Notice pour le contrôle
MAC23 V24 et MAC34 V24
en mode expert

Date : 14.12.06

Référence : mac23_34_ex_v5_um_fr.pdf
Réf. MI : CMND80697.DOC

Révision : 5

Auteur : C.MARTY

FICHE DE MODIFICATION DOCS MI 1/1

Documentation concernée : Notice pour le contrôle MAC23 V24 et MAC34 V24 en mode expert réf. : **CMND80697.DOC**

Date et demandeur de la (des) modification(s)	**Type** (corrective ou Evolutive) **et nature de la modification(s) :** (noter chapitres, paragraphes concernés)	**Approbation de la (des) modification(s)**	**Mise en place de la (des) modification(s)**	**Indice**
C. MARTY 09/05/00	**Création**			0
C. MARTY 12/2006	**Evolutive** - EVOLUTION de la relecture 21h, concerne les MAC en version logicielle 7.4 ou supérieure. - détail du calcul des vitesses de consigne - Ajout index 1Eh	Nom : C.MARTY Date : Oui ☒ Non ❑ motif du refus :	Personne chargée de la réalisation : N.ROUMEGOUX Date réalisation : 14.12.06	5

Midi Ingénierie se réserve le droit d'apporter, sans préavis, toute modification jugée opportune

SOMMAIRE

I – GÉNÉRALITÉS 1

II – PROTOCOLE POUR LES MESSAGES DE TYPE "COMMANDES" 1
II.1 – En émission 1
II.2 – En réception 1

III – PROTOCOLE POUR LES MESSAGES DE TYPE "REQUETES" 2
III.1 – En émission 2
III.2 – En réception 2

IV – TRAVAIL AVEC PLUSIEURS MODULES 2

V – MESSAGES POUR CONTROLE MAC23 ET MAC34 3
V.1 – Messages de type "Commandes" 3
V.2 – Messages de type "Requêtes" 4

VI – DETAIL DES COMMANDES 5

Midi Ingénierie se réserve le droit d'apporter, sans préavis, toute modification jugée opportune

I – GENERALITES

Les modules MAC23 ou MAC34 sont contrôlés par un ensemble de messages sous forme de "Commandes" et "Requêtes".

Les messages sont véhiculés sous forme de caractères ASCII au format 8 bits sans parité + 1 bit stop à la vitesse de 38400 bauds.

Le protocole utilise des trames de format fixe.

II – PROTOCOLE POUR LES MESSAGES DE TYPE "COMMANDES"

II.1 – En émission

Stx 014 aa ii ss pppppppp cc Etx

avec	aa	2 cars	Hexa	codés ASCII	représentant l'adresse du module de 00 à 7F
	ii	2 cars	Hexa	codés ASCII	représentant la partie Index de la commande
	ss	2 cars	Hexa	codés ASCII	représentant la partie Sous-index de la commande
	pp . . . pp	8 cars	Hexa	codés ASCII	représentant la partie Paramètres de la commande
	cc	2 cars	Hexa	codés ASCII	représentant la somme modulo 100h des 14 caractères ASCII a, i, s et p

II.2 – En réception

Ack	Xoff	Xon	si message correctement interprété et action exécutée
Nack			si checksum (cc) incorrect
Ack	Xoff	Xonerreur	si trame correcte mais commande refusée par le module

Caractères ASCII de contrôle

Stx	02h
Etx	03h
Ack	06h
Nack	15h
Xon	1Ah
Xoff	13h
Xonerreur	17h
0	30h
1	31h
4	34h

III – PROTOCOLE POUR LES MESSAGES DE TYPE "REQUETES"

III.1 – En émission

Stx 014 aa $i_1 i_2$ ss dddddddd cc Etx

avec	aa	2 cars	Hexa	codés ASCII	représentant l'adresse du module de 00 à 7F
	ii	2 cars	Hexa	codés ASCII	représentant la partie Index de la commande
	ss	2 cars	Hexa	codés ASCII	représentant la partie Sous-index de la commande
	dd . . . dd	8 cars	Hexa	codés ASCII	représentant la partie Paramètres de la commande
	cc	2 cars	Hexa	codés ASCII	représentant la somme modulo 100h des caractères ASCII a, i, s et d

