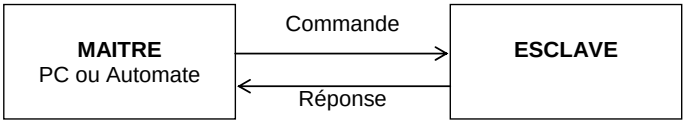
La liaison série présente sur chaque carte ou module des familles développées par Midi Ingénierie, permet le contrôle des modules par un calculateur hôte. Une même liaison série permet de contrôler jusqu'à 64 modules d'un système multiaxe.

Le dialogue avec les modules SIMPA, SIMPA MICROPAS, MICROSIMPA, MAC, DMAC ou microMAC peut être grandement facilité par l'utilisation du logiciel WINSIM2, véritable interface opérateur des modules SIMPA et MAC, ou grâce à la DLL DrvMi pour l'interface avec un programme écrit par le client dans le langage de son choix.

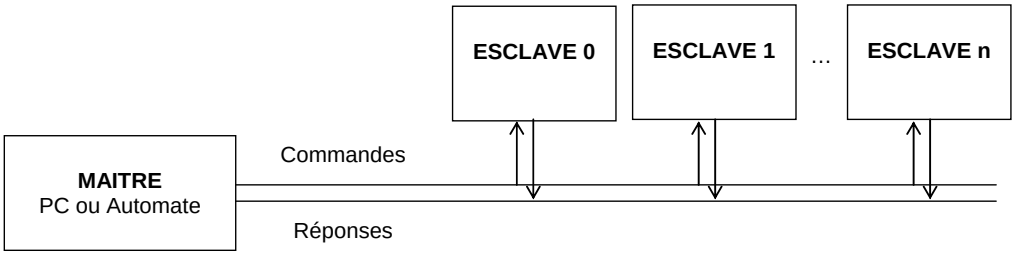
Ces deux logiciels, disponibles uniquement pour les calculateurs compatibles PC sous Windows, suppriment pour l'utilisateur tout l'aspect protocole lié au dialogue. Il ne reste plus à l'utilisateur qu'à donner le contenu de ses commandes.

I.1 – Architecture Maître-Esclave

Le protocole de communication des modules SIMPA, SIMPA MICROPAS, MICROSIMPA, MAC, DMAC et microMAC est du type "Maître-Esclave", c'est-à-dire que le maître (PC ou Automate) envoie des commandes au(x) module(s) qui peut(vent) alors éventuellement retourner une réponse. En aucun cas un module ne peut envoyer de message sans avoir été interrogé par le maître.



Il est possible d'implémenter une structure de type "bus" avec un maître et plusieurs esclaves repérés par leur adresse.

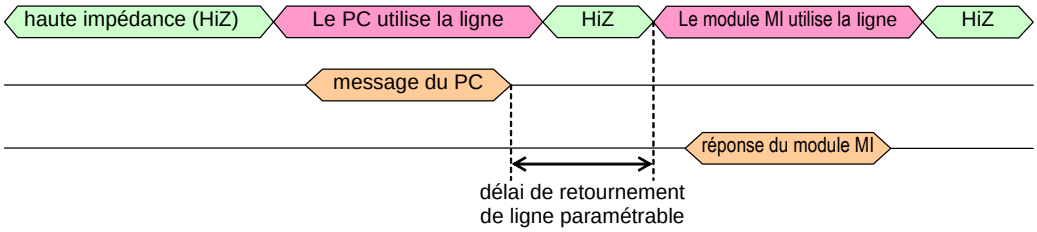


Suivant les cas, la structure de bus est réalisée soit par un chaînage de simples interfaces V24 RS232C, soit par un bus spécifique doit par un bus standard de type RS485.

I.2 – Prise de ligne RS485

Certains modules utilisent un protocole RS485 comme support de communication. Ceci implique que la même "ligne" est utilisée en émission par le PC ou l'automate de commande et en réponse par le module MI.
Au repos, lorsque qu'aucune commande ou réponse n'est émise, la ligne est à l'état de haute impédance. Pour émettre des caractères sur cette ligne, le PC, l'automate ou le module doivent "prendre la ligne".
Deux entités présentes sur la ligne (PC, automate ou module) ne peuvent pas émettre des caractères sur la ligne en même temps.

Conséquence :
Le PC ou l'automate doit repositionner la ligne à haute impédance dès que la commande a été émise.
La variable #LINE_DELAY du MAC17 doit être ajustée pour ne pas répondre avant que la ligne soit à haute impédance.
Le PC ou l'automate ne doit pas envoyer de nouvelle commande tant que la réponse du MAC17 n'est pas totalement émise (notons que toute réponse d'un module commence par le caractère ACK et finit par le caractère XON ou XONERROR, ce qui permet au maître de savoir que l'esclave a fini d'envoyer sa réponse).



II – PROTOCOLES

II.1 – Lignes et protocoles de communication

II.1.1 – Baudrate

L'interface de communication de type RS232 V24 ou RS485 est configurée comme ci-après :

- ✓ vitesse : les vitesses utilisables dépendent du type de module

SIMPA	SIMPA micropas	MAC	microSIMPA, DMAC, microMAC
4800 b	4800 b		
	9600 b	9600 b	9600 b
		19200 b	19200 b
		3840 b	38400 b
			115200 b

- ✓ données : 8 bits
- ✓ bits de stop : 1
- ✓ parité : pas de parité
- ✓ lignes utilisées : Rx et Tx en V24 , Z et /Z en RS485

Les modules peuvent être configurés, suivant les cas, dans l'un des deux modes de dialogue suivants : mode calculateur ou mode console.

Tous les modules placés sur une même ligne de communication doivent utiliser les mêmes modes, vitesses de transmission et protocoles.

II.1.2 – Protocoles et types de modules

Le tableau des protocoles utilisable en fonction des types de module.

