Go Safer, Smarter, Smaller, Simpler.
Go Platinum.

Command Reference for Platinum & Titanium Line Drives

March 2025 (Ver. 3.002)

This guide is delivered subject to the following conditions and restrictions:

- This guide contains proprietary information belonging to Elmo Motion Control Ltd. Such information is supplied solely for the purpose of assisting users of Elmo's servo drive(s) in their installation.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Elmo Motion Control and the Elmo Motion Control logo are trademarks of Elmo Motion Control Ltd.
- Information in this document is subject to change without notice.

Elmo Motion Control	Elmo Motion Control and the Elmo Motion Control logo are registered trademarks of Elmo Motion Control Ltd.
EtherCAT Safety over EtherCAT	EtherCAT® and Safety over EtherCAT® are registered trademarks and patented technologies, licensed by Beckhoff Automation GmbH, Germany.
EtherCAT Conformance tested	EtherCAT® Conformance tested is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
CANopen	CANopen compliant. CANopen® is a registered trademark and patented technology, licensed by CAN in Automation (CiA) GmbH, Kontumazgarten 3, DE-90429 Nuremberg, Germany.

Document no. MAN-P-CR (Ver. 3.002)

Revision History

Version	Date	Details
Ver. 1.000	Apr 2017	Initial version
Ver. 1.001	Apr 2017	Initial Release version
Ver. 1.002	Jul 2017	**Addition of command EE.** **Changes to GI, SR, S1…8/F1….8,**
Ver. 1.003	Aug 2017	**Changes to AC, BP, CA, DC, EM, OB, QS, SR, US, VH/VL** **Addition of commands DH, EI, OB, OL, OP, SA, TR**
Ver. 1.004	Dec 2017	**Addition of commands AD, AS, CV,** **Changes to EM, BZ, EE, EC[], IB[], RS, S1…8[]/F1….8[], SW**
Ver. 1.005	Feb 2018	**Addition of commands DN, MF, YS, GC, GG, TW, WF** **Changes to DH, EC, EE, IB, SR, S1…8/F1….8, MF, CA, SC, SG, SR, TS, UM, VR, XP** **Removed obsolete commands PY, YM**
Ver. 1.006	Jun 2018	**Changes to ER, GI, GX, GY, HM, IL, IP, OL, S1…8[]/F1….8[], SA, TR** **Updated section Drive General error List**
Ver. 1.007	Nov 2018	**Changes to AE, BZ, CN, HM, LD, MF, OL, RS, SV, H1..H8[]** **Addition of AP, TM, WS, XF, XI, XO** **Addition of Interpreter Commands/Objects: OJ, OH**
Ver. 1.008	Jan 2018	**Changes to XA, EE, GX, H1..H8[], HM, IL, OB, OL, OP, SR, WS, CZ** **Addition of KR, MR, G1..G5, GO** Changes to the Drive General Error Codes
Ver. 1.009	Jun 2019	**Changes to SA, AE[], CV, EE[], EG[N], MP[], OP, S1….S8[], T1…8[N], W1...8[], WS, XA[]** **Addition of AB[], EA[N],**
Ver. 1.010	Aug 2019	**Changes to AB, MP, N#, MF, EA**
Ver. 1.011	Nov 2019	**Changes to AB, AF, IB, BZ, MS, DV, GS, PO, S#F#, TR, VR, WS, XI** **Addition of NF, NT**
Ver. 1.012	Dec 2019	**New commands AN, ZX, ZF, TC, FP, CS** **Changes to CV, AB, AC, AD, AS, BG, BH, BV, CA, CL, CN, DC, DV, E#, EG, EM, ER, ET, FC, FF, FV, GS, H#, HL/LL, KC, KI/KP, LC, MF, NF, PE, PL, PO, PZ, QS, RM, S#, F#, SA, SC, SO, SP, ST, T#, TI, TR, TS, TW, US, VH/VL, VU, VX, W#, WF, WS, XA, XP,** **Correction to DH,**
Ver. 1.013	Jan 2020	Changed banner **Changes to command BZ**
Ver. 1.014	Mar 2020	**Changes to command PP,** **Added new Interpreter Prefixes AX, GR**