III.2 – En réception

Ack Xoff Stx 011 aa i_2 dddddddd cc Etx Xon

avec	aa	2 cars	Hexa	codés ASCII	identiques aux caractères aa du message de commande
	i_2	1 car	Hexa	codé ASCII	identique au car i_2 du message de commande
	dd . . . dd	8 cars	Hexa	codés ASCII	représentant la partie Réponse du message de requête
	cc	2 cars	Hexa	codés ASCII	représentant la somme modulo 100h des caractères ASCII a, i_2 et d

IV – TRAVAIL AVEC PLUSIEURS MODULES

- **Configuration des adresses**

  Il est nécessaire de connecter un seul module à la fois pour configurer son adresse en utilisant la commande d'index 08h.

- **Exploitation multi-axes**

  Il est possible de paralléliser plusieurs modules MAC d'adresses distinctes sur un même port série.

  Attention, un des modules doit se trouver à l'adresse 00.

  Chacun des axes est alors pilotable séparément.

  Pour envoyer une commande destinée à tous les modules (commande générale), il suffit de donner la valeur F aux caractères d'adresse aa (aa = FF soit 46h 46h).

V – MESSAGES POUR CONTROLE MAC23 ET MAC34

"!" : commandes interdites en cours de mouvement.

Le MAC23/34 peut interpréter 32 commandes et 16 relectures. Les codes hexa sont répartis comme suit: 00h à 1Fh pour les commandes (32 commandes possibles) et 20h à 2Fh pour les relectures (16 relectures).
La liste ci-dessous résume les commandes utilisées.

V.1 – Messages de type "Commandes"

- **Initialisation**

Index	Sous-index	Paramètres	Fonction
00h	00h	x x x x x x x	réinitialisation du module
01h	00h	x x x x x x x	réinitialisation de l'EEPROM (retour en configuration usine)

- **Paramétrage**

Index	Sous-index	Paramètres	Fonction	
02h	00h	x x x x x x00	forçage du courant nominal à l'arrêt du moteur	!
02h	00h	x x x x x x01	mise en standby automatique à l'arrêt moteur (Couple moitié)	!
03h	00h	x x x x x x00	butées soft et hard inactives	!
03h	00h	x x x x x x01	autorisation des butées hard	!
03h	00h	x x x x x x08	autorisation des butées soft	!
03h	00h	x x x x x x09	autorisation des butées soft et hard	!
04h			non utilisée	
05h			non utilisée	
06h			non utilisée	
07h	00h	x x x x x x01	autorisation de l'entrée Référence	
07h	00h	x x x x x x00	inhibition de l'entrée Référence	
08h	00h	x x x x x xAA	nouvelle adresse du module = AA	!
09h	00h	BBBB BBBB	définit la butée soft supérieure	!
0Ah	00h	BBBB BBBB	définit la butée soft inférieure	!
0Bh	00h	x x x x x x x	RAZ compteur de position (position origine)	
0Ch	00h	x x x x x xCC	couple CC avec 00h<CC<80h	
0Dh	00h	x x x x VVVV	vitesse max définie en période (VVVVh * 0.125µs par incrément)	!
0Eh	00h	x x x x VV00	vitesse de rattrapage	
0Fh	00h	x x x x x xNN	coefficient de rampe NNh	!

- **Mouvement**

Index	Sous-index	Paramètres	Fonction
10h	00h	x x x x x x x x	attente synchro: le prochain mouvement sera synchronisé
11h	00h	x x x x x x x x	signal de synchronisation
12h	00h	x x x x x x x x	non utilisée
13h	00h	PPPP PPPP	aller à la position PPPPPPPPh suivant le profil de vitesse !
14h	00h	x x x x x x x x	retour à la position 0 suivant le profil de vitesse !
15h	00h	xx 00 VVVV	mouvement infini avec consigne en vitesse VVVVh, sens+
15h	00h	xx F6 VVVV	mouvement infini avec consigne en vitesse VVVVh, sens-
16h	00h	DDDDVVVV	segment: déplacement relatif DDDDh avec la vitesse VVVVh
17h	00h	x x x x x x x x	arrêt avec rampe prédéfinie
18h	00h	x x x x x x x x	arrêt sans rampe (freinage maximum)
19h	00h	x x x x x x x x	mise sous puissance (automatique si mouvement)
1Ah	00h	x x x x x x x x	coupure puissance moteur
1Bh			non utilisé
1Ch			non utilisé
1Dh			non utilisé
1Eh	00h	DDDD VVVV	segment "virtuel" : ie index 16h mais mouvement non exécuté
1Fh			non utilisé

