MAC23 V24 / MAC34 V24
Expert mode manual

Date: 14-Dec-06

Référence : mac23_34_ex_v5_um_en.pdf
Réf. MI : NSND80812.DOC

Release: 5

Author: CM / NS

TABLE OF CONTENT

I – GENERALITIES

MAC23 or MAC34 modules are controlled by "Commands" and "Queries" messages.

Warning: MAC23 and MAC34 have different motor resolutions:
MAC23 → 2000 increments per revolution
MAC34 → 10000 increments per revolution

Messages are ASCII characters, 8 bits, no parity, 1 stop bit, 38400 bauds.

The protocol uses fixed format data frames.

II – "COMMANDS" MESSAGES PROTOCOL

III.1 – Emission

Stx 014 aa ii ss pppppppp cc Etx

with	aa	2 ASCII-encoded Hex chars	Standing for the module address from 00 to 7F
	ii	2 ASCII-encoded Hex chars	Standing for the index part of the command
	ss	2 ASCII-encoded Hex chars	Standing for the sub-index part of the command
	pp…pp	8 ASCII-encoded Hex chars	Standing for the parameters part of the command
	cc	2 ASCII-encoded Hex chars	Standing for the sum modulus 100h of the 14 previous ASCII characters aa, ii, ss and pp…pp

III.2 – Reception

Ack	Xoff	Xon	If message correctly interpreted and action executed.
Nack			If wrong checksum (cc).
Ack	Xoff	Xonerror	If frame is correct but command is refused by the module.

Note:

ASCII control characters	Hexadecimal value
Stx	02h
Etx	03h
Ack	06h
Nack	15h
Xon	1Ah
Xoff	13h
Xonerror	17h
0	30h
1	31h
4	34h

III – "QUERIES" MESSAGES PROTOCOL

III.1 – Emission

Stx 014 aa i_1i_2 ss ddddddddd cc Etx

with	aa	2 ASCII-encoded Hex chars	Standing for the module address from 00 to 7F
	ii	2 ASCII-encoded Hex chars	Standing for the index part of the command
	ss	2 ASCII-encoded Hex chars	Standing for the sub-index part of the command
	dd…dd	8 ASCII-encoded Hex chars	Standing for the parameters part of the command
	cc	2 ASCII-encoded Hex chars	Standing for the sum modulus 100h of the 14 previous ASCII characters aa, ii, ss and dd…dd

III.2 – Reception

Ack Xoff Stx 011 aa i_2 ddddddddd cc Etx Xon

with	aa	2 ASCII-encoded Hex chars	Similar to command message aa characters
	I_2	1 ASCII-encoded Hex chars	Similar to command message I_2 character
	dd…dd	8 ASCII-encoded Hex chars	Standing for the Answer part of the query message
	cc	2 ASCII-encoded Hex chars	Standing for the sum modulus 100h of the ASCII characters aa, i_2 and dd…dd

IV – WORKING WITH SEVERAL MODULES

- **Addresses configuration**

  It is necessary to connect only one module at a time to configure its address using the 08h command index.

- **Multi-Axis operation**

  It is possible to parallelize several MAC modules with distinct addresses on the same serial port.

  Warning, one of the modules must be located at address 00.

  Each axis is then separately drivable.

  To send a command to all the modules (general command), one has to specify FF as aa address characters (aa = FF = 4646h)

V – MAC23 AND MAC34 CONTROL MESSAGES

⚠ : command is forbidden while axis is running.

The MAC23/34 can interpret 32 commands and 16 readings. Hex codes are split as follows:
00h to 1Fh for commands (32 possible commands)
20h to 2Fh for readings (16 readings)
The list here after summarizes commands.