Version	Date	Details
Ver. 1.015	Mar 2020	**Changes to command BZ,** Added new section Drive Communication & Interpreter
Ver. 1.016	Apr 2020	**Changes to commands GI, GO, IL, OL**
Ver. 1.017	May 2020	**Changes to commands BP, EE, GO[N], OB[N], OL[N], OP, SR** **Added new sections:** **CR[N] – Capture Resource Selection** **GC[N] – Configure Port C Mux** **GT[N] – Capture Time MUX Selection** **Removed WS – Miscellaneous Reports**
Ver. 1.018	Aug 2020	**Changes to commands CR, CN, SA, GI, IL, IP, OL, OP, RS, SA**
Ver. 1.019	Aug 2020	Small corrections to text in Commands
Ver. 2.000	Oct 2020	Corrections to all Attribute tables and new template
Ver. 2.001	Dec 2020	**Corrections to all Attribute tables and changes to commands:** **AN, BP, EA, EE, GI, GT, GX, GY, IL, OP,** **New Commands GV, OC,**
Ver. 2.002	Mar 2021	**Corrections to Interpreter Prefixes and OH.** **Corrections to all Attribute tables and changes to commands:** **CA, DC,**
Ver. 2.003	Apr 2021	Corrections to **Interpreter Prefixes and OH.** Corrections to **all Attribute tables and changes to commands:** **AE, AY, CA, CN, CR, DC, EE, GO, MF, OC, S1…8[]/F1….8[], WE, WT** New Commands: **EV,VP, VS**
Ver. 2.004	May 2021	Corrections to **RP, SO,**
Ver. 2.005	May 2021	Corrections to **GS, OS, SA, AB, GT, IF,** New command **SG**
Ver. 2.006	June 2021	New commands **MO, R1, R2, R3, R4 added**
Ver. 2.007	Aug 2021	**Correction to OC, AP,**
Ver. 2.008	Oct 2021	New Commands added: **CT, HP, MI, PS, RZ, SA, XC, XQ** Correction to **AP, ER, GT, GV, HM, MO, OC, PL, PC, SR, WT,**
Ver. 2.009	Dec 2021	Update to links and small corrections
Ver. 2.010	Jan 2022	Removal of command **ZA** **New Commands added: DD** **Corrections to OC, G1…G5, GC, AN, TF, IF, GG, CA, EE, PC, IL, WT**
Ver. 2.011	May 2022	Corrections to **PC, PL, RC, RV, MP, RP**
Ver. 2.012	May 2022	**New Commands added: BS, CP, KL, XT**
Ver. 2.013	Jun 2022	Corrections to **PU**, **WF**, **X0**

Version	Date	Details
Ver. 2.014	Aug 2022	Corrections to **S1…8/F1….8** and **SC[N]**
Ver. 2.015	Aug 2022	**Corrections to GI[N], S1…8/F1….8, XO** **New Commands added: MQ**
Ver. 2.016	Nov 2022	Addition of Sensor Errors section related to W1...8[] command.
Ver. 2.017	Nov 2022	Corrections to: **PC, SC,** New Command: **PW**
Ver. 2.018	Feb 2023	Corrections to: **WS, GC, CL, LC, D**rive Error Codes, **OC,** TF, **PL,** New Command: **BU**
Ver. 2.019	Mar 2023	Change Alias Object values of a number of commands to "Not Available"
Ver. 2.020	May 2023	Addition of section "Sensor Error Codes and Warnings"
Ver. 2.021	May 2023	Update of section "Sensor Error Codes and Warnings"
Ver. 2.022	Aug 2023	Correction of section "Sensor Error Codes and Warnings" Correction to: **GB**, **EE, KL. WS** New Commands: **GD**,
Ver. 2.023	Oct 2023	Correction to **S1….8** Sensor Socket Parameters
Ver. 2.024	Nov 2023	Added Note to Remarks section in commands: **CA**, **H1…H8[]**, **S1…S8[]/F1…F8[]**, **SA**
Ver. 2.025	Mar 2024	Update of: **AB**, **AF**, **BZ**, **CR**, **GX, Drive Error Codes, CF, EE, PP, RS** New command **CC,**
Ver. 3.000	Aug 2024	Added **GA[N]** (new). Updated **ET[N]** with Alias object and a Remark. Updated **FI[]** with Titanium Drive (for index 17). Updated **GO[N]** with info and tables for Titanium Drive Updated **IL[]** with info and tables for Titanium Drive. Updated **IP** with info on Titanium Drive for Bits 24 - 31. Updated **OL[N]** with info and tables for Titanium Drive. Updated **OP** with info on Titanium Drive for Bits 8 – 15. Updated **S1…S8[]/F1….F8[]** with into on Titanium Drive in the table "Encoder Quad". Update of: **AB**, **AF**, **BZ**, **CR**, **GX, Drive Error Codes, CF, EE, PP, RS, S1…8 Motion Sensor 0x3604, T#[1…8], Updated Sensor error Codes,** New command **CC,** New table **EE[1]** Corrections to **CF**, **CL**, **CR**, **ET, GA, GO, IF, IL**