V.2 – Messages de type "Requêtes"

Index	Sous-index	Paramètres	Fonction
20h	00h	x x x x x x x x	relecture de la position en cours
21h	00h	x x x x x x x x	relecture de l'état du module
22h	00h	x x x x x x x x	relecture de la butée soft supérieure
23h	00h	x x x x x x x x	relecture de la butée soft inférieure
24h	00h	x x x x x x x x	relecture de la version logicielle
25h	00h	x x x x x x x x	relecture des vitesses maximum et rattrapage
26h	00h	x x x x x x x x	lecture coeff rampe + couple + mode(courant/butées)
27h	00h	x x x x x x x x	relecture température + tension d'alim + E/S
28h	SSh	x x x x x x x x	relecture 4 octets RAM à partir de l'adresse SSh
29h	SSh	x x x x x x x x	relecture 4 octets EEPROM à partir de l'adresse SSh

2Ah à 2Fh : non utilisées

VI – DETAIL DES COMMANDES

Index Commande: **00h**

Type de fonction: Initialisation

Fonction: Initialisation du module

Commande: Stx 0 1 4 a a **0 0** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Le module est remis dans un état identique à l'état de mise sous tension.

Attention !Attendre 2 secondes au minimum après l'envoi de cette commande. Tout message envoyé pendant ce délai ne serait pas pris en compte.

Index Commande: **01h**

Type de fonction: Initialisation

Fonction: Initialisation EEPROM

Commande: Stx 0 1 4 a a **0 1** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Tous les paramètres du module sont remplacés par les paramètres usine. Attention, l'adresse est réinitialisée à 00 et le baudrate à 38400 bauds.

Attention ! Attendre 2 secondes au minimum après l'envoi de cette commande. Tout message envoyé pendant ce délai ne serait pas pris en compte.

Index Commande: **02h**

Type de fonction: Paramétrage

Fonction: Mode de gestion du couple à l'arrêt

Commande: Stx 0 1 4 a a **0 2** 0 0 x x x x x x 0 d C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: d
0 Forçage du courant nominal à l'arrêt du moteur
1 Mise en standby automatique à l'arrêt moteur (Couple moitié)
Le mode nominal fournit un couple de maintien important alors que le mode standby permet de diminuer la consommation sur l'alimentation et l'échauffement du moteur.

Index Commande: **03h**

Type de fonction: Paramétrage

Fonction: Mode de gestion des butées

Commande: Stx 0 1 4 a a **0 3** 0 0 x x x x x x 0 d C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: d
0 Butées soft et hard inactives
1 Autorisation des butées hard seulement
8 Autorisation des butées soft seulement
9 Autorisation des butées soft et hard

Index Commande: **07h**

Type de fonction: Paramétrage

Fonction: Gestion de l'entrée Référence

Commande: Stx 0 1 4 a a **0 7** 0 0 x x x x x x O d C_1 C_2 Etx

Action: d = 1 Pour autoriser l'entrée Référence (RAZ position sur un front)
d = 0 Pour inhiber l'entrée Référence. C'est la configuration par défaut à chaque mise sous tension du module.

Index Commande: **08h**

Type de fonction: Configuration

Fonction: Programmation de l'adresse du module

Commande: Stx 0 1 4 F F **0 8** 0 0 x x x x x x a a C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Transmet la nouvelle adresse d'un module seul présent sur la ligne.

Index Commande: **09h**

Type de fonction: Paramétrage

Fonction: Définit la butée soft supérieure

Commande: Stx 0 1 4 a a **0 9** 0 0 b b b b b b b b C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Programme la valeur de la butée soft supérieure avec la valeur hexadécimale signée bbbbbbbbh.
Cette valeur est mémorisée en cas de coupure d'alimentation.
Si la position de l'axe dépasse cette butée, le mouvement est immédiatement stoppé (freinage maximum) pourvu que la détection des butées soft soit autorisée.