V.1 –"Commands" messages

- **Initialization**

Index	Sub-index	Parameters	Function
00h	00h	x x x x x x x x	re-initialization of the module
01h	00h	x x x x x x x x	re-initialization of the EEPROM (back to factory configuration)

- **Parameterization**

Index	Sub-index	Parameters	Function	
02h	00h	x x x x x x 0 0	forcing nominal current when motor stopped	⚠
02h	00h	x x x x x x 0 1	Automatic standby when motor stopped (Half torque)	⚠
03h	00h	x x x x x x 0 0	Hardware and Software stops inactive	⚠
03h	00h	x x x x x x 0 1	Hardware stops enable	⚠
03h	00h	x x x x x x 0 8	Software stops enable	⚠
03h	00h	x x x x x x 0 9	Hardware and Software stops enable	⚠
04h			not used	
05h			not used	
06h			not used	
07h	00h	x x x x x x 0 1	Reference input enable	
07h	00h	x x x x x x 0 0	Reference input disable	
08h	00h	x x x x x x A A	New module address = AA	⚠
09h	00h	BBBB BBBB	Software superior stop definition	⚠
0Ah	00h	BBBB BBBB	Software inferior stop definition	⚠
0Bh	00h	x x x x x x x x	Position Counter reset (origin position)	
0Ch	00h	x x x x x x C C	Torque CC with 00h<CC<80h	
0Dh	00h	x x x x V V V V	Maximum Speed defined as a period (VVVVh * 0.125µs per increment)	
0Eh	00h	x x x x V V 0 0	Unstall Speed	
0Fh	00h	x x x x x x N N	Ramp coefficient NNh	⚠

- **Motion**

Index	Sub-index	Parameters	Function	
10h	00h	x x x x x x x x	Wait Synchro: the next motion will be synchronized	
11h	00h	x x x x x x x x	Synchronization signal	
12h	00h	x x x x x x x x	Not used	
13h	00h	PPPP PPPP	Go to position PPPPPPPPh following the velocity profile	⚠
14h	00h	x x x x x x x x	Go back to position 0 following the velocity profile	⚠
15h	00h	x x 0 0 V V V V	infinite motion with speed set-point VVVVh, direction+	
15h	00h	x x F 6 V V V V	infinite motion with speed set-point VVVVh, direction+	
16h	00h	DDDDVVVV	segment: relative move DDDDh at VVVVh speed	
17h	00h	x x x x x x x x	stop with predefined ramp	
18h	00h	x x x x x x x x	stop without ramp (maximum braking)	
19h	00h	x x x x x x x x	Power On (automatic if motion)	
1Ah	00h	x x x x x x x x	Motor Power Off	
1Bh			not used	
1Ch			not used	
1Dh			not used	
1Eh	00h	DDDD VVVV	Virtual segment (temporization)	
1Fh			not used	

V.2 –"Queries" Messages

Index	Sub-index	Parameters	Function	
20h	00h	x x x x x x x x	Reading of the current position	
21h	00h	x x x x x x x x	Reading of the module status	
22h	00h	x x x x x x x x	Reading of the superior end-stop	
23h	00h	x x x x x x x x	Reading of the inferior end-stop	
24h	00h	x x x x x x x x	Reading of the software release	
25h	00h	x x x x x x x x	Reading of Maximal and unstall speeds	
26h	00h	x x x x x x x x	Reading of ramp coefficients + Torque + Mode (current/stops)	
27h	00h	x x x x x x x x	Reading of Temperature + Supply voltage + I/Os	
28h	SSh	x x x x x x x x	Reading of 4 bytes in RAM starting at address SSh	
29h	SSh	x x x x x x x x	Reading of 4 bytes in EEPROM starting at address SSh	
2Ah			not used	
2Bh			not used	
2Ch			not used	
2Dh			not used	
2Eh			not used	
2Fh			not used	

VI – DETAILED COMMANDS

Command Index	**00h**
Function type	Initialization
Function	Module Initialization
Command	Stx 0 1 4 a a **0 0** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	The module is set back to a status similar to the startup status. ***Warning! Wait for at least 2 seconds after having sent this command. Every message sent before that delay will not be taken into account.***