	Console	Terminal	Ack Nack	XON/XOFF	XON XOFF étendu
SIMPA	X		X	X	
SIMPA micropas	X		X	X	
MicroSIMPA	X		X	X	
MAC				X	
DMAC					X
microMAC					X

II.2 – Mode calculateur

Il assure un contrôle global du message (nombre de caractères et "check sum").

II.2.1 – Simple protocole ACK/NACK

Tout message reçu (ou émis) donne lieu à acquittement (caractère ACK : 06h) ou non acquittement (caractère NACK : 15h) suivant que la structure du message est correcte ou incorrecte (nombre de caractères et "check sum", mais pas de contrôle de syntaxe).

Les erreurs d'interprétation d'une commande, si elles ne sont pas prises en compte par un protocole XON/XOFF, ne sont signalées que lors du prochain adressage du module par le renvoi, à la place du caractère ACK concernant le message en cours, du caractère BEL (07h). L'émission d'un NACK pour la commande en cours reporte l'émission du BEL au prochain adressage du module.

Les non-acquittements provoquent de la part des modules SIMPA ou SIMPA MICROPAS deux tentatives supplémentaires d'émission avant de positionner un statut d'erreur. Les autres types de modules ne répètent pas le message en cas de non acquittement.

Les modules SIMPA ou SIMPA Micropas ayant émis un message sur la ligne série considèrent l'absence d'acquittement dans un délai de 70 ms ou tout autre caractère que ACK comme un non acquittement.

Remarque : Les anciennes versions des modules SIMPA gèrent le non-acquittement du calculateur hôte. Cette fonctionnalité a été supprimée sur les nouvelles versions pour faciliter l'interfaçage avec le mode "Windows".

II.2.2 – Protocole XON/XOFF

Le protocole ACK/NACK peut être complété par une couche protocolaire supplémentaire permettant à chaque module :

- ✓ d'éviter de recevoir d'autres commandes tant que celles en cours ne sont pas totalement interprétées,
- ✓ de fournir un compte-rendu d'exécution.

Par la transmission après le caractère ACK ou BEL, du caractère XOFF (13h), le module concerné demande la suppression des émissions sur la ligne série, y compris de commandes destinées à d'autres modules. Le module concerné, après complète interprétation de la commande reçue, autorise la reprise du dialogue sur la ligne en émettant le caractère XON (1Ah) si la ou les commandes reçues ont été correctement interprétées ou le caractère XONERREUR (17h) si une commande n'a pas pu être exécutée (erreur de syntaxe ou paramètre hors limite). Dans le cas d'un message comportant plusieurs commandes, si une commande est détectée incorrecte, il y aura génération d'un XONERREUR, les commandes précédentes ayant été exécutées, les commandes suivantes étant perdues.

II.2.3 – Protocole XON/XOFF étendu

Sur les dernières générations de produits MI, le protocole XON/XOFF a été étendu pour permettre de retourner à chaque dialogue avec le module, un caractère appelé X_ETAT résumant l'état de ce dernier en lieu et place du caractère XOFF. Dans ce cas, le dernier caractère du dialogue est toujours le même : XON, ce qui peut faciliter la compatibilité avec certains modems.

S'il tient lieu de caractère XOFF, le caractère X_ETAT n'est retourné qu'après interprétation et n'a, a priori, aucune conséquence puisque le dialogue doit toujours être suspendu jusqu'à réception de XON.

Ce protocole étendu reste donc compatible avec le protocole XON/XOFF initial.

II.2.4 – Gestion des erreurs en fonction du protocole retenu

Protocole ACK/NACK : le caractère d'acquittement retourné par le module adressé (module 0 pour commande globale) dépend :

- ✓ de la syntaxe du message précédent,
- ✓ de la structure du message reçu.

		Structure du message reçu	
		Incorrecte	Correcte
Syntaxe du message précédent	incorrecte	NACK (15 h)	BEL (07 h)
	correcte	NACK (15 h)	ACK (06 h)

Protocole XON/XOFF

Le caractère d'acquittement retourné par le module adressé dépend uniquement de la structure du message reçu, le caractère de libération de ligne dépend de sa syntaxe

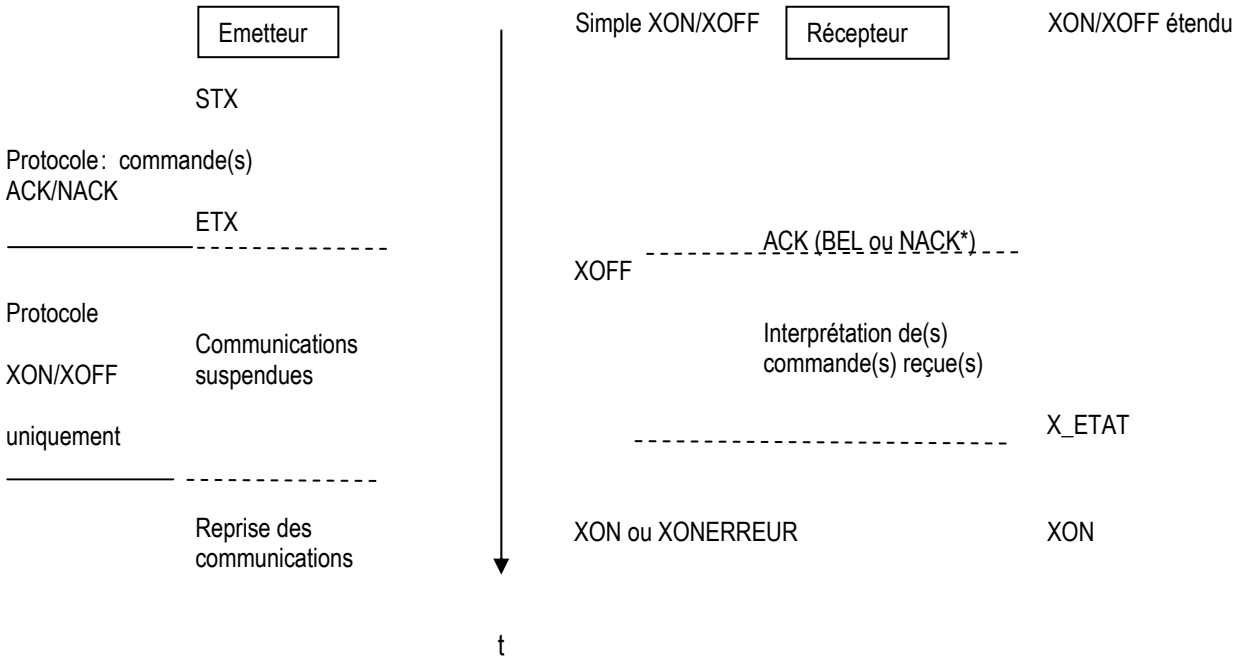
		Acquittement	Libération
Sructure du Message Reçu	correcte	ACK (06 h)	
	incorrecte	NACK (15 h)	
Syntaxe du Message Reçu	correcte		XON (1A h)
	incorrecte	BEL (07 h)*	XON ERREUR (17 h)**