Version	Date	Details
Ver. 3.001	Dec 2024	**MF** EMCY Hex Code 3130 corrected Corrections to Drive Error Codes Corrected the following commands: **AB, CV, WS, XA,**
Ver. 3.002	Mar 2025	Corrected the following commands: **AN, BZ, TI, FF, GO, GS, KI, KP, OC, OL,** Added the following commands: **CG,**

General

Throughout this Platinum command reference document there will be parameter variations between the Platinum, Quartet, and Titanium drives. It should therefore be noted that when referencing a specific command and its Indexes/ Sub-indexes, to make sure to check if there is a variation of its Indexes/ Sub-indexes parameters dependent on the drive type, before applying the relevant parameter.

Description of Attributes

Note:
In some situations where indexes(bits) are not tabulated, the indexes are to be considered as Reserved and may be set to 0.

For indexed commands:

Attribute	Description
Type	The data type of the variable (integer, float, bit field etc.) and the access mode (read-only, read/write or command).
Source	The entry source: RS232, USB, TCP, EoE, CoE or CANopen *Mapping* means that the object can be mapped to a process data object (PDO). *User Program* means that the variable can be manipulated from an Elmo user program. *FoE* in some commands means that File Over EtherCAT can be used.
Restrictions	Commands might be limited to specific conditions. These limits should be described in this attribute. The reason for a restriction may be a safety consideration, consistency with other commands or relevance in a specific context or product model. For example: *Not User Program* means that the command cannot be used in a User Program. *Motor Must Be On* indicates that the command can only be executed if the servo is enabled.
Range	Indicates the maximum and minimum permissible values for the specific commands. In some cases, the command alters CANopen (CoE) objects which are user units, and conflicts may occur when the resulting value is out of range. This may happen when the user sets a factor that multiplies the value to an "out of range" value.
Index range	For indexed commands that have several inputs, there are two cases: **1) Inputs with the same meaning and weight. For these commands all entries have the same meaning and are described as scalar commands (e.g., ZX[*N*], ET[*N*]).** **2) Inputs with different meanings. Inputs may have different meanings for different uses. In this manner even the context might differ. The manual**

Attribute	Description
	describes each input as a specific command (e.g., CA[*N*], SC[*N*]). In this case an entry description that details every input is added (see below). **In some cases, when the specific feature which is normally used by the array is not defined, the array memory space may be used as a buffer for other needs. These cases are described in the relevant commands (see, for example, HM[*N*] (captured mode))**
Default	**The default value after the RS command.**
Unit modes	**UM. The relevant controller which can be used for the specific command/parameter.**
Non- Volatile	***Yes* means that the information is saved to flash memory after the SV command (or CANopen object 0x1010).** *No* means that the information is not saved.
Axis Related	Is the Command axis related? When YES, the command may influence the motion of the axis.