Command Index	**01h**
Function type	Initialization
Function	EEPROM Initialization
Command	Stx 0 1 4 a a **0 1** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	All the parameters of the module are replaced by factory parameters. Warning: the module address is reset to 00 and the baudrate is reset to 38400 bauds. ***Warning! Wait for at least 2 seconds after having sent this command. Every message sent before that delay will not be taken into account.***

Command Index	**02h**
Function type	Parameterization
Function	Stop Torque management mode.
Command	Stx 0 1 4 a a **0 2** 0 0 x x x x x x 0 d C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	d 0 Nominal current is forced when motor stops. 1 Automatic standby when motor stops (Half torque) Nominal mode provides an important holding torque whereas Standby mode lessens power consumption and motor heating.

Command Index	**03h**
Function type	Parameterization
Function	Stops management mode
Command	Stx 0 1 4 a a **0 3** 0 0 x x x x x x 0 d C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	d 0 Hardware and Software stops disabled 1 Hardware stops enabled, Software stops disabled 8 Hardware stops disabled, Software stops enabled 9 Hardware and Software stops enabled

Command Index	**07h**
Function type	Parameterization
Function	Reference input management
Command	Stx 0 1 4 a a **0 7** 0 0 x x x x x x O d C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	d = 1 Reference input enabled (edge-triggered position reset) d = 0 Reference input disabled. This is default configuration at startup.

Command Index	**08h**
Function type	Configuration
Function	Module address programming
Command	Stx 0 1 4 F F **0 8** 0 0 x x x x x x a a C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Sets the new address of the only module on the serial data-line.

Command Index	**09h**
Function type	Parameterization
Function	Defines superior software stop
Command	Stx 0 1 4 a a **0 9** 0 0 b b b b b b b b C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Programs the value of the superior software stop with signed hexadecimal value bbbbbbbbh. This value is memorized in case of a power-off. If the axis position is superior to that stop, the motion is immediately stopped (maximal braking) provided software stops are enabled.

Command Index	**0Ah**
Function type	Parameterization
Function	Defines inferior software stop
Command	Stx 0 1 4 a a **0 A** 0 0 b b b b b b b b C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Programs the value of the inferior software stop with signed hexadecimal value bbbbbbbbh. This value is memorized in case of a power-off. If the axis position is inferior to that stop, the motion is immediately stopped (maximal braking) provided software stops are enabled.

Command Index	**0Bh**
Function type	Parameterization
Function	Position Reset
Command	Stx 0 1 4 a a **0 B** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Axis position counter is reset. This command enables the definition of origin position.

Command Index	**0Ch**
Function type	Parameterization
Function	Axis maximum torque limitation
Command	Stx 0 1 4 a a **0 C** 0 0 x x x x x x d d C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	This command enables the limitation of the torque provided by the motor. The hexadecimal "dd" value must be between 00h and Imax (See query 28h). The maximum torque is proportional to this value and to the axis' maximum torque. This data is stored in case of a power-off.

Command Index	**0Dh**
Function type	Parameterization
Function	Module High Speed definition
Command	Stx 0 1 4 a a **0 D** 0 0 x x x x v v v v C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Modification of the module's High Speed. The new value is given as a period (2 hexadecimal bytes): the time interval between two increments is then v v v v * 0,125µs. This parameter is stored at power-off. MAC23: $\Omega_{(RPM)} = \frac{60}{VVVV * 0{,}125 \cdot 10^{-6} * 2000} = \frac{240000}{VVVV}$ MAC34: $\Omega_{(RPM)} = \frac{60}{VVVV * 0{,}125 \cdot 10^{-6} * 10000} = \frac{48000}{VVVV}$

Command Index	**0Eh**
Function type	Parameterization
Function	Unstall Speed definition
Command	Stx 0 1 4 a a **0 E** 0 0 x x x x V V 0 0 C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	The hexadecimal value V V h gives the speed used to adjust the real position to the set-point position in case of a point-to-point motion (see index 13h and 14h). The time interval between two increments is then V V * 200µs. This parameter is stored at power-off.