Nota :
Dans le cas de commandes globales, seul le module 0 (obligatoire) assure le renvoi des caractères ACK, NACK, XOFF, XON et XONERREUR. Les autres modules mémorisent les erreurs éventuelles et dans ce cas transmettront le caractère BEL lors de leur prochain adressage individuel.
Lorsqu'un caractère BEL ou XONERREUR est retourné par un module, il est possible d'interroger le module (lecture code état) pour obtenir plus d'informations sur la cause de l'erreur.
Les modules MAC et DMAC ne gèrent pas le caractère BEL.

**Dans le cas d'un message global erroné (non utilisé dans le cas du protocole étendu).*

***Dans le cas du protocole étendu, le caractère XON ERREUR n'est plus utilisé, de sorte que, tant que tout dialogue se termine toujours par le même caractère XON.*

II.2.5 - Résumé des protocoles



* Dans ce cas le caractère XOFF est immédiatement suivi du caractère XON

II.2.6 – Etat du module : X_ETAT

BIT0	Moteur sous tension
BIT1	Moteur en mouvement
BIT2	Busy (moteur non asservi ou position non atteinte)
BIT3	Réservé
BIT4	Réservé
BIT5	Disjonction
BIT6	Warning
BIT7	Toujours forcé à 1

X_ETAT=XOFF_ERREUR (18h) en cas d'erreur d'interprétation.

Warning traduit en général, soit un changement d'état des butées ou la présence d'une erreur de dialogue (syntaxe ou exécution) : se reporter au manuel utilisateur de chaque module.

II.3 – Mode console, mode terminal

Ces protocoles ont été développés pour faciliter le dialogue avec les modules à partir d'une simple console, sans calculateur, ou éventuellement à partir d'un automate.

Dans le mode 'Console', le dialogue de type "écho" reporte les vérifications vers l'utilisateur qui doit s'assurer que les caractères retournés en écho par le module sont bien ceux souhaités. Le module retourne les caractères reçus à l'identique. Pour faciliter le dialogue avec certains automates ou certains logiciels, le protocole console est modifié sur les modules MICROSIMPA : il n'y a plus d'écho des caractères reçus par le module, d'où sa nouvelle dénomination : Terminal.

Dès réception du caractère CR (retour chariot) le module adressé retourne :

CR	
LF	saut de ligne
>	prompt pour prochaine commande

Si une erreur est détectée par le module 0 dans le champ adresse, les caractères CR, LF et "prompt" retournés sont précédés de :

"␣:?" erreur sur le champ adresse du message (renvoyé par le module 0).

Si une erreur est détectée sur les autres champs de la commande par le module adressé, la suite de caractères retournés CR LF est précédée de :

"␣:!" erreur de syntaxe, de paramètre ou commande non autorisée.

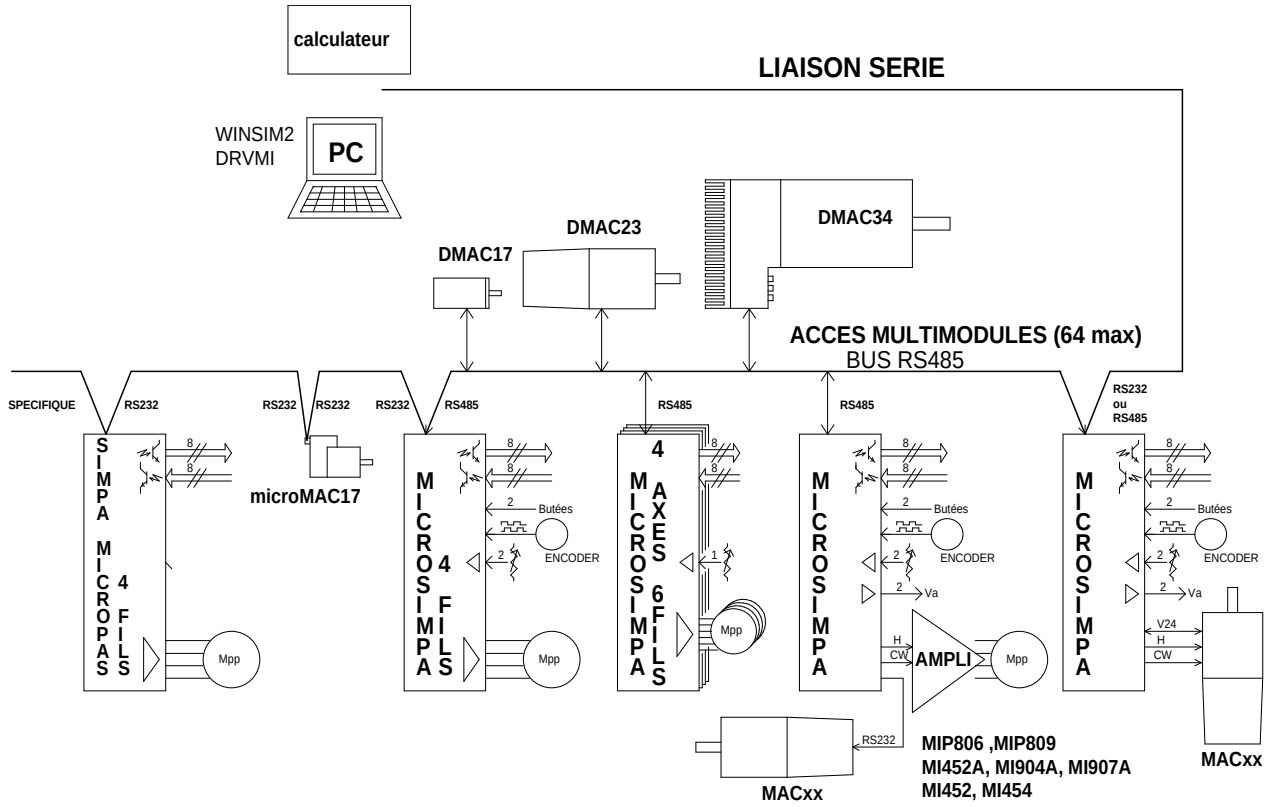
Note : le symbole "␣" matérialise l'espace (20 H)