Index Commande: **0Ah**

Type de fonction: Paramétrage

Fonction: Définit la butée soft inférieure

Commande: Stx 0 1 4 a a **0 A** 0 0 b b b b b b b b C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Programme la valeur de la butée soft inférieure avec la valeur hexadécimale signée bbbbbbbbh. Cette donnée est mémorisée en cas de coupure d'alimentation.
Si la position de l'axe dépasse cette butée, le mouvement est immédiatement stoppé (freinage maximum) pourvu que la détection des butées soft soit autorisée.

Index Commande: **0Bh**

Type de fonction: Paramétrage

Fonction: Raz position

Commande: Stx 0 1 4 a a **0 B** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Remise à zéro du compteur de position de l'axe. Cette commande permet donc de redéfinir la position d'origine.

Index Commande: **0Ch**

Type de fonction: Paramétrage

Fonction: Limite le couple maximum de l'axe.

Commande: Stx 0 1 4 a a **0 C** 0 0 x x x x x x d d C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Cette commande permet de limiter le couple fourni par le moteur. La valeur hexadécimale "dd" doit être comprise entre 00h et Imax (cf requête 28h). Le couple maximum sera proportionnel à cette valeur et au couple maximum de l'axe.
Cette donnée est mémorisée en cas de coupure d'alimentation.

Index Commande: **0Dh**

Type de fonction: Paramétrage

Fonction: Paramétrage de la vitesse haute du module

Commande: Stx 0 1 4 a a **0 D** 0 0 x x x x v v v v C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Modification de la vitesse haute du module. Celle-ci est donnée par 2 octets hexadécimaux sous forme de période : la durée d'un incrément vaut v v v v * 0,125 µs.

MAC23: vitesse (tours/min) = 240000 / valeur décimale de VVVV
MAC34: vitesse (tours/min) = 48000 / valeur décimale de VVVV

Ce paramètre est conservé en mémoire lors des coupures d'alimentation.

Index Commande: **0Eh**

Type de fonction: Paramétrage

Fonction: Fixe la vitesse de rattrapage

Commande: Stx 0 1 4 a a **0 E** 0 0 x x x x V V 0 0 C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: La valeur hexadécimale V V h donne la vitesse utilisée pour ajuster la position réelle à la position de consigne lors d'un mouvement point à point (index 13h et 14h).

La vitesse est telle que chaque incrément durera V V * 200µs.

Ce paramètre est conservé en mémoire lors des coupures d'alimentation.

Index Commande: **0Fh**

Type de fonction: Paramétrage

Fonction: Ajustement de l'accélération

Commande: Stx 0 1 4 a a **0 F** 0 0 x x x x x x A A C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Définit le nombre d'incréments effectués sur les rampes d'accélération et de décélération lors des mouvements qui suivent le profil de vitesse trapézoïdal (mouvements d'index 13h,14h,15h).

Le nombre d'incréments effectués sur chaque rampe vaut 120 fois AA.
Ce paramètre est mémorisé en cas de coupure d'alimentation.

Index Commande: **10h**

Type de fonction: Mouvement

Fonction: Passage en mode synchrone

Commande: Stx 0 1 4 a a **1 0** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Signale que le prochain mouvement sera synchronisé (ie lancé par la commande d'index 11h).

Index Commande: **11h**

Type de fonction: Mouvement

Fonction: Signal de synchronisation

Commande: Stx 0 1 4 a a **1 1** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Lance le mouvement préchargé. Il faut pour cela que le module ait séquentiellement reçu la commande 10h suivie d'une commande de mouvement acceptant le mode synchrone (13h,14h,15h) avant d'avoir cette synchronisation.

Index Commande: **13h**

Type de fonction: Mouvement

Fonction: Mouvement vers la position donnée avec profil de vitesse trapézoïdal

Commande: Stx 0 1 4 a a **1 3** 0 0 p p p p p p p p C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Cette commande lance un mouvement vers la position absolue hexadécimale pppppppph (valeur signée). La vitesse suit un profil trapézoïdal:

- accélération jusqu'à la vitesse maximum (cf commande 0Dh) sur un nombre d'incréments donné (cf commande 0Fh),
- palier à vitesse maximum si le déplacement global est suffisamment important,
- décélération suivant une rampe symétrique à celle de l'accélération,
- ajustement de la position à la valeur de consigne avec une vitesse bornée par la vitesse de rattrapage.

