

# ***Famille SIMPA***

## ***Note d'application***

Evolution des produits SIMPA, SIMPA Micropas et MICROSIMPA



Date : 14.03.05  
Référence : BLN1570829.DOC  
Révision : 0  
Auteur : B.LOPEZ

**FICHE DE MODIFICATION DOCS MI**
**1/1**

Documentation concernée : Famille SIMPA – Note d'application

Evolution des produits SIMPA, SIMPA Micropas et Microsimpa

réf. : BLN1570829.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 : 10/03/05	0

## SOMMAIRE

I – INTRODUCTION .....	1
II – EVOLUTION ENTRE PAS ENTIER ET MICROPAS .....	1
II.1 – Résolution .....	1
II.1.1 – Modules pas entier .....	1
II.1.2 – Modules micropas.....	1
II.2 – Limites et cohérence des paramètres de mouvement .....	2
II.2.1 – Modules pas entier ½ pas.....	2
II.2.2 – Modules micropas.....	2
II.3 – Variables .....	3
II.4 – Entrées, sorties et Standby .....	3
II.5 – Commandes immédiates .....	4
II.6 – Gestion des séquences.....	5
III – EVOLUTION ENTRE SIMPA MICROPAS ET MICROSIMPA.....	6
III.1 – Evolution des protocoles de dialogue.....	6
III.1.1 – Baudrate .....	6
III.1.2 – Protocole console ou terminal .....	6
III.2 – Evolution de la gestion des variables.....	7
III.2.1 – Types de variables.....	7
III.2.2 – Assignation des variables .....	7
III.2.3 – Calculs et tests sur les variables.....	8
III.2.4 – Relecture des variables .....	8
III.2.5 – Consigne du dernier mouvement .....	8
III.3 – Evolution de la gestion des séquences.....	9
III.3.1 – Nombre de séquences et de phases.....	9
III.3.2 – Mémorisation des séquences.....	9
III.3.3 – Ecriture des phases .....	9

## I – INTRODUCTION

Les produits de la famille SIMPA développés par la société Midi Ingénierie sont des unités intelligentes destinées à la commande de moteurs pas à pas. Ces modules intègrent la fonction indexeur et, pour la plupart, l'électronique de puissance propre à la commande des moteurs pas à pas. Certaines cartes intègrent jusqu'à quatre modules SIMPA.

Chaque module SIMPA comprend : l'ensemble des fonctions et composants nécessaires au contrôle d'un axe moteur (pas à pas) une unité logique à microprocesseur et, suivant le cas, une unité de puissance à découpage de courant pouvant délivrer une intensité de 2A à 7A efficace ou plus par phase (3A à 10A crête) sous une tension d'alimentation de 15 à 100V suivant le type de carte.

Chaque module dispose d'une liaison série lui permettant de communiquer avec un ordinateur ou un automate. Celle-ci permet de définir à distance les paramètres de fonctionnement de chaque module et d'y mémoriser des séquences de mouvement prédéfinies. Grâce à ces séquences, les modules SIMPA peuvent fonctionner de façon autonome et réaliser de véritables petits automates.

Au fur et à mesure du développement de cette famille, ces produits ont acquis des fonctionnalités et des performances de plus en plus importantes. Pour réaliser ces évolutions certains modes d'utilisation ou de programmation ont été légèrement modifiés, et si une compatibilité ascendante a été recherchée, elle n'a pas pu être obtenue dans tous les cas de figures. Aussi le présent document s'attache t'il à décrire ces différences et à montrer les modifications à apporter pour passer d'un produit de la famille à un autre.

Les différents produits de la famille SIMPA peuvent être classés en 3 types dans l'ordre des évolutions :

- SIMPA pas entier
- SIMPA micro pas
- MICROSIMPA

L'évolution entre les deux premiers types est avant tout liée au mode de contrôle des moteurs : fonctionnement en pas entier ou demi pas en regard d'un fonctionnement en mode micropas, ce qui se traduit principalement par un ajout de commandes nécessaires au paramétrage du fonctionnement en micropas. L'évolution entre les deux derniers types est liée à l'augmentation des performances du microprocesseur de gestion utilisé dans les modules et au mode de mémorisation des données dans le module, ce qui autorise une augmentation des fonctionnalités au niveau notamment des variables.