Remarque : La longueur du message transmis est limitée par la taille du buffer de réception du module. Elle est fixée pour tous les modules de la famille SIMPA à 127 caractères.
Au-delà de cette limite, le message est perdu et le module adressé retournera "␣!CRLF>" à la réception du CR.

II.4 – Configuration

Le mode de communication, la vitesse de transmission et l'adresse des modules sont configurés en général par microswitches pour les familles SIMPA, SIMPA micropas et microSIMPA ou par dialogue série pour les familles MAC, DMAC et microMAC.

Il convient de se référer au manuel utilisateur spécifique du module ou de la carte pour en trouver la description. Comme précisé plus haut, plusieurs modules peuvent être contrôlés par la même liaison série quel que soit leur type. Il est parfaitement possible de contrôler sur le même lien série des modules de type différents tels que SIMPA, SIMPA micropas et microSIMPA, MAC, DMAC ou microMAC avec comme seule contrainte d'utiliser le même protocole et le même baudrate.



La plupart des modules possèdent une interface liaison série de type RS232 V24.

Lorsque l'on n'utilise qu'un seul module, la connexion la plus simple reste la liaison RS232C V24. Dans le cas d'un système multiaxe, les modules des familles microSIMPA ou le DMAC34 assurent la fonction passerelle V24, RS485.

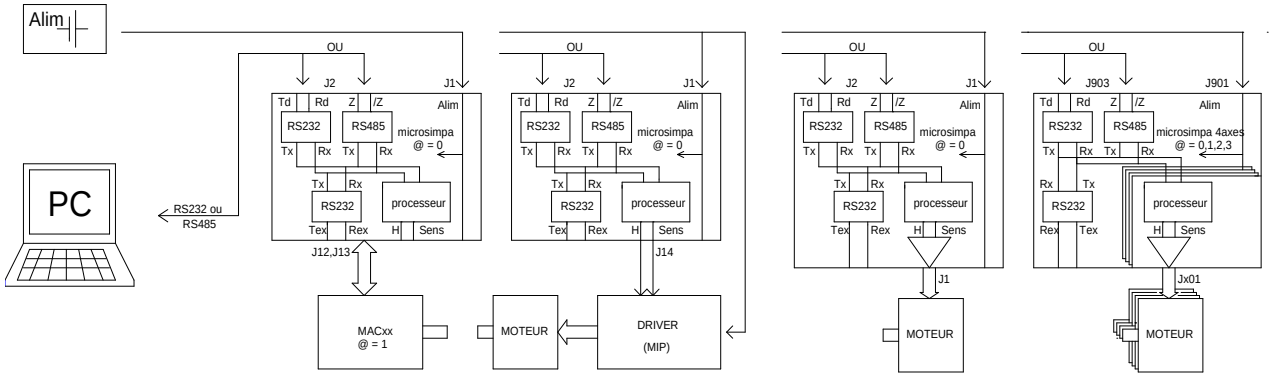
Dans ce cas on utilise le module d'adresse 0 pour s'interfacer avec la ligne RS232 V24 et l'on place les modules suivants en parallèle sur le Bus RS485 géré par ce premier module.

Les modules microMAC se chaînent directement V24 ou se placent sur le bus RS485 lorsqu'ils ont l'option RS485.

Les modules des familles SIMPA et SIMPAmicropas s'utilisent de la même façon que les modules microSIMPA à ceci près que le bus commun n'est pas de type RS485 mais spécifique Midi Ingénierie. Le premier module reçoit la liaison série RS232, les autres modules de ces familles se placent en parallèle sur le bus spécifique Midi Ingénierie.