Command Index	**0Fh**
Function type	Parameterization
Function	Acceleration setting
Command	Stx 0 1 4 a a **0 F** 0 0 x x x x x x A A C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	defines the number of increments on acceleration and deceleration ramps when motions follow the trapezoidal velocity profile (motions executed by index 13h, 14h and 15h). The number of increments made on each ramp is 120 * AAh. This parameter is stored at power-off.

Command Index	**10h**
Function type	Motion
Function	Set Synchronous Mode
Command	Stx 0 1 4 a a **1 0** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Notifies that the next motion will be synchronized (i.e. started by the command of index 11h).

Command Index	**11h**
Function type	Motion
Function	Synchronization signal
Command	Stx 0 1 4 a a **1 1** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Starts the preloaded motion. The module has to sequentially receive the command 10h followed by a command which accepts Synchronous Mode (13h, 14h or 15h) and finally receive this synchronization.

Command Index	**13h**
Function type	Motion
Function	Movement toward the given position with a trapezoidal velocity profile.
Command	Stx 0 1 4 a a **1 3** 0 0 p p p p p p p p C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	This command starts a movement toward the absolute position pppppppph (signed hexadecimal value). The speed follows a trapezoidal profile: - Acceleration up to Maximum Speed (see command 0Dh) on a given number of increments (see command 0Fh). - Level at constant Maximum Speed provided the global displacement is important enough. - Deceleration following a ramp similar to the acceleration ramp. - Adjusting of the position to the set point with a speed limited by the unstall speed.

Command Index	**14h**
Function type	Motion
Function	Back to Origin with a trapezoidal velocity profile
Command	Stx 0 1 4 a a **1 4** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Go back to origin position (see command 13h with a parameter of 00000000h).

Command Index	**15h**
Function type	Motion
Function	"Infinite" motion at given speed
Command	Stx 0 1 4 a a **1 5** 0 0 x x S S V V V V C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Starts a movement at speed V V V V. The first movement of this type starts the motor in the counter-clockwise direction if SS=F6h or in the clockwise direction in any other case. The movement is accelerated following the predefined ramp up to the set-up velocity. This velocity is given by the 2 hexadecimal bytes V V V V as a period: the time interval between two increments is V V V V * 0,125µs. MAC23: $\Omega_{(RPM)} = \frac{240000}{VVVV}$ MAC34: $\Omega_{(RPM)} = \frac{48000}{VVVV}$ During this kind of motion, the speed can be modified while the motor is running by a new command 15h, but the direction can not be modified. That motion can be stopped by commands 17h, 18h or by this command 15h with an argument V V V V = 0000.

Command Index	**16h**
Function type	Motion
Function	Interpolated motion
Command	Stx 0 1 4 a a **1 6** 0 0 p p p p v v v v C_1 C_2 Etx aa module address (2 ASCII-encoded characters) pppp Position parameter as the number of steps to be carried out (4 hexadecimal ASCII-encoded characters) vvvv Velocity parameter as the number of time interval (0,125µs) per step (4 hexadecimal ASCII-encoded characters) C_1 C_2 module checksum (2 ASCII-encoded characters)
Action	Executes a motion at constant speed according the Velocity parameter for the number of steps specified by the Position parameter. Note: This motion command is stored by the module in a 3-stages buffer memory (FIFO) before interpolation, which enables "interpolated motion" type operating mode. When the FIFO is saturated, additional commands are rejected (reception of a XONERROR character) until the FIFO is no longer saturated. The 1Eh index gives an equivalent temporization with no motor movement.

Command Index	**17h**
Function type	Motion
Function	Stop with a predefined deceleration
Command	Stx 0 1 4 a a **1 7** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Executes a braking limiting the deceleration to the predefined ramp coefficient. The motor stops but remains supplied in order to keep a holding torque in that position. This command can interrupt motions 13h, 14h and 15h.