Index Commande: **14h**

Type de fonction: Mouvement

Fonction: Retour vers la position zéro avec un profil de vitesse trapézoïdal

Commande: Stx 0 1 4 a a **1 4** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Retour à la position d'origine (ie commande 13h avec argument 00000000h).

Index Commande: **15h**

Type de fonction: Mouvement

Fonction: Demande un mouvement "infini" de vitesse donnée

Commande: Stx 0 1 4 a a **1 5** 0 0 x x S S V V V V C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Lance ou enchaîne un mouvement de vitesse V V V V. Le premier mouvement de ce type lance le moteur dans le sens anti-horaire si SS=F6h ou horaire dans les autres cas. Le mouvement est accéléré en suivant la rampe prédéfinie jusqu'à la vitesse de consigne. Celle-ci est donnée par les 2 octets hexadécimaux V V V V sous forme de période: la durée d'un incrément vaut V V V V * 0.125µs.

MAC23: vitesse (tours/min) = 240000 / valeur décimale de VVVV
MAC34: vitesse (tours/min) = 48000 / valeur décimale de VVVV

Lors de ce type de mouvement, la vitesse peut être modifiée en cours de déplacement par une nouvelle commande 15h mais le sens de rotation est forcément conservé. Ce mouvement peut être stoppé par les commande 17h,18h ou bien par cette commande 15h avec un argument VVVV=0000.

Index Commande: **16h**

Type de fonction: Mouvement

Fonction: Mouvement interpolé

Commande: Stx 0 1 4 a a **1 6** 0 0 p p p p v v v v C_1 C_2 Etx

aa adresse hexa du module (2 caractères codés Ascii)
pppp paramètre hexadécimal de position sous forme d'un nombre de pas à effectuer (4 cars codés Ascii)
vvvv paramètre de vitesse hexadécimal sous forme d'une période d'unité 0,125 µs / pas (4 cars codés Ascii)
C_1 C_2 checksum module (2 caractères codés Ascii)

Action: Effectue un mouvement à vitesse constante suivant le paramètre vitesse pour un nombre de pas selon le paramètre position.

Remarques Cette commande de mouvement est stockée par le module dans une mémoire tampon à 3 étages (FiFo) avant interprétation.
Ce système permet un fonctionnement de type "mouvement interpolé".
Lorsque la FiFo est saturée, une nouvelle commande est rejetée (retour XONERROR) jusqu'à désaturation.
Utiliser l'index 1Eh pour réaliser un arrêt moteur temporisé.

Index Commande: **17h**

Type de fonction: Mouvement

Fonction: Arrêt avec décélération prédéfinie

Commande: Stx 0 1 4 a a **1 7** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Effectue un freinage en limitant la décélération avec le coefficient de rampe prédéfini. Le moteur s'immobilise puis reste sous tension pour conserver un couple de maintien en position.
Cette commande peut interrompre les mouvements 13h,14h et15h.

Index Commande: **18h**

Type de fonction: Mouvement

Fonction: Arrêt d'urgence (freinage maximum)

Commande: Stx 0 1 4 a a **1 8** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Effectue un arrêt d'urgence. Le moteur freine avec son couple maximum et s'immobilise. Il reste sous tension une fois arrêté. La position de l'axe reste valide mais la boucle d'asservissement de position est stoppée. Cette commande peut interrompre n'importe quel type de mouvement.

Index Commande: **19h**

Type de fonction: Mouvement

Fonction: Mise sous puissance

Commande: Stx 0 1 4 a a **1 9** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Le moteur est mis sous tension et fournit un couple de maintien limité par la valeur choisie (cf commande 0Ch) et divisé par 2 si le mode standby est activé. Si nécessaire, cette mise sous tension est effectuée automatiquement lors de l'exécution d'une commande de mouvement.

Index Commande: **1Ah**

Type de fonction: Mouvement

Fonction: Coupure de puissance moteur

Commande: Stx 0 1 4 a a **1 A** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: Coupe la puissance moteur. Noter que le moteur n'a plus de couple de maintien.

Attention: il peut être risqué pour l'alimentation de couper la puissance en cours de rotation. D'autre part, cette commande ne donne pas de couple de freinage efficace. Il faut utiliser la commande 18h pour avoir un arrêt d'urgence.