II.4.1 – Configuration mono axe

Famille microSIMPA



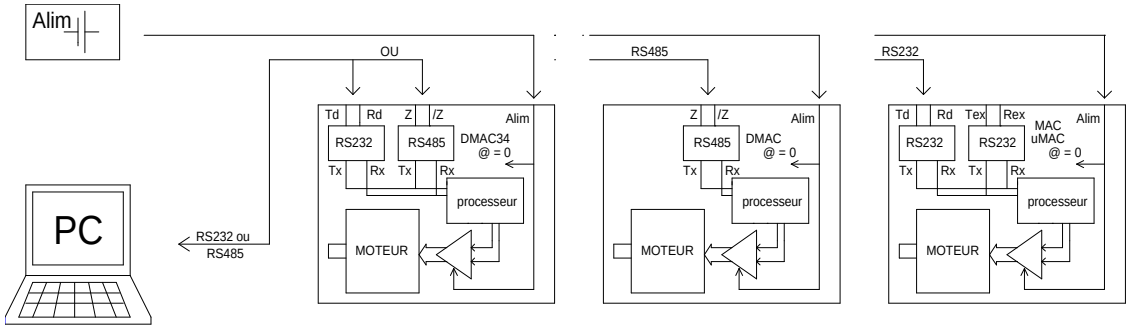
MicroSIMPA +MAXxx

MicroSIMPA + driver MIP, MI907.MI904…

MicroSIMPA 1Axe

MicroSIMPA 4 Axes

Familles MAC, DMAC, microMAC

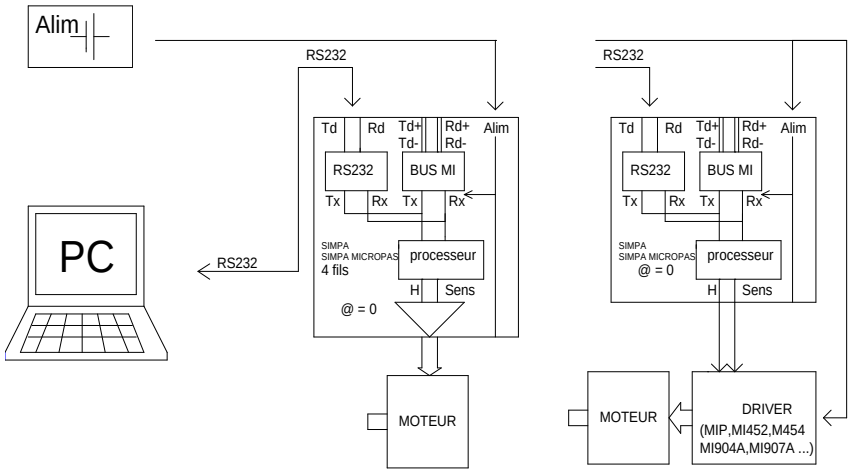


DMAC34
DMAC23

DMAC17

microMAC

Familles SIMPA et SIMPA micropas

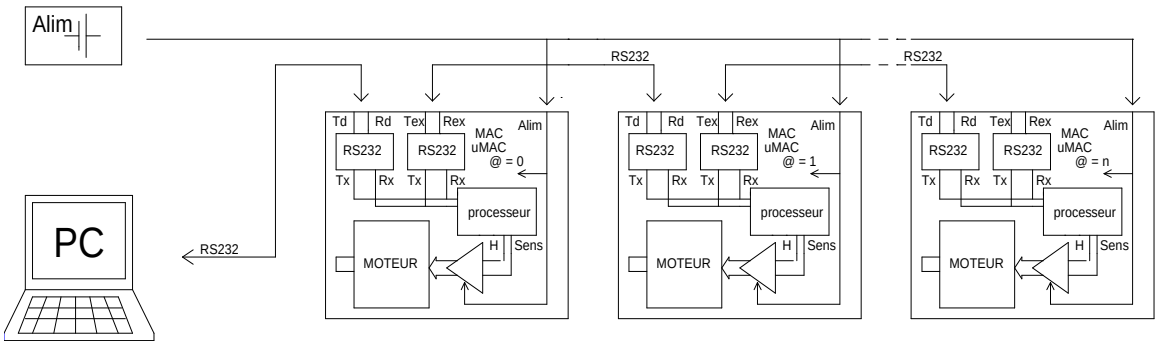


SIMPA ou
SIMPA micropas
1 axe

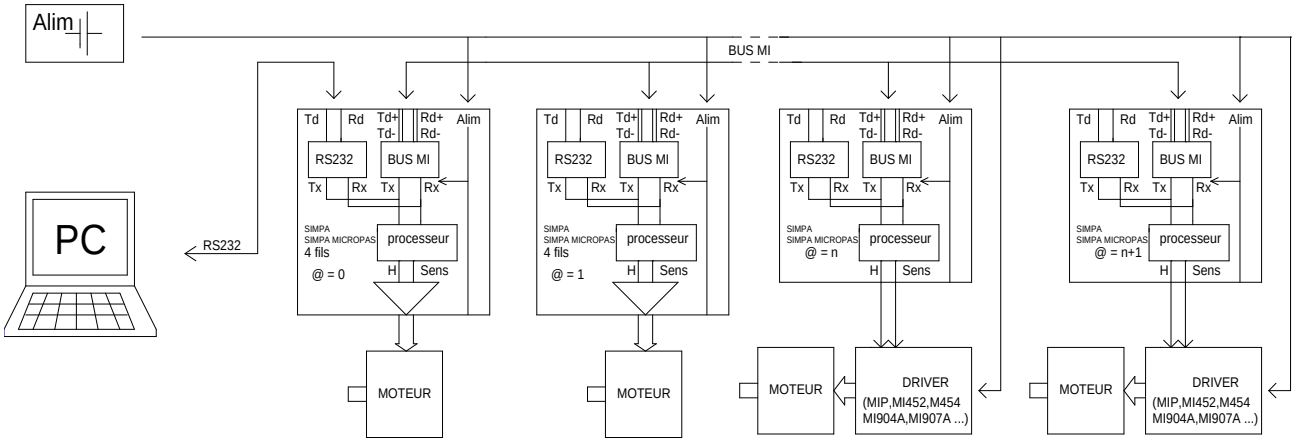
SIMPA ou
SIMPA micropas
+ driver

II.4.2 - Configuration multiaxe

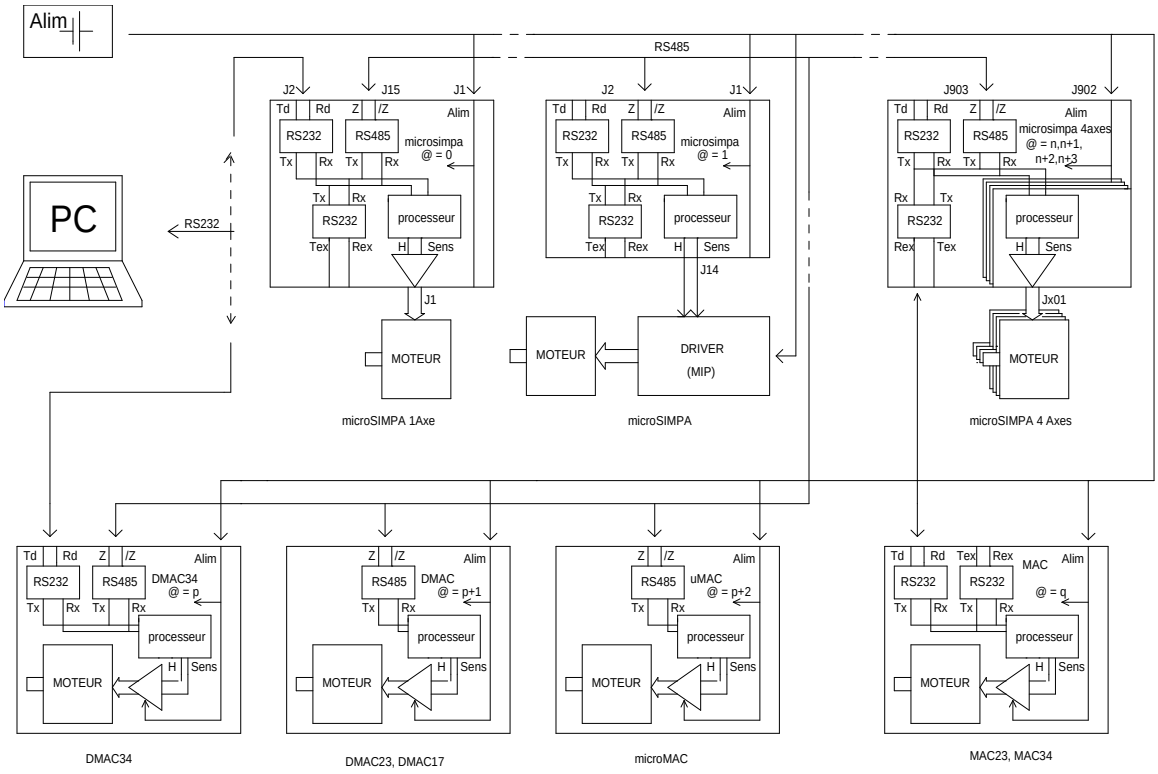
Plusieurs microMAC



Plusieurs SIMPA et SIMPA micropas



Plusieurs familles différentes



III - SYNTAXE DES COMMANDES

III.1 – Conventions utilisées

Dans ce chapitre et dans toutes les documentations des modules des familles SIMPA, MAC, DMAC et microMAC chaque commande est décrite en utilisant un certain nombre de conventions pour exprimer :

- ✓ le type et la taille des paramètres,
- ✓ le type d'adressage (mono ou multimodule),
- ✓ la validité de la commande en fonction de l'état du module ou de son type.

Un paramètre optionnel est représenté entre crochet '[' et ']'

Exemples :
GO [ddddddd] POW [ON] [OFF]

III.1.1 – Type et taille des paramètres

Trois types sont différenciés :

- ✓ numériques : d
- ✓ numériques codés hexadécimal : h
- ✓ alphanumériques : a

La taille maximum est représentée, en général, par le nombre de caractères (d, h ou a) utilisés pour exprimer le paramètre.
Les espaces sont souvent représentés par le caractère ␣.

Exemples :

- ✓ paramètre de déplacement absolu : Pa = ±ddddddd
  signifie que la position absolue peut être donnée en utilisant au maximum 7 digits décimaux, plus un signe.
- ✓ dans la définition d'une phase : Ne : aa␣dd␣dd␣dd␣dd␣dd␣dd␣dd␣dd␣