Command Index	**18h**
Function type	Motion
Function	Emergency stop (maximum braking)
Command	Stx 0 1 4 a a **1 8** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Executes an emergency stop. The motor brakes with its maximal torque and stops. It remains supplied once stopped. The axis position is still valid but the position regulation loop is stopped. This command can interrupt any motion.

Command Index	**19h**
Function type	Motion
Function	Motor Power On
Command	Stx 0 1 4 a a **1 9** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	The motor is powered-on and supplies a holding torque limited by a predefined value (see command 0Ch) divided by 2 if Standby mode is enabled. If necessary, this powering-on is automatically executed when a motion command is executed.

Command Index	**1Ah**
Function type	Motion
Function	Motor Power off
Command	Stx 0 1 4 a a **1 A** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	Powers off the motor. Note that the axis no longer has any holding torque. Warning: it can be damageable for the power supply to power off while the motor is running. Moreover, this command does not provide an effective braking, one should use the command 18h to have an emergency stop.

Command Index	**1Bh**
Function type	Parameterization
Function	Baudrate setting
Command	Stx 0 1 4 a a **1 B** 0 0 x x x x x x v v C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) vv 01 for 9600 bauds 02 for 19200 bauds 03 for 38400 bauds
Action	Modifies the baudrate and reinitializes the module. The baudrate is then stored even after power off. The default factory setting is 38400 bauds.

Command Index	**1Ch**
Function type	Parameterization
Function	Tolerance in Position mode setting
Command	Stx 0 1 4 a a **1 C** 0 0 x x x x x x T T C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Action	The hexadecimal value TTh gives the number of increments tolerated between real position and set-point position.

Command Index	**1Eh**
Function type	Motion
Function	Interpolated motion, virtual segment
Command	Stx 0 1 4 a a **1 6** 0 0 p p p p v v v v C_1 C_2 Etx aa module address (2 ASCII-encoded characters) pppp Position parameter as the number of steps to be carried out (4 hexadecimal ASCII-encoded characters) vvvv Velocity parameter as the number of time interval (0,125µs) per step (4 hexadecimal ASCII-encoded characters) C_1 C_2 module checksum (2 ASCII-encoded characters)
Action	Produces a segment with no motor movement. The timing of this "virtual" segment is similar to real segment (index 16h) programmed with the same pppp and VVVV parameters. This index can be used to stop an axis with no synchronization loss in a multi axis interpolation. For example : … Stx 0 1 4 0 0 1 6 0 0 0 3 E 8 1 0 0 0 C_1 C_2 Etx (motor 00 moves) Stx 0 1 4 0 1 1 E 0 0 0 3 E 8 1 0 0 0 C_1 C_2 Etx (motor 01 is stopped while 00 moves) … Note: This motion command is stored by the module in a 3-stages buffer memory (FIFO) before interpolation, which enables "interpolated motion" type operating mode. When the FIFO is saturated, additional commands are rejected (reception of a XONERROR character) until the FIFO is no longer saturated.

Command Index	**20h**
Function type	Query
Function	Reading of the current position
Command	Stx 0 1 4 a a **2 0** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Answer	Stx 0 1 1 a a **0** p p p p p p p p C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) The signed hexadecimal 4-bytes-long value pp pp pp pp gives the current position of the interrogated axis. This position is memorized at power-off. At power-on, it is then still valid, provided the axis has not been moved.