## II – EVOLUTION ENTRE PAS ENTIER ET MICROPAS

### II.1 – Résolution

#### II.1.1 – Modules pas entier

Hormis la possibilité de configurer en général par switch les modules avec ampli driver pour fonctionner en pas entier ou demi pas, il n'y a pas à proprement parler de prise en compte de la résolution réelle du module. Les modules ne gèrent en fait que des incréments de position, aussi, les vitesses doivent-elles être définies en pas/s pour un fonctionnement en pas entier et en 1/2 pas par seconde pour un fonctionnement en demi pas.

#### II.1.2 – Modules micropas

La résolution des modules micropas peut être choisie entre 1 et 256 micropas/pas. Dans le cas d'un simple indexeur il convient en général de choisir une résolution identique pour l'indexeur et l'amplificateur. Dans le cas d'un module intégrant l'amplificateur, seules les résolutions suivantes sont accessibles : 1, 2, 4, 8, 16, 32, 64 (en micropas/pas). Dans tous les cas, quelle que soit la résolution choisie, les vitesses sont toujours définies en pas/s, les déplacements et positions sont exprimés dans la résolution choisie.

## II.2 – Limites et cohérence des paramètres de mouvement

### II.2.1 – Modules pas entier ½ pas

#### II.2.1.1 – Limites générales

- ✓  $20 \leq V_{min} < V_{max} \leq 20000$  (pas/s ou ½ pas/s)
- ✓  $\mu = 1$  (uniquement 'non programmable')
- ✓  $1 \leq TR \leq 65535$  (ms)
- ✓  $-999998 \leq N_{relatif} \leq +999998$  (pas ou ½ pas)
- ✓  $-8388608 \leq N_{absolu} \leq +8388607$  (pas ou ½ pas)

#### II.2.1.2 – Cohérence entre paramètres

- ✓  $V_{max} * TR < 65 \cdot 10^6$

### II.2.2 – Modules micropas

#### II.2.2.1 – Limites générales

- ✓  $\frac{62}{\mu} \leq V_{min} < V_{max} \leq 20000$  (pas/s)
- ✓  $1 \leq \mu \leq 256$ 
  - pour les modules micropas sans puissance intégrée (SIMPA micropas, SIMPA micropas Face Avant ou MICROSIMPA ...)
  - pour les modules micropas avec puissance intégrée (SIMPA micropas 1 axe, SIMPA micropas 1 axe Face Avant, SIMPA micropas 4 fils ou SIMPA micropas 4 fils Face Avant, MICROSIMPA 4 fils...)
- ✓  $1 \leq TR \leq 65535$  (ms)
- ✓  $-2147483647 \leq N_{relatif} \leq +2147483647$  (micropas)
- ✓  $-2147483647 \leq N_{absolu} \leq +2147483647$  (micropas)

#### II.2.2.2 – Cohérence entre paramètres

- ✓  $\frac{10^{-3}}{\mu * Tr(ms)} \leq V_{min} (p/s) \leq \frac{63,75 \cdot 10^6}{\mu * Tr(ms)}$
- ✓  $\mu * V_{min} \leq 20000$  (p/s)
- ✓  $\mu * V_{max} \leq 1,28 \cdot 10^6$  (p/s)

$\mu$	X	Tr (ms)		
		$\mu V_{max} \leq 16 \text{ KHz}$	$16 \text{ KHz} < \mu V_{max} \leq X$	$X < V_{max}$
1		$Tr < \frac{63,75 \cdot 10^6}{V_{MAX}}$		
2..3	32 KHz	$Tr \leq \frac{63,75 \cdot 10^6}{\mu * V_{max}}$	$Tr \leq 3984 \text{ ms}$	$Tr \leq \frac{127,50 \cdot 10^6}{\mu * V_{max}}$
4..7	64 KHz			$Tr \leq \frac{255 \cdot 10^6}{\mu * V_{max}}$
8..15	128 KHz			$Tr \leq \frac{510 \cdot 10^6}{\mu * V_{max}}$
16..31	256 KHz			$Tr \leq \frac{1,2 \cdot 10^9}{\mu * V_{max}}$
32..63	512 KHz			$Tr \leq \frac{2,4 \cdot 10^9}{\mu * V_{max}}$
64..256	1024 KHz			$Tr \leq \frac{4,08 \cdot 10^9}{\mu * V_{max}}$

### II.3 – Variables

Les modules pas entier ½ pas ne connaissent pas la notion de variable. Les modules micropas définissent 32 variables codées chacune sur 32 bits. Chaque variable possède une valeur d'initialisation qui peut être imposée par l'utilisateur.

### II.4 – Entrées, sorties et Standby

Les modules SIMPA et SIMPA MICROPAS gèrent le même nombre d'entrées et sorties : 8 entrées, 8 sorties.