  signifie que le paramètre Ne doit être constitué de deux caractères alphanumériques suivi de 8 groupes de trois caractères (un espace et deux digits décimaux).

Nota : Pour les valeurs numériques, les 0 non significatifs ainsi que le signe "+" peuvent être omis ou remplacés par des espaces.

Exemple : 00340 = ␣340 = 340 = + 340

III.1.2 – Type d'adressage

Le paramètre adresse du module est représenté par :

- ✓ @ si la commande ne peut être adressée qu'à un seul module à la fois.
- ✓ [@] si la commande peut être adressée à un ou à l'ensemble des modules.
- ✓ rien pour les commandes destinées à l'ensemble des modules.

III.1.3 – Validité de la commande en fonction de l'état du module ou de son type

Dans les documentations SIMPA, SIMPA micropas et microSIMPA, si le caractère " * " précède une commande celle-ci est utilisable quel que soit l'état du module, dans le cas contraire le module doit être hors séquence et le moteur arrêté.

III.2 – Syntaxe générale des commandes élémentaires (en mode calculateur)

III.2.1. – Emission d'un message vers le module

La forme générale d'une commande est la suivante:

STX	T1	T2	T3	A1	A2	X	X	X	X	C1	C2	ETX

STX: caractère ASCII 0x02

T1-T2-T3: trois caractères ASCII codant le nombre de caractères de la commande (format décimal centaines-dizaines-unités). S'applique aux champs A1-A2 et X-X-X-X.

A1-A2: deux caractères ASCII représentant l'adresse du module (format décimal dizaine-unités). Le champ d'adresse est facultatif, si aucune adresse n'est précisée, tous les modules présents sur la ligne reçoivent et exécutent la commande mais seul le module d'adresse 00 (zéro) acquitte et/ou répond.

X-X-X-X: caractères de la commande proprement dite, comme définis dans le manuel utilisateur.