Remarks

An indexed command which has a different meaning for every input will include the following table with a description of each entry:

Index	Description	Type	Values	Restrictions
Number of input	Command description	**Similar to the Type attribute**	Range or any other value description, as is applicable to the command	**Similar to the Restriction attribute**

Drive and Sensor General Error Lists

The Drive general error list contains error codes returned by **EC[]**, **EE[4]** and **EE[5]** commands. When SDO (used in CANopen/ECAT) returns a general abort message `0x08000000` which indicates an error while processing the SDO command, **EE[4]** provides further details about the error. Other commands return error codes from this list as well when a request was not successful, unless otherwise specified in command description.

Drive Error Codes

The following table details the drive general errors which are referenced throughout this document.

Error code	Error String	Description/Example/Remedy
1	Do not update	The value cannot be updated due to the present drive state or drive definition. Possible reasons are no sensor was assigned to the socket, an attempt to change PDO parameters while PDO is enabled
2	Bad command	The command is not supported, or command syntax is incorrect.
3	Bad index	Command index (object sub-index) does not exist
4	Min or Max Conflict	Conflict between the setting of Max value and the Min value of the parameter.
7	Bad mode initial data	Requested mode or sub-mode cannot be started due to not suitable initialization data
8	Bad output Compare initialization	Incorrect parameters set of Output Compare. Refer to **EE[20]** for details
10	Not CAN PDO	**Request string commands BH, CD, DL, UL, DB, DF, RH, WB, VR, XQ via CAN binary interpreter returns with this error code.**
11	Flash write failed	SIL user parameters save procedure had failed. Cannot write to internal flash memory
12	Safety not enabled	Change safety mode failed due to DI definition "Non safety".
16	Array '[ ]' is expected	Incorrect command syntax. The command should be an array or empty expression in array
17	Upload process failed	Serial FLASH upload failed. See specific error in **EE[7].** In the case of **EE[7]=8**. Refer to details in **EE[11]**
19	Bad command format	The command contains invalid character or expected character missing
20	Bad PVT send order	Drive received position and velocity set points in incorrect order, for example: position, position, velocity, velocity while correct order is: position, velocity, position, velocity.

Error code	Error String	Description/Example/Remedy
		Relevant for Interpolated position operation mode, sub-mode -1.
21	Out of range	Command value out of range defined for its value. Refer to table Description of Attributes for relevant command description in Command Reference for Platinum drive.
23	Attempt to write RO	An attempt to write read-only parameter. Command cannot be assigned
26	Profiler mode not supported in actual unit mode	**An attempt to set operation mode (Object 0x6060**) to value that is not fit to actual unit mode **(UM).** For example, an attempt to set **Object 0x6060 to 1 (Profiled position)** while the drive is in "Speed control loop" **(UM==2).** Refer to **UM** command description.
27	Bad ECAM setting.	**ECAM settings by EM[] command is incorrect. Refer to EE[6] for details.**
28	Out of limit	**An attempt to set value that is out of limits, defined by another command, for example VL[], VH[].**
29	Bad TS time setting	Bad cycle time.
30	Bad error mapping table definition	An attempt to set table start index **PC[4]** greater than table end index **PC[5].** An attempt to set start position **PC[7]** other than 0 in cyclic error correction mode. Refer to **PC[]** command explanation**.**
31	Bad socket value	Bad socket selected - **HM[10]\HF[10]**
32	Bad sensor selected	Cannot send main position sensor via **HF[]**, to Home.
33	Bad sensor setting	Commutation fails on motor enable process due to incorrect sensor settings. Check **CA[18], MP[4], SA[] and S#[]** command setting.
34	Command conflict	There is a conflict due to another request pending at the time of this command
35	Capture resource not assigned	Capture resource required for **HM/HF** homing was not assigned by **CR[]** command
36	Bad commutation setting	Commutation method **(CA[17])** or commutation table does not fit to sensor
37	Hall location conflict	Hall sensors are defined to the same place, check **H1…H8** commands**, bits 3 to 8**
39	Total Hall width exceeding	Total halls width is incorrect, the width for the halls state is defined in **H#[2-6].** The sum of Hall width shall not exceed 4096
40	Sensor command conflict	Command cannot be executed due to a conflict with another process

