

Famille SIMPA
Note d'application
Liaison calculateur : protocoles et syntaxe



Date : 14.03.05
Référence : BLN1570828.DOC
Révision : 0
Auteur : B.LOPEZ

FICHE DE MODIFICATION DOCS MI
1/1

Documentation concernée : Famille SIMPA – Note d'application – Liaison calculateur : protocoles et syntaxe
 réf. : BLN1570828.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 14/03/05	Création	Nom : B.LOPEZ Date : 14/03/05 Oui <input checked="" type="checkbox"/> Non <input type="checkbox"/> motif du refus :	Personne chargée de la réalisation : N.ROUMEGOUX Date réalisation : 09/03/05	0

SOMMAIRE

I – LIAISON SERIE.....	1
I.1 – Lignes et protocoles de communication.....	1
I.2 – Mode calculateur.....	1
I.2.1 – Simple protocole ACK/NACK.....	1
I.2.2 – Protocole XON/XOFF.....	2
I.2.3 – Gestion des erreurs en fonction du protocole retenu.....	2
I.2.4 – Résumé des protocoles.....	3
I.3 – Mode console, mode terminal.....	3
I.4 – Configuration.....	4
I.4.1 – Configuration mono axe.....	5
I.4.2 – Configuration multi axes.....	5
II – SYNTAXE DES COMMANDES.....	6
II.1 – Conventions utilisées.....	6
II.1.1 – Type et taille des paramètres.....	6
II.1.2 – Type d'adressage.....	6
II.1.3 – Validité de la commande en fonction de l'état du module ou de son type.....	6
II.2 – Syntaxe générale des commandes élémentaires.....	7
II.2.1 – Mode calculateur.....	7
II.2.2 – Mode console ou terminal.....	7
II.3 – Etat du module et codes d'erreurs.....	8

I – LIAISON SERIE

La liaison série présente sur chaque carte ou module de la famille SIMPA permet le contrôle des modules par un ordinateur hôte. Une même liaison série permet de contrôler jusqu'à 64 modules d'un système multi-axes.

Le dialogue avec les modules SIMPA, SIMPA MICROPAS ou MICROSIMPA peut être grandement facilité par l'utilisation du logiciel WINSIM2, véritable interface opérateur des modules SIMPA, 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 ordinateurs 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 – Lignes et protocoles de communication

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	MICRO SIMPA
4800 b	4800 b	
	9600 b	9600 b
		19200 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

Tous les modules de la famille SIMPA peuvent être configurés dans l'un des deux modes de dialogue suivants : mode ordinateur ou mode console.

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

I.2 – Mode ordinateur

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

I.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 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 modules MICRO SIMPA ne répètent pas le message en cas de non acquittement.

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

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

I.2.2 – Protocole XON/XOFF

Le protocole ACK/NACK peut être complété par un protocole 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.

I.2.3 – Gestion des erreurs en fonction du protocole retenu

Protocole ACK/NACK : le caractère d'acquiescement 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'acquiescement retourné par le module adressé dépend uniquement de la structure du message reçu, le caractère de libération de ligne dépend de sa syntaxe

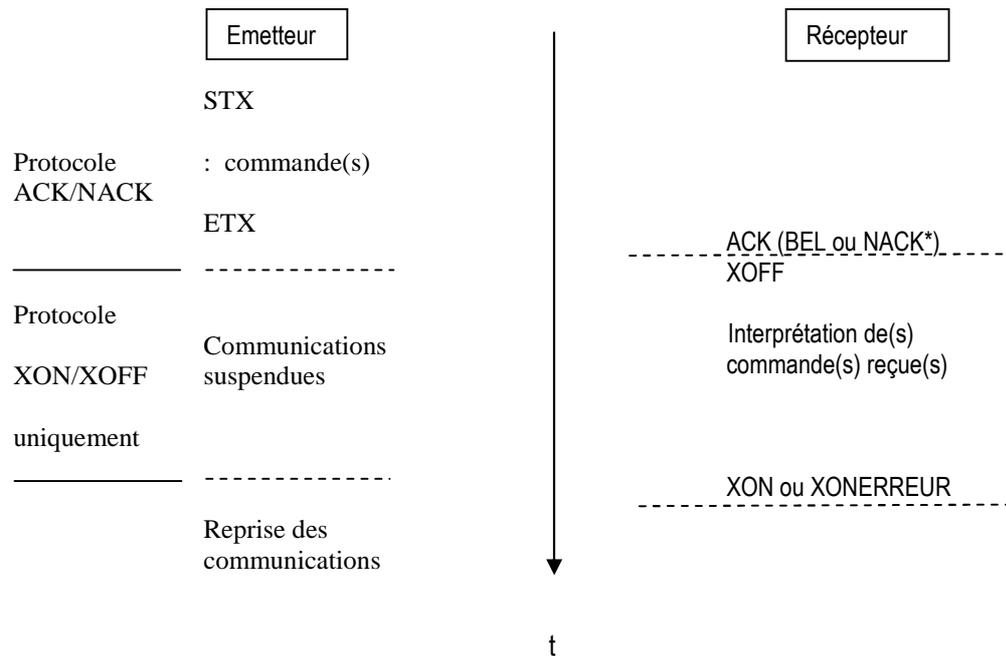
		Acquiescement	Libération
Structure du Message Reçu	correcte	ACK (06 h)	
	incorrecte	NACK (15 h)	
Syntaxe du Message Reçu	correcte		XON (1A h)
	incorrecte	BEL (07 h)*	XON ERREUR (17 h)

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.