C1-C2: deux caractères de checksum (somme modulo 256). S'applique aux champs A1-A2 et X-X-X-X.

ETX: caractère ASCII 0x03

Exemple: caractères effectivement émis sur la ligne lors de l'envoi de la commande "02MOVE_ON 123" au module 2:

ASCII:	STX	0	1	3	0	2	M	O	V	E	_	O	N	esp	1	2	3	4	B	ETX
décimal:	2	48	49	50	48	50	77	79	86	69	95	79	78	32	49	50	51	52	66	3
hexa:	02	30	31	33	30	32	4D	4F	56	45	5F	4F	4E	20	31	32	33	34	42	03

Taille du message: "02MOVE_ON 123" → 13 caractères → T1-T2-T3 = 0-1-3
Adresse: A1-A2 = 0-2 (pour les adresses inférieures à 10, préciser le zéro non significatif)
Checksum: 48 + 50 + 77 + 79 + 86 + 69 + 95 + 79 + 78 + 32 + 49 + 50 + 51 = 843
843 modulo 256 = 75
hexa(75) = 4B
C1-C2 = 4-B (pour les checksum inférieurs à 16, préciser le zéro non significatif)

III.2.2 – Réception d'un message du module

La forme générale d'une réponse est la suivante:

Dans le cas où le module renvoit une réponse (exemple "READ #POSITION"):

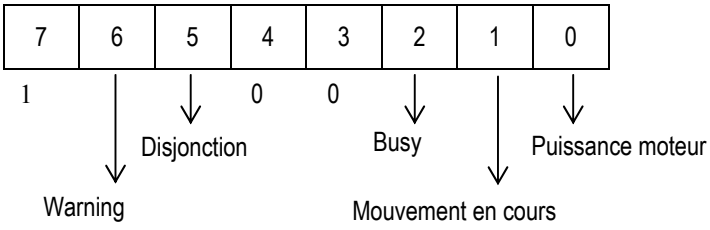
ACK	XETAT	STX	T1	T2	T3	A1	A2	X	X	X	X	C1	C2	ETX	XON

Dans le cas où le module ne renvoit pas de réponse et se contente d'acquitter (exemple "#POSITION:=2905"):

ACK	XETAT	XON

ACK: caractère ASCII 0x06 si la trame reçue est correcte ou **NACK** (caractère ASCII 0x15) dans le cas contraire.

XETAT: caractère représentant l'état du module.



Se reporter à la documentation du module pour le détail

uniquement dans le cas où le module renvoie une réponse

STX: caractère ASCII 0x02

T1-T2-T3: trois caractères ASCII représentant le nombre de caractères du message (format décimal centaines-dizaines-unités). S'applique aux champs A1-A2 et X-X-X-X.

A1-A2: deux caractères ASCII représentant l'adresse du module (format décimal dizaine-unités).

X-X-X-X: caractères de la réponse proprement dite, comme définis dans le manuel utilisateur.

C1-C2: deux caractères ASCII de checksum (somme modulo 256) des caractères de la réponse. S'applique aux champs A1-A2 et X-X-X-X.

ETX: caractère ASCII 0x03

XON: caractère ASCII 0x1A si la commande a été acceptée ou **XONERROR** (caractère ASCII 0x17) dans le cas contraire.

Exemple: Réponse à "00READ #POSITION" (le module zéro se trouve à la position -1000): "00#POS=-1000"

ASCII:	ACK	XETAT	STX	0	1	2	0	0	#	P	O	S	=	-	1	0	0	0	A	0	ETX	XON
décimal:	6	129	2	48	49	50	48	48	35	80	79	83	61	45	49	48	48	48	65	48	3	26
hexa:	06	81	02	30	31	32	30	30	23	50	4F	53	3D	2D	31	30	30	30	41	30	03	1A

Taille du message: "00#POS=-1000" → 12 caractères → T1-T2-T3 = 0-1-2
Adresse: A1-A2 = 0-0 (pour les adresses inférieures à 10, préciser le zéro non significatif)
Checksum: 48 + 48 + 35 + 80 + 79 + 83 + 61 + 45 + 49 + 48 + 48 + 48 = 672
672 modulo 256 = 160
hexa(160) = A0
C1-C2 = A-0 (pour les checksum inférieurs à 16, préciser le zéro non significatif)

III.3 – Etat du module et codes d'erreurs

Suivant les familles de modules, différentes informations concernant d'éventuelles erreurs et l'état du module peuvent être relues.

III.3.1 – Famille SIMPA, SIMPA micropas et microSIMPA

Cette famille ne dispose que d'un registre d'état avant tout réservé à la gestion des codes d'erreurs.

Le registre d'état est positionné lors de l'interprétation de chaque commande reçue par un module.

Le registre peut être relu grâce à la demande QX (RX pour les modules SIMPA pas entier ½ pas)

La réponse à cette demande est :

@EE_code

@	:	adresse du module qui répond
EE	:	type de réponse : état du module
Code	:	caractère précisant l'état du module

* En l'absence de tout défaut le code N est retourné.
* En cas d'anomalie, un code d'erreur est retourné

Il existe deux niveaux d'erreurs :

Niveau 1 : erreurs liées à l'état du module ou erreurs générales.
Niveau 2 : erreurs spécifiques à chaque commande. Ces erreurs sont détaillées plus particulièrement avec chacune des commandes.

Erreurs de niveau 1 : le caractère retourné est alphabétique

- A : commande non autorisée du fait de l'état du module (moteur en mouvement, mode séquence...).
- B : arrêt immédiat du mouvement sur détection butées.
- C : commande inconnue (erreur de syntaxe).
- D : erreur de coordination de phase :
  une commande phase hors de la commande SP
  une commande non phase dans la commande SP
- E : défaut sur entrées logiques (fréquence de basculement supérieure à 10 KHz).
- I : la commande GI est interdite lorsque le module utilisé est un indexeur seul (le courant est alors imposé par l'amplificateur associé).
- M : réinitialisation de la mémoire (suite à un défaut de sauvegarde) ou suite à une demande forcée de réinitialisation mémoire (commande MR0)
- S : passage à la phase 255.
- W : défaut de l'alimentation de puissance.
- X : défaut matériel.