Error code	Error String	Description/Example/Remedy
41	**FP** command conflict	Command cannot be applied to feedback sockets
42	No such label	User program fails. Incorrect function name.
44	Advanced filter setting error	Bad advanced filter setting, refer to **EE[16]**
45	Attempt to read WO	An attempt to read write only command or object
47	Program not compiled	An attempt to run or resume user program that was not previously compiled or was erased
51	Inhibit or Abort function is active	Motor enable process failed. Refer to **IL** and **IP** command
54	Bad database	Parameters save into FLASH or parameters load to RAM from FLASH fails due to corrupted parameters data base
56	Motor must be stopped	**Set PX or Set PU command fails due to motion status "In motion" (MS=2). Refer to MS command.**
57	Motor must be off	Requested command can be processed on motor disabled state only
58	Motor must be on	Requested command can be processed on motor enabled state only
60	Bad unit mode	Requested set command cannot be processed in current unit mode. Refer to **UM** command.
61	Current Sensor Supply	A malfunction in the power supply to the Current sensor. The servo drive cannot be enabled.
62	Socket change capture enabled	An attempt to change Touch probe socket assignment while Touch probe is active or homing socket assignment while homing is active. **Refer to SA[8] and SA[9] commands.**
63	Kinematics error	**Kinematics error, check EE[19]**
64	SIL must be off	**SIL must be disable**
65	SIL drive reboot needed after LD or RS	LD or RS was executed, reboot the drive to run SIL again
66	System timing was changed	An attempt to enable motor after system timing change by **TS**, **HS**, **XP[2]** commands. Drive reboot is required
67	Recorder is busy	**A**n attempt to set R**G, RC, RL, RV, FT, RP, RH, BH, BS, UL** commands while recorder is active
68	Non legal profiler mode	An attempt to enable motor in Mode of operation object 0x6061=0
69	Recorder setting error	Wrong settings of commands **RV, RP, BS** or **BH**

Error code	Error String	Description/Example/Remedy
70	Invalid recorder state	Requested command cannot be processed when recorder is in non-active state or not in Buffered mode **(for RA command)**
71	Homing is busy	Requested command cannot be processed when homing is active
72	Emulation following error	Emulation cannot be started due to emulation following error. To start emulation first disable it and reset error with **EA[1]=0**
73	Socket profiler in motion	**FP** command cannot be processed when socket profiler is in motion. Stop socket profiler
74	DS-402 Profiler problem	DS-402 Profiler configuration fails. Refer to **EE[2]** for details.
75	DL in progress	Requested command cannot be processed while download procedure is in progress
76	**HM HF** homing is enabled	Error mapping procedure cannot be processed when **HM/HF** homing is active
78	Out of program range	Incorrect jump address in user program. An attempt to jump out of program range.
79	Sensor setting error	Incorrect sensor settings. Check **S1[]…S8[]**, **F1[]…F8[]**
80	Bad HW configuration	Bad **HW** configuration. Check ECAT configuration **PP[23]**
81	Wrong sensor port resource	Sensor setting error, wrong combination of resource and port connection. Check **S#[2]** for the selected sensor and its allowed combinations.
82	Program is running	Requested command cannot be processed while user program is running
83	Command not for program	An attempt to run not permitted command in user program
84	Emulation in progress	Emulation is already running. To start emulation first disable it and reset error with **EA[1]=0**
87	Halls not connected	Halls are not connected or in illegal state
88	FW distribution failed	FW image burn failed, or debugger is enabled
89	Actual number of axes differs from number in FLASH	Total number of axes been read from the FLASH is greater than the permitted number of axes in the system
90	Thread is not allowed	Check if the thread number is >= **WS[37]**
91	User program must be active	User program must be activated