Command Index	**21h**
Function type	Query
Function	Reading of the module status
Command	Stx 0 1 4 a a **2 1** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Answer	Stx 0 1 1 a a **2** p p p p p p p p C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) The signed hexadecimal 4-bytes-long value pp pp pp pp gives the position of the Software superior end-stop of the axis. This position is memorized at power-off. Stx 0 1 1 a a **1** d d f f g g h h C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) 1st byte (dd) : redhibitory defect. Those defects are memorized as they appear. They are acknowledged by a reset (index 00h or powering off the module). Disjunctions cause the motor power to be switched off. All motion commands are then refused until the default is acknowledged. dd Defect word Bit 7 Reserved Bit 6 Over-voltage warning (not memorized) Bit 5 Under-voltage warning (not memorized) Bit 4 MAC23 Thermal disjunction Bit 3 MAC34 Amplifier disjunction (over-current, ground leak, thermal disjunction, power supply over-voltage > 125% Vmax) Bit 2 MAC34 Thermal disjunction Bit 1 MAC34 Encoder not found Bit 0 Power line disturbance disjunction ff Status word #1 Bit 7 Reserved Bit 6 Reserved Bit 5 Reserved Bit 4 Reserved Bit 3 Hardware superior end-stop Bit 2 Hardware inferior end-stop Bit 1 Software superior end-stop Bit 0 Software inferior stop gg Status word #2 Bit 7 Reserved Bit 6 to 0 Reserved hh Status word #3 Bit 6 to 0 over-voltage warning (memorized), available since 7.4 software release. This bit is cleared by a power-on reset or by an "00h" reset order. It can be set by two causes: an over-voltage on the supply power lines or a strong braking (mechanical energy => electrical energy). This situation is not dangerous for the MAC but if necessary the braking efficiency can be improved by a ballast system - please contact us. Note: this bit can be set even if the system is equipped with a ballast because the ballast threshold voltage is generally higher than this warning level.

Command Index	**22h**
Function type	Query
Function	Reading of Software superior end-stop
Command	Stx 0 1 4 a a **2 2** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Answer	Stx 0 1 1 a a **2** p p p p p p p p C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) The signed hexadecimal 4-bytes-long value pp pp pp pp gives the position of the Software superior end-stop of the axis. This position is memorized at power-off.

Command Index	**23h**
Function type	Query
Function	Reading of Software inferior end-stop
Command	Stx 0 1 4 a a **2 3** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Answer	Stx 0 1 1 a a **2** p p p p p p p p C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) The signed hexadecimal 4-bytes-long value pp pp pp pp gives the position of the Software inferior end-stop of the axis. This position is memorized at power-off.

Command Index	**24h**
Function type	Query
Function	Reading of the software release.
Command	Stx 0 1 4 a a **2 4** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Answer	Stx 0 1 1 a a **4** V V V V V V V V C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) V V V V V V V V gives the version release number.

Command Index	**25h**
Function type	Query
Function	Reading of the set-point velocity
Command	Stx 0 1 4 a a **2 5** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Answer	Stx 0 1 1 a a **5** r r 0 0 v v v v C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) r r 0 0 stands for the unstall velocity (see index Eh) v v v v stands for the set-point velocity (see index 0Dh).

Command Index	**26h**
Function type	Query
Function	Reading of Ramp and Torque coefficients
Command	Stx 0 1 4 a a **2 6** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Answer	Stx 0 1 1 a a **6** n n c c K m d d C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) n n Ramp hexadecimal coefficients (from 00h to FFh). The acceleration or deceleration ramp performs a number of increments equal to 120 * n n. c c Torque hexadecimal coefficient (from 00h to 80h). Motor torque is limited to the value Tm = Tmot * c c / Imax With Tmot maximum motor torque and Imax adjustable with command 28h. K Stop torque management: 0 Nominal current forced when motor stopped 1 Automatic stand-by when motor stopped (half torque) m End-stops management: 0 Software and Hardware end-stops disabled 1 Software end-stops disabled, Hardware end-stops enabled 8 Software end-stops enabled, Hardware end-stops disabled 9 Software and Hardware end-stops enabled dd Position mode tolerance