1.2.4 - Résumé des protocoles



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

1.3 – 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.

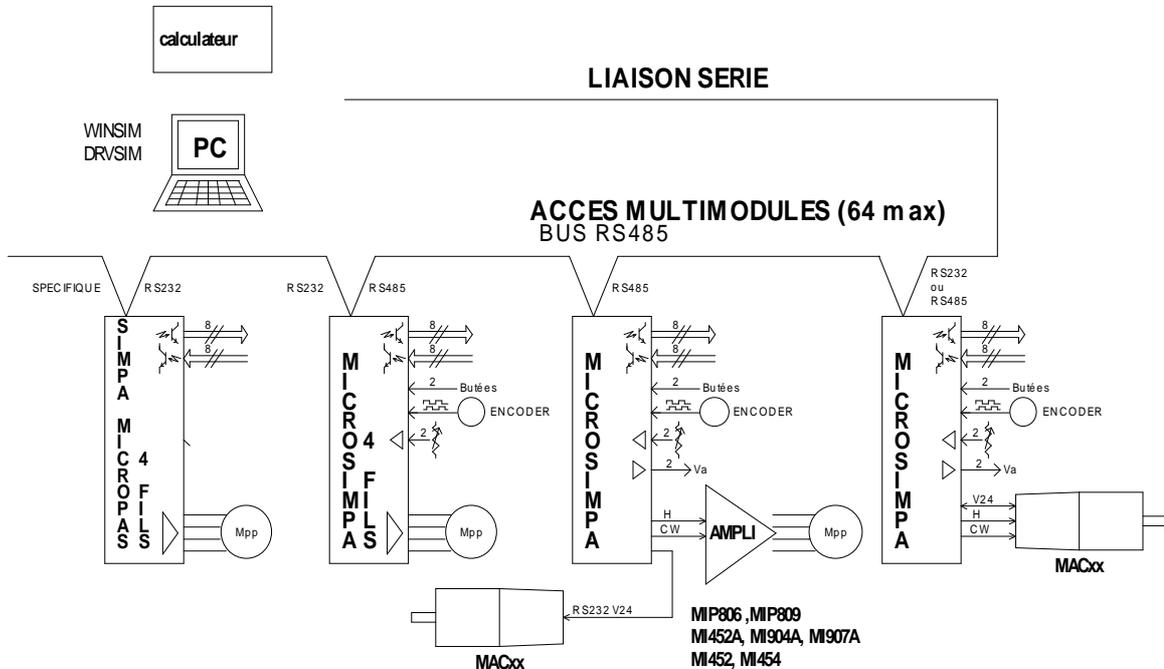
Note : le symbole "_" matérialise l'espace (20 H)

Remarque : La longueur du message transmis est limitée par la taille du buffer de réception du module SIMPA. 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.

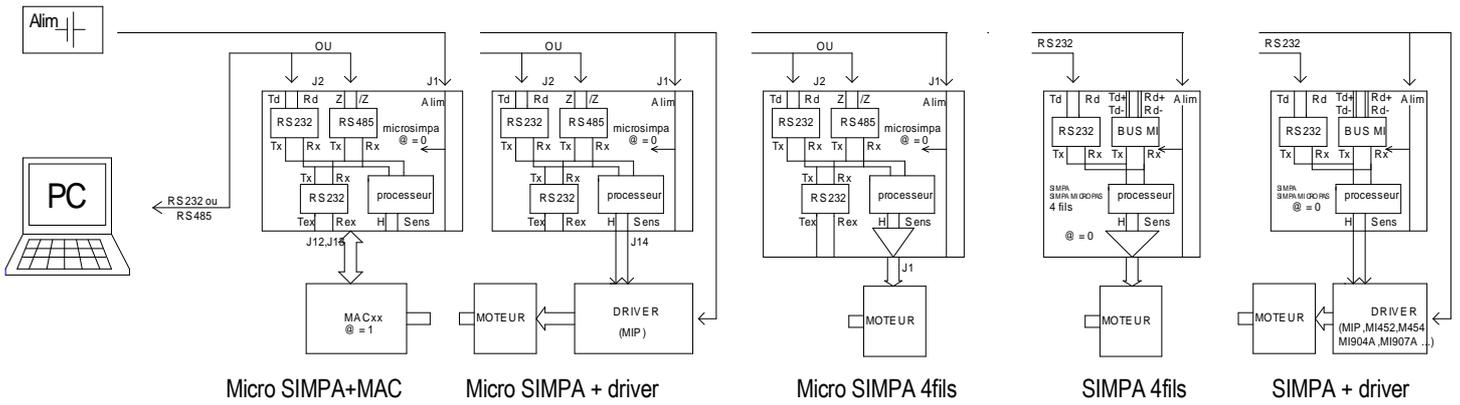
I.4 – Configuration

Le mode de communication, la vitesse de transmission et l'adresse des modules SIMPA sont en général configurés au moyen de microswitches placés sur les modules. 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, avec comme seule contrainte d'utiliser le même protocole et le même baudrate.