Index Commande: **1Bh**

Type de fonction: Paramétrage

Fonction: Réglage du baudrate

Commande: Stx 0 1 4 a a **1 B** 0 0 x x x x x x v v C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)
vv 01 pour 9600 bauds
02 pour 19200 bauds
03 pour 38400 bauds

Action: Modifie le baudrate du module et réinitialise le module. Le baudrate est ensuite conservé même en cas de coupure d'alimentation. La valeur usine est 38400 bauds.

Index Commande: **1Ch**

Type de fonction: Paramétrage

Fonction: Réglage de la tolérance en mode position

Commande: Stx 0 1 4 a a **1 C** 0 0 x x x x x x T T C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Action: La valeur hexadécimale TTh donne le nombre d'incréments tolérés entre position réelle et position de consigne.

Index Commande: **1Eh**

Type de fonction: Mouvement

Fonction: Mouvement interpolé, segment virtuel

Commande: Stx 0 1 4 a a **1 E** 0 0 p p p p v v v v C_1 C_2 Etx

aa adresse hexa du module (2 caractères codés Ascii)
pppp paramètre hexadécimal de position sous forme d'un nombre de pas à effectuer (4 caractères codés Ascii)
vvvv paramètre de vitesse hexadécimal sous forme d'une période d'unité 0,125 µs / pas (4 caractères codés Ascii)
C_1 C_2 checksum module (2 caractères codés Ascii)

Action: Donne un arrêt moteur dont la durée est strictement celle du segment de pppp pas à la vitesse VVVV.
Cette commande permet de gérer les phases d'arrêt les phases d'arrêt sans perte de synchronisation dans une interpolation multiaxe.
Par exemple :
…
Stx 0 1 4 0 0 1 6 0 0 0 3 E 8 1 0 0 0 C_1 C_2 Etx (segment axe 0)
Stx 0 1 4 0 1 1 E 0 0 0 3 E 8 1 0 0 0 C_1 C_2 Etx
(arrêt axe 1 pendant le segment axe 0)

Remarques Cette commande de mouvement est stockée par le module dans une mémoire tampon à 3 étages (FiFo) avant interprétation.
Ce système permet un fonctionnement de type "mouvement interpolé".
Lorsque la FiFo est saturée, une nouvelle commande est rejetée (retour XONERROR) jusqu'à désaturation.

Index Commande: **20h**

Type de fonction: Requête

Fonction: Lecture de la position actuelle

Requête: Stx 0 1 4 a a **2 0** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Réponse: Stx 0 1 1 a a **0** p p p p p p p p C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

La valeur sur 4octets hexadécimaux signés pp pp pp pp donne la position en cours de l'axe interrogé. Cette position est mémorisée en cas de coupure d'alimentation. A la remise sous tension, elle est donc valide pourvu que l'axe n'ait pas été déplacé.

Index Commande: **21h**

Type de fonction: Requête

Fonction: Lecture de l'état du module

Requête: Stx 0 1 4 a a **2 1** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Réponse: Stx 0 1 1 a a **1** d d f f g g h h C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

1er octet (dd) : défauts rédhibitoires.
Ces défauts sont mémorisés lorsqu'ils apparaissent. Ils sont acquittés par réinitialisation (index 00h ou remise sous tension du module).
Les disjonctions coupent la puissance moteur.
Toutes les commandes de mouvement sont alors refusées jusqu'à acquittement du défaut.

dd: valeur hexadécimale représentant bit à bit les éventuels défauts relevés.
bit 7: réservé
bit 6: avertissement non mémorisé de surtension d'alimentation
bit 5: avertissement non mémorisé d'alimentation insuffisante
bit 4: disjonction pour surchauffe MAC23
bit 3 : disjonction amplificateur MAC34 (surcourant, fuite à la masse, surchauffe, surtension alim > 125% Vmax)
bit 2 : disjonction pour surchauffe moteur MAC34
bit 1 : disjonction pour absence de codeur MAC34
bit 0 : disjonction pour micro coupure d'alimentation

f f: valeur hexadécimale représentant bit à bit l'état du module.
bit 7: réservé
bit 6: réservé
bit 5: réservé
bit 4: réservé
bit 3: arrêt sur la butée hard positive
bit 2: arrêt sur la butée hard négative
bit 1: arrêt sur la butée soft positive
bit 0: arrêt sur la butée soft négative