Pour les versions pas entier ½ pas les sorties sont réinitialisées à la fin de chaque séquence alors que l'état de ces dernières est conservé d'une séquence à l'autre dans les versions micropas.

Les versions micropas possèdent en plus des 8 sorties générales, 2 sorties dédiées pour gérer les fonctionnalités de boost et standby, alors que sur les versions pas entier ½ pas la sortie logique 1 est réservée au contrôle du courant de standby lors des commandes immédiates, en séquence contrairement aux modules micropas, le standby doit être géré par l'utilisateur.

Sur les versions micropas il est possible de gérer les sorties logiques bit à bit en utilisant un masque pour ne modifier que les bits désirés. De plus l'état des sorties peut être modifié en fonction de la position moteur grâce au mode de gestion déporté des sorties.

## II.5 – Commandes immédiates

Fonction	SIMPA	SIMPA micropas
Reset général du module	MR	MR [Z] (Z : configuration usine)
Sélection du mode de fonctionnement Butée	MB	MB [H]/[L] (H/L : polarité des E/S)
Annulation du mode de fonctionnement Butée	MN	MN [H]/[L] (H/L : polarité des E/S)
Mode de gestion du courant		MS [N][S][B] (Nom /Stb/ Stb + Boost)
Valeur décimale d'une variable		PD #n:v
Valeur hexadécimale d'une variable		PH #n:v
Définition globale des variables et synchro		PG v1[: v2...[: :v32]]
Vitesse de consigne (20 à 20000 p/s) palier	WH V <sub>max</sub>	WH V <sub>max</sub>
Vitesse de démarrage (20 à 20000 p/s)	WL V <sub>min</sub>	WL V <sub>min</sub>
Durée rampe accélération en ms	WT t <sub>a</sub> (accélération = décélération)	WT t <sub>a</sub> :t <sub>d</sub> (t <sub>a</sub> = accélération, t <sub>d</sub> =décélération)
Résolution en micro pas par pas		WN μ
n micro pas tolérés en glissement	DG n	DG n
Initialisation de la position absolue	DI	DI
Définition de la Référence absolue	DP Pr	DP Pr
Exécution d'un mouvement absolu	GA Pa (-2 <sup>23</sup> -1 < Pa < 2 <sup>23</sup> -1)	GA Pa (-2 <sup>31</sup> -1 < Pa < 2 <sup>31</sup> -1)
Exécution d'un mouvement relatif (± n micropas)	DR n, [D+],[ D-] GO	GO [±]/[n]
arrêt immédiat mouvement ou séquence	GS	GS
Arrêt avec décélération	GE	GE
mouvement continu	GF (vitesse = WH)	GF[+]/[-][v] (v = vitesse mouvement)
Retour origine	GH	GH
Imoteur = Inom * Im/255	GI Im	GI Im
Puissance moteur ON	GM	GM
Coupe puissance moteur	GR	GR
Positionnement des sorties logiques	GL Out (global)	GL Out[:MSq] (MSq = masque)
gestion Sortie Logique déportée		GP P[ ][M]
Lecture paramètres locaux	RL EL V <sub>min</sub> V <sub>max</sub> t <sub>cns</sub> I <sub>m</sub> mode gl	QL EL WL:V <sub>min</sub> WH:V <sub>max</sub> WT:t <sub>a</sub> :t <sub>d</sub> WN:rés DR:±cns GI:I <sub>m</sub> DG:gl MD :M Pol
Lecture compteur de micropas, états E/S	RP EP Pos_abs Entrées Sorties	QP EP Pos_abs Entrées Sorties
Relecture variable n		QN #n[H] EN # n : v [, (V)]
Lecture version et indice du logiciel	RV EV VR code	QV EV VR code code2 texte
Lecture code état module	RX EE_Code_état	QX EE_Code_état
Lecture de la phase np de la séquence ns	QS ns np ES ns np Nat Cns NS ns NE_E1... E8 NL ss NO out NF NQ ss	QS ns np ES ns np Nat Cns NS ns NE_E1... E8 NL ss NO out NF NQ ss
Suivi des séquences et mouvements	RD ED ns np S Nat pos in out ctrl ss Rc	QD ED ns np S Nat pos in out ctrl ss Rc
Suivi des séquences et des phases		QG EG_Séquence_Phase_Sens_Contrôle
État sorties logiques déportées (GL/GP)		QC