Command Index	**27h**
Function type	Query
Function	Reading of analog measures
Command	Stx 0 1 4 a a **2 7** 0 0 x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters)
Answer	Stx 0 1 1 a a **7** t t v v c c d d C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) t t Hexadecimal value giving the temperature: **In case of a MAC23**, between 0 and 100℃, temperature is given by: Temperature (℃) = 2,5 x measure – 215 (measure = hexadecimal value of tt) **In case of a MAC34-1 or a MAC34-2**, temperature is given by : Temperature (℃) = 2,45 x measure – 140 v v Hexadecimal value giving the Power supply voltage measurement: MAC23 : Valim (V) = measure/4,8 (measure = hexadecimal value of v v) MAC34-1 : Valim (V) = measure x 0,4 MAC34-2 : Valim (V) = measure x 0,8 c c 1 hexadecimal byte representing inputs/outputs bit 7: reserved bit 6: reserved bit 5: Positive end-stop input (=1 if input is activated) bit 4: reserved bit 3: "busy" (=0 if the motor cannot follow the motion set-point velocity) bit 2: reserved bit 1: reserved bit 0: reserved d d 1 hexadecimal byte representing inputs/outputs bit 7: reserved bit 6: Negative end-stop input (=1 if input is activated) bit 5: Reference input (=1 if input is activated) bit 4: reserved bit 3: reserved bit 2: reserved bit 1: reserved bit 0: reserved Note: Reading that information enables checking the wiring of inputs and reference top.

Command Index	**28h**
Function type	Query
Function	RAM reading
Command	Stx 0 1 4 a a **2 8** @ @ x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) @ @ Address of the first byte to be read.
Answer	Stx 0 1 4 a a **8** p p p p p p p p C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) pp pp pp pp are the four RAM bytes read, starting at address @ @.
Example	Reading of Imax coefficient (this coefficient is located at address AFh). Query Stx 0 1 4 a a **2 8** A F x x x x x x x x C_1 C_2 Etx Answer Stx 0 1 4 a a **8** v v x x x x x x C_1 C_2 Etx The hexadecimal value v v gives Imax coefficient.

Command Index	**29h**
Function type	Query
Function	EEPROM reading
Command	Stx 0 1 4 a a **2 9** @ @ x x x x x x x x C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) @ @ adresse du premier octet à relire
Answer	Stx 0 1 1 a a **9** p p p p p p p p C_1 C_2 Etx aa module address (2 ASCII-encoded characters) C_1 C_2 message checksum (2 ASCII-encoded characters) pp pp pp pp are the four EEPROM bytes read, starting at address @ @.

MAC EXPERT PROTOCOL EXAMPLE:

Infinite motion at 300 rounds/min, clockwise direction for a MAC23 which address is 00.

300rounds/min ⇔ 5rounds/s ⇔ 10000inc/s (MAC23: 2000inc/round)
⇔ 10^{-4} s/increment.

Set-point velocity parameter will then have to be:

$$VVVV = \text{hexa}\left(\frac{10^{-4}}{0.125\mu s}\right) = \text{hexa}(800) = 0320h \text{ (2 bytes)}$$

Rotation direction: clockwise direction if SS=00h, counterclockwise if SS=F6h.

The command frame is then built as follows:

Stx	length			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	0	3	2	0	c	c	Etx
02h	30h	31h	34h	30h	30h	31h	35h	30h	30h	30h	30h	30h	30h	30h	30h	33h	32h	30h	41h	42h	03h
				CHECKSUM? 1) Hexadecimal sum: 30h+30h+31h+35h+30h+30h+30h+30h+… …+30h+30h+30h+33h+32h+30h = 2ABh 2) 2ABh modulus 100h is ABh 3) Checksum c c characters are then "A" and "B", which ASCII values are 41h and 42h															

Note: you can check parameters conversions using MACSIM.
Example with the message up-above:
Stx 0 1 4 0 0 1 5 0 0 0 0 0 0 0 3 2 0 c c Etx
With MACSIM, send directly "00150000000320" without having to specify neither "stx 0 1 4" nor "c c etx