g g: valeur hexadécimale représentant bit à bit l'état du module.
bit 7: réservé
bit 6: réservé
bit 5: réservé
bit 4: réservé
bit 3: réservé
bit 2: réservé
bit 1: réservé
bit 0: réservé

h h: valeur hexadécimale représentant bit à bit l'état du module.
bit 7: avertissement mémorisé de surtension d'alimentation, implémenté sur les MAC en version logicielle 7.4 ou supérieure.
Ce bit est remis à zéro par la commande d'index 00h ou bien lors d'une coupure d'alimentation.
Il peut être activé par deux phénomènes distincts: surtension d'alimentation externe ou bien freinage important (transformation d'énergie mécanique en énergie électrique).
Dans ce dernier cas veuillez nous contacter afin de mettre en place si nécessaire un ballast ou une autre récupération d'énergie.
Note: ce bit peut également être activé sur un système doté d'un ballast car le seuil de déclenchement du ballast peut être supérieur à la tension d'avertissement.
bit 6: réservé
bit 5: réservé
bit 4: réservé
bit 3: réservé
bit 2: réservé
bit 1: réservé
bit 0: réservé

Index Commande: **22h**

Type de fonction: Requête

Fonction: Lecture de la butée soft supérieure

Requête: Stx 0 1 4 a a **2 2** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Réponse: Stx 0 1 1 a a **2** p p p p p p p p C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

La valeur sur 4octets hexadécimaux signés pp pp pp pp donne la position de la butée soft supérieure de l'axe. Cette position est mémorisée en cas de coupure d'alimentation.

Index Commande: **23h**

Type de fonction: Requête

Fonction: Lecture de la butée soft inférieure

Requête: Stx 0 1 4 a a **2 3** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Réponse: Stx 0 1 1 a a **3** p p p p p p p p C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

La valeur sur 4octets hexadécimaux signés pp pp pp pp donne la position de la butée soft inférieure de l'axe. Cette position est mémorisée en cas de coupure d'alimentation.

Index Commande: **24h**

Type de fonction: Requête

Fonction: Relecture de la version logicielle

Requête: Stx 0 1 4 a a **2 4** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Réponse: Stx 0 1 1 a a **4** V V V V V V V V C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

V V V V V V V V donne la version et l'indice logiciel.

Index Commande: **25h**

Type de fonction: Requête

Fonction: Lecture des vitesses de consigne

Requête: Stx 0 1 4 a a **2 5** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Réponse: Stx 0 1 1 a a **5** r r 0 0 v v v v C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

r r 0 0 correspond à la vitesse de rattrapage (cf index Eh)
v v v v correspond à la vitesse de consigne (cf index 0Dh).

Index Commande: **26h**

Type de fonction: Requête

Fonction: Lecture des coefficients de rampe et de couple

Requête: Stx 0 1 4 a a **2 6** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Réponse: Stx 0 1 1 a a **6** n n c c K m d d C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

n n coefficient de rampe hexadécimal, valeur de 00h à FFh
La rampe d'accélération ou de décélération s'effectue sur un nombre d'incréments égal à 120 * n n .

c c coefficient de couple hexadécimal, valeur de 00h à 80h.
Le couple moteur est limité par la valeur Cm = Cmot * c c / Imax avec Cmot couple maximum du moteur et Imax valeur accessible par la commande d'index 28h.

k mode de gestion du couple à l'arrêt:
0 Forçage du courant nominal à l'arrêt du moteur
1 Mise en standby automatique à l'arrêt moteur (Couple moitié)

m gestion des butées:
0 Butées soft et hard inactives
1 Autorisation des butées hard seulement
8 Autorisation des butées soft seulement
9 Autorisation des butées soft et hard

dd Tolérance en mode position

Index Commande: **27h**

Type de fonction: Requête

Fonction: Lecture des mesures analogiques

Requête: Stx 0 1 4 a a **2 7** 0 0 x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

Réponse: Stx 0 1 1 a a **7** t t v v c c d d C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

t t valeur hexadécimale donnant une mesure de la température:

Dans le cas d'un MAC23, dans la gamme [0,100℃] on a :
Température (℃) = 2,5 x mesure – 215 (mesure = val eur décimale de tt)