## II.6 – Gestion des séquences

Fonction	SIMPA	SIMPA micro pas
Sélection séquence de démarrage	SD ns (1 ≤ ns ≤ 99)	SD ns (1 ≤ ns ≤ 100)
Suppression séquence de démarrage	SR	SR
Effacement séquence	SE ns	SE ns
Lancement d'une séquence	SS ns (1 ≤ ns ≤ 99)	SS ns (1 ≤ ns ≤ 100)
Création d'une séquence	SN ns np (1 ≤ ns ≤ 99 ; 1 ≤ np ≤ 50)	SN ns np (1 ≤ ns ≤ 100 ; 1 ≤ np ≤ 200)
Définition de phase	SP ns np Sens Na Cns [ NL sc] [ NE X1.. X8] [NS ps][NO out] [ NQ sq][ NF X1.. X8] [NS@1[:@2[:@3]]]	SP ns np Na Cns [ NL sc] [ NE X1.. X8] [NS ps][NO out][:msq]] [ NQ sq][ NF X1.. X8] [NS@1[:@2[:@3]]]
Directives de phase		
mouvement absolu	NX	NX
mouvement relatif	NP	NP
retour origine (HOME)	NH	NH
accélération	NA	NA
décélération	ND	ND
vitesse constante	NV	NV
modification vitesse palier	NC	NC
détection glissement	NG	NG
attente	NW	NW
RAZ position absolue	NZ	NZ
puissance moteur ON	NT	NT
puissance moteur OFF	NU	NU
attente	NW	NW
modification de variable		PV
addition sur variable		PA
transfert CPA variable		PC
valeur d'initialisation d'une variable		PI
valeur d'init variable = valeur courante		PR
test variable		PT

### III – EVOLUTION ENTRE SIMPA MICROPAS ET MICROSIMPA

Hormis quelques évolutions sur les protocoles de dialogue, les différences entre les modules SIMPA MICROPAS et les modules MICROSIMPA ne se situent qu'au niveau du jeu de commandes, particulièrement pour la manipulation des variables et la mémorisation des séquences.

#### III.1 – Evolution des protocoles de dialogue

##### III.1.1 – Baudrate

Baurate	SIMPA MICROPAS	MICROSIMPA
4800 b	*	
9600 b	*	*
19200 b		*
38400 b		*
115200 b		*

##### III.1.2 – Protocole console ou terminal

Le protocole console a été développé pour faciliter le dialogue avec les modules à partir d'une simple console, sans calculateur, ou éventuellement à partir d'un automate.

Dans ce mode, 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.

Dans les deux cas l'encadrement du message reste le même : 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)

## III.2 – Evolution de la gestion des variables

### III.2.1 – Types de variables

Les modules SIMPA Micropas ne gèrent qu'un seul type de variable notée #1 à #32. Chaque variable possède sa valeur courante qui évolue au gré des opérations qui y sont appliquées, et une valeur d'initialisation qui est mémorisée d'une mise sous tension à l'autre et qui permet de forcer la valeur de la variable à la mise sous tension : ces 2 valeurs sont données simultanément par les commandes PD # n : v ou PH # n : v.

Les modules MICROSIMPA gèrent 3 types de variables :

- ✓ Les variables utilisateurs simples notées : #1 à #32.
- ✓ Les variables utilisateurs mémorisées notées : #M1 à #M32.
- ✓ Les variable systèmes : fonctionnalité totalement nouvelle

La notion de valeur d'initialisation de variable n'existe plus. La mémorisation des variables mémorisées se fait sur leur valeur courante au moment de la coupure de l'alimentation. Pour reproduire l'ancienne fonctionnalité d'initialisation il convient d'utiliser une variable mémorisée pour conserver la valeur d'initialisation et une variable simple pour travailler sur la valeur courante, comme décrit par l'exemple suivant :

```

PO #M5 := 345          ; pour définir la valeur d'initialisation
et
PO #5 := #M5          ; pour réinitialiser la variable en début de séquence.
    
```

### III.2.2 – Assignation des variables

Avec les modules SIMPA Micropas les commandes d'assignation sont :

En commande immédiate :

DI	initialisation du compteur de pas absolu
DP Pr	force la position de référence
PD #n : v	pour définir en décimal la valeur d'une variable
PH #n : v	pour définir en hexadécimal la valeur d'une variable

En directive de séquence :

PV #n : v ou PV #n : # m	affectation par une constante ou une variable
PC #n	recopie de la valeur du compteur de position absolue dans une variable
PI #n	force la valeur de réinitialisation à la valeur courante
PR #n	réinitialisation d'une variable avec sa valeur d'init

Avec les modules MICROSIMPA la seule commande d'assignation est :

PO #n := v      ou      PO #n := #m      que ce soit en commande immédiate ou en séquence.