Erreurs de niveau 2 : le caractère retourné est numérique.

- 0 : défaut lié au(x) paramètre(s) de la commande :
  - absence de paramètre
  - trop de paramètres
  - syntaxe du paramètre
- 1 : premier paramètre hors limite (commandes DG, GI, DR, GA, RS, SE, SS, SA, SC, SD, WH, WL et WT) ou séquence inexistante (commande SP)
- 2 : second paramètre hors limite (commandes RS ou SN) ou phase inexistante (commande SP)
- 3 : la séquence précisée dans la commande n'existe pas (commande RS, SS, SA, SC, SD)
- 4 : phase inconnue (commande RS)
  séquence déjà créée (commande SN)
  défaut sur paramètre NE ou NF (commande SP)
- 5 : le nombre de phases restant disponible est insuffisant pour créer la séquence.

III.3.2 – Famille DMAC et microMAC

Les modules des familles DMAC et microMAC gèrent séparément un compte-rendu d'état du module et un compte rendu d'erreur.

Ces deux comptes-rendus sont accessibles via la relecture respective des variables système #STATUS et #ERROR.

Read #STATUS ; READ #ERROR

III.3.2.1 – *#STATUS : état du module*

Retourne l'état du module :

- l'état actif ou non du moteur, la gestion d'un mouvement
- les différents modes de fonctionnement possibles
- la gestion des éléments de sécurité type butée ou non

Mnémonique #STA
Paramètre: Aucun
Description: La relecture de l'état du module renvoie une valeur dont les bits représentent l'état du module:

	bit	Description	
poids fort	bit 32	Mouvement interrompu anormalement	
	bit 31	Erreur (contrôler la variable #ERROR)	❶
	bit 30		
	bit 29	Busy	❶
	bit 28		
	bit 27	Asservissement de position en cours	
	bit 26	Mouvement en cours	
	bit 25	Puissance ON (1) ou OFF (0)	
	bit 24	Warning (voir X_ETAT)	❷
	bit 23	mode Synchro	❶
	bit 22		
	bit 21		
	bit 20	Etat Butée Soft négative	
	bit 19	Etat Butée Soft positive	
	bit 18	Etat Butée Hard négative	
	bit 17	Etat Butée Hard positive	
	bit 16	Séquenceur en mode Edition	
	bit 15	Séquenceur en cours d'exécution	
	bit 14	Polarité des sorties standard (0) ou inversée (1)	
	bit 13	Polarité des entrées standard (0) ou inversée (1)	
	bit 12	Rampes en « S » (1) ou Trapèze (0)	❶
	bit 11		
	bit 10		
	bit 9		
	bit 8	Inversion de polarité des entrées butées hard	❶
	bit 7	Autorisation des butées soft	
	bit 6	Autorisation de la butée hard Inférieure	
	bit 5	Autorisation de la butée hard Supérieure	
	bit 4	Mode Courant Optimisé	❶
	bit 3		
	bit 2		
poids faible	bit 1	Gestion du Standby activée	❷

Exemple: 03READ h#STATUS → 03#STA=h13000800
Relecture de l'état du module d'adresse 3 au format hexadécimal.

❶ uniquement DMAC
❷ uniquement microMAC

III.3.2.2 – *#ERROR : compte rendu d'erreur*

Cette variable rend compte des défauts de fonctionnement type disjonction, des erreurs de paramétrage et des erreurs de dialogue. Les erreurs ne sont effacées que par une écriture à 0 de la variable, sous réserve de la disparition de la cause ayant provoqué le défaut.

Mnémonique: ERR

Paramètres: Error = hhhhhhhh ensemble des flags d'erreur du module (à lire en hexa ou binaire)

Description: Liste des Flags d'erreur du module :

	bit	Description	
poids fort	32		
	31		
	30		
	29		
	28		
	27		
	26		
	25		
	24		
	23	erreur sur le type de paramètre	❷
	22	point hors champ des séquences	❷
	21	dépassement taille mémoire	❷
	20	dépassement taille séquence	❷
	19		
	18	commande non autorisée selon l'état du module	❷
	17	adresse module requise	❷
	16		
	15	commande refusée selon l'état du module	❷
	14	erreur de cohérence entre paramètres	❷
	13	syntaxe incohérente	❷
	12	paramètre non défini	
	11	nom de variable erroné	❷
	10	paramètre fourni non booléen	❷
	9	paramètre fourni non numérique	❷
	8	erreur de calcul (division par 0 …)	
	7	paramètre ou variable hors limite	
	6		
	5	erreur surtension	
	4	erreur sous tension	
	3	erreur court circuit moteur	
	2	disjonction thermique	
poids faible	1		

Exemple #ERR = h200 → le paramètre fourni avec la commande doit être booleen

❶ uniquement DMAC
❷ uniquement microMAC

FICHE DE MODIFICATION DOCS MI 1/1

Documentation concernée : Famille …MAC & …SIMPA – Note d'application – Liaison calculateur : protocoles et syntaxe
réf. : **BLN740876.DOC**

Date et demandeur de la (des) modification(s)	**Type** (corrective ou Evolutive) **et nature de la modification(s) :** (noter chapitre, paragraphe,… concernés)	**Approbation de la (des) modification(s)**	**Mise en place de la (des) modification(s)**	**Indice**
B.LOPEZ Juin 2008	**Création**	Nom : B.LOPEZ Date : Oui ☒ Non ❑ motif du refus :	Personne chargée de la réalisation : N.ROUMEGOUX Date réalisation : Juillet 2008	0