Dans le cas d'un MAC34-1 ou MAC34-2, on a :
Température (℃) = 2,45 x mesure – 140

v v valeur hexadécimale donnant une mesure de la tension d'alimentation:
MAC23 : Valim (V) = mesure/4,8 (mesure = valeur décimale de v v)
MAC34-1 : Valim (V) = mesure x 0,4
MAC34-2 : Valim (V) = mesure x 0,8

c c 1 octet hexadécimal représentant les valeurs d'entrées/sorties
bit 7: réservé
bit 6: réservé
bit 5: entrée butée positive (1 ssi l'entrée est activée)
bit 4: réservé
bit 3: "busy" : valeur 0 ssi le moteur ne parvient pas à suivre la consigne de vitesse du mouvement
bit 2: réservé
bit 1: réservé
bit 0: réservé

d d 1 octet hexadécimal représentant les valeurs d'entrées/sorties
bit 7: réservé
bit 6: entrée butée négative (1 ssi l'entrée est activée)
bit 5: entrée référence (1 ssi l'entrée est activée)
bit 4: réservé
bit 3: réservé
bit 2: réservé
bit 1: réservé
bit 0: réservé

Remarque: cette relecture permet de vérifier votre câblage des entrées butées et du top référence.

Index Commande: **28h**

Type de fonction: Requête

Fonction: Relecture RAM

Requête: Stx 0 1 4 a a **2 8** @ @ x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)
@ @ adresse du premier octet à relire

Réponse: Stx 0 1 4 a a **8** p p p p p p p p C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

pp pp pp pp sont les 4 octets RAM lus à partir de l'adresse @ @.

Exemple: Lecture du coefficient Imax
Ce coefficient est placé à l'adresse AFh
Requête Stx 0 1 4 a a **2 8** A F x x x x x x x x C_1 C_2 Etx
Réponse Stx 0 1 4 a a **8** v v x x x x x x C_1 C_2 Etx
La valeur vv donne le coefficient Imax en hexadécimal

Index Commande: **29h**

Type de fonction: Requête

Fonction: Relecture de 4 octets EEPROM

Requête: Stx 0 1 4 a a **2 9** @ @ x x x x x x x x C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)
@ @ adresse du premier octet à relire

Réponse: Stx 0 1 1 a a **9** p p p p p p p p C_1 C_2 Etx

aa adresse du module (2 caractères codés Ascii)
C_1 C_2 checksum du message (2 caractères codés Ascii)

pp pp pp pp sont les 4 octets EEPROM lus à partir de l'adresse @ @.

EXEMPLE D'APPLICATION DU PROTOCOLE EXPERT MAC

Mouvement infini à 300 tours/min dans le sens horaire pour le MAC23 d'adresse 00.

300tr/min ⇔ 5tr/s, soit 10000inc/s (MAC23: 2000inc/tour). Un incrément dure donc 10^{-4} s soit 800 x 0.125µs. D'où le paramètre **VVVV** = hexa(800) = **0320h** sur 2octets.
Sens de rotation : horaire si SS=00h, anti-horaire si SS=F6h.
La trame de commande se constitue donc de la manière suivante :

Stx	longueur		a	a	i	i	s	s	x	x	S	S	V	V	V	V	c	c	Etx
Stx	0	1	4	0	0	1	5	0	0	0	0	0	0	0	3	2	0	c	c	Etx

02h 30h 31h 34h 30h 30h 31h 35h 30h 30h 30h 30h 30h 30h 30h 33h 32h 30h 41h 42h 03h

CHECKSUM ?

1) Calcul de la somme hexadécimale:
   30h+30h+31h+35h+30h+30h+30h+30h+30h+30h+30h+33h+32h+30h = 2ABh

2) Résultat modulo 100h: ABh

3) Le checksum est donc ABh, pour "c c" il faut envoyer les caractères "A" and "B" : 41h and 42h

Note : vous pouvez valider vos conversions de paramètres grâce à MACSIM. Soit par exemple le message Stx 0 1 4 0 0 1 5 0 0 0 0 0 0 0 3 2 0 c c Etx

⇨ Supprimer "stx 0 1 4" et "c c etx" et envoyer directement sous MACSIM le message "00150000000320".