Il convient donc de traduire les anciennes commandes par :

DI	⇒	PO #CPA := 0	
DP Pr	⇒	PO #CPA := Pr	
PH #n : v	⇒	PO #n := hv	( v valeur exprimée en hexa )
PV #n : v	⇒	PO #n := v	
PC #n	⇒	PO #n := #CPA	
PI #n	⇒	PO #Mn := #n	
PR #n	⇒	PO #n := #Mn	

### III.2.3 – Calculs et tests sur les variables

Avec les modules SIMPA MICROPAS, seule l'addition est accessible et 3 formes de tests sont utilisables :

PA #n : v (ou PA #n : #m)	↔	#n = #n + v (ou #n = #n + #m)
PT #n : v	NS @1 :@2 :@3	↔ #n ?? v
PT #n : #m	NS @1 :@2 :@3	↔ #n ?? #m
PT #n	NS @1 :@2	↔ #n = #n - 1 ; n ?? 0

Avec les modules MICROSIMPA une seule forme de commande permet de réaliser toute une série d'opérations ou de tests :

PO #n := #m (op) #p [NS @1[:@2[:@3]]]

Ou (op) peut prendre les signes suivants : + , - , \* , / , % , & , | , ^ , >> , ?

Il convient donc de traduire les anciennes commandes par :

PA #n : v	⇒	PO #n := #n + v
PA #n : #m	⇒	PO #n := #n + #m
PT #n : #v NS @1[:@2[:@3]]	⇒	PO #n ? v NS @1 :@2 :@3
PT #n : #m NS @1[:@2[:@3]]	⇒	PO #n ? #m NS @1 :@2 :@3
PT #n NS @1[:@2]	⇒	PO #n := #n - 1 NS @1 :@2 :@2

### III.2.4 – Relecture des variables

La commande de relecture QN est remplacée par la commande QR qui permet la relecture simultanée de plusieurs variables; la forme de la réponse est légèrement différente :

QN #n	⇒	00EN #n :v [V] doit être traduite par
QR #n #Mn	⇒	00 #n=v #Mn=V (si l'on associe #Mn à #n pour lui servir de valeur d'initialisation)

### III.2.5 – Consigne du dernier mouvement

Sur les modules SIMPA MICROPAS le sens du dernier mouvement relatif et son amplitude sont conservés d'une mise sous tension à l'autre, de sorte qu'une commande GO sans paramètre reproduise toujours le dernier mouvement relatif réalisé même après remise sous tension.

Sur les modules MICROSIMPA les valeurs ne sont plus mémorisées d'une mise sous tension à l'autre, aussi une commande GO sans paramètre ne produit aucun mouvement à la suite d'un reset.

### III.3 – Evolution de la gestion des séquences

#### III.3.1 – Nombre de séquences et de phases

Modules SIMPA MICROPAS : 384 phases réparties à discrétion entre 100 séquences d'au plus 200 phases.

Modules MICROSIMPA : 1984 phases réparties entre 10 séquences de 128 phases et 22 séquences de 32 phases.

#### III.3.2 – Mémorisation des séquences

Sur les modules SIMPA MICROPAS, la sauvegarde des séquences est automatique, les phases des séquences peuvent être écrites ou modifiées n'importe quand du moment que la séquence a déjà été créée et non détruite. Par contre il est nécessaire de définir a priori le nombre de phases de la séquence au moment de la création dans la commande SN ns np.

Sur les modules MICROSIMPA, s'il n'est pas nécessaire de définir a priori le nombre de phases d'une séquence dans la commande d'ouverture, la description des phases doit être donnée juste après et la séquence doit être fermée pour la mémoriser au moyen de la commande SF. Il n'est pas possible de modifier une séquence qui a été fermée et mémorisée sans l'effacer, la recréer et la réécrire complètement.

#### III.3.3 – Ecriture des phases

Puisque les phases doivent être obligatoirement écrites entre la commande d'ouverture SN et la commande de fermeture SF, ces dernières ne comportent plus le numéro de séquence puisque celui ci est implicite dans les modules MICROSIMPA, d'où les deux formes d'écriture :

##### SIMPA MICROPAS

```
SE ns
SN ns np
SP ns 1 ....
SP ns 2 ....
...
SP ns np ....
```

##### MICROSIMPA

```
SE ns
SN ns
SP 1 ....
SP 2 ....
...
SP np ....
SF
```