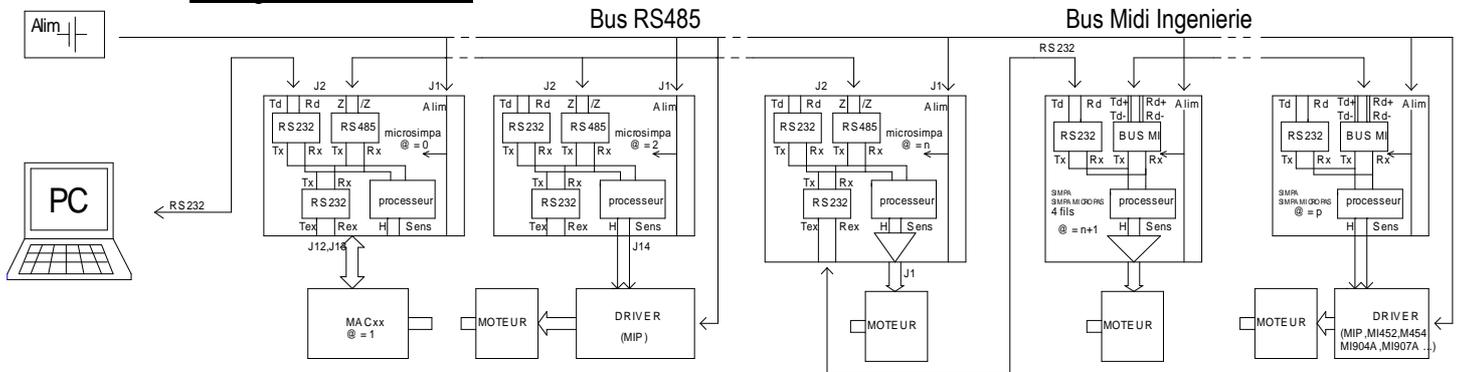


Tous les modules de la famille SIMPA possèdent une interface liaison série type RS232 V24 donc, lorsque l'on n'utilise qu'un seul module, la connexion la plus simple reste la liaison RS232 V24. Dans le cas d'un système multi-axe on utilise toujours le module d'adresse 0 pour se connecter au lien série RS232 V24 et ce premier module sert de passerelle vers un bus série qui permet de placer tous les autres modules en parallèle sur la connexion série.

I.4.1 – Configuration mono axe



I.4.2 - Configuration multi axes



II - SYNTAXE DES COMMANDES

II.1 – Conventions utilisées

Dans ce chapitre et dans toutes les documentations des modules de la famille SIMPA, 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 multi-modules),
- ✓ la validité de la commande en fonction de l'état du module ou de son type.

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

Exemple :

GO [ddddddd]

II.1.1 – Type et taille des paramètres

Trois types sont différenciés :

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

La taille maximum est représentée 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_
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

II.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.

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

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.

II.2 – Syntaxe générale des commandes élémentaires

II.2.1 – Mode calculateur

protocole ACK/NACK avec ou sans protocole XON/XOFF

STX nc [@] cde1 [, cde2, cde3,...] CS ETX

STX et ETX	:	respectivement caractère de début (02h) et fin (03h) de message.
nc	:	nombre total sur 3 digits codés décimal des caractères transmis hormis STX, ETX, nc et CS. Le nombre de caractères doit être inférieur à 128 ($nc \leq 127$)
[]	:	caractères optionnels.
@	:	adresse du module (00 à 63) auquel est destiné le message. Si l'adresse est omise, le message est destiné à l'ensemble des modules présents sur la ligne. Un module d'adresse 00 est indispensable dans toutes les configurations.
Cdei	:	libellé de la commande i avec ses paramètres.
,	:	séparation de commandes si plusieurs commandes sont regroupées dans un même message.
CS	:	contrôle de validité de type "check sum". Codé sur deux caractères ASCII représentant chacun un des deux digits hexadécimaux obtenus en faisant la somme arithmétique modulo 256 de tous les caractères du message sauf STX, nc, CS et ETX.

Les différents champs des commandes peuvent être séparés par un ou plusieurs espaces sauf entre le champ adresse (@) et la première commande (cde 1) où aucun espace n'est autorisé.

Exemple : Commande MR sur l'axe 00

STX	"0"	"0"	"4"	"0"	"0"	"M"	"R"	"F"	"F"	EXT
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Soit en hexadécimal

02	30	30	34	30	30	4D	52	46	46	03
----	----	----	----	----	----	----	----	----	----	----

Somme : $30 + 30 + 4D + 52 = FF$

II.2.2 – Mode console ou terminal

avec ou sans écho

[@] cde1 [, cde2, cde3, ...] CR

CR : retour chariot

Remarque : le nombre de caractères transmis entre 2 CR ne doit pas excéder 127 caractères !

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

Un 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.
  
```