Error code	Error String	Description/Example/Remedy
92	Thread cannot change to active	Thread is already active and cannot be re-activated
93	Emulation in progress	Emulation is already running.
94	Error mapping in progress	Requested command cannot be processed while error mapping is in progress
95	Cannot run during user program	Cannot be executed during user program, need to terminate the program.
96	Program timeout	User program crashes due to timeout
98	Wrong current offset	Current offsets are beyond the allowed limit
99	Sensor does not support the command	Requested command is not relevant for sensor defined in system
100	Cannot modify PWM	Cannot modify PWM, its factor is not defined. Refer to **XP[2]** command
102	Output compare is busy	Requested command cannot be processed because the sensor is used by output compare
103	Output compare sensor not Quad	Output Compare sensor requires Quad encoder sensor settings.
104	Output compare table error	A conflict in the Output Compare table values.
107	Emulation parameter is out of range	This error is released on emulation initialization (**EA[1]=1…7**) when one of parameter **EA[2]….EA[17]** has incorrect value or emulated socket/axis is not defined. Refer to **EE[16]** command for details.
109	Incompatible system timing	Requested command cannot be processed due to conflict between **TS** and **HS** settings
110	Number too long	The number contains too many characters
111	Current exceeded or short on motor enable	Current exceeded or short detected on motor enable
112	Wrong signal source	Wrong function selected as a signal source to the socket
114	Vector axis must be virtual	Separate vector axis must be virtual
115	Vector must_be_stopped	Vector must be stopped
116	All axes must be on	All vector axes must be Enabled
117	Wrong axes servo status	All vector axes must have the same Servo On status
118	Axis is used by another vector	The axis is used by another vector

Error code	Error String	Description/Example/Remedy
119	Axis is vector member	Scalar Motion or Stop request to vector axis is not permitted.
120	Not supported for vector	Command or motion mode is not supported for vector axis
121	Vector must be disabled	Vector definitions cannot be changed when vector is enabled
122	Operation mode not supported	An attempt to run not supported operation mode
123	Profiler queue full	Profiled position operation mode supports up to 4 targets
124	Incorrect password	Safety password value is incorrect, or drive is in state that does not permit to change password
125	Mode under SIL control	Motion mode cannot be accepted while under SIL control
126	User program failed	User program run-time error. Incorrect variable or operand
128	Bad variable index	User program run-time error. Incorrect variable array index
129	Variable is not array	User program run-time error. Variable is not array
130	Bad variable name	Incorrect variable name in user program
131	Local user variable	User program error. Cannot record local variable
132	Sigma out of range	The Total value of the continuances (**CL[1]**) or peak (**PL[1]**) current limits was exceeded
133	Mismatch function arguments	User program error. Incorrect number of function arguments
134	Local user function	User program error. Only global functions or global labels are legal to use with **XQ** command
135	Frequency identification fail	Frequency identification of sin-cos signal (**TW[81]**) failed due to short sample vector or bad signal
136	Not a number	Interpreter fails on the command because processed character is not a number while number expected
138	Interpolation underflow	Interpolated position mode fails due to underflow protection. Refer to **Object 0x3775** description for details
139	Too many breakpoints	User program supports up to 6 breakpoints
140	Not relevant breakpoint	User program error. An attempt to set or clear breakpoint at the not relevant line
144	Numeric stack underflow	Processed command failed because there is no operand in numeric stack
145	Numeric stack overflow	Processed command failed due to numeric stack overflow
147	Executable command	Executable command within math expression

Error code	Error String	Description/Example/Remedy
150	Division by zero	Divide by zero is not allowed
151	Parentheses mismatch	Parentheses mismatch in user program
152	Bad operand type	Operand type is not suitable to address mode
153	Numeric overflow	An attempt to divide with zero or operand is out of range defined for its type
154	Out of data segment	User program error. Function argument is placed out of data segment
155	Beyond program stack range	User program error. Program stack pointer is out of stack range
156	Bad opcode	User program error. Incorrect opcode.
158	Parameters out of FLASH range	Parameters' size exceeds permitted FLASH range
159	FLASH verify error	User program was uploaded from FLASH with error
160	User SIL error	User SIL error - check **EE[14]**
161	Program not halted	Set or release breakpoint, or resume program fails because program is running
162	User SIL parameters error	User SIL parameters error - check **EE[18]**
163	Out of program data memory	User program error. Program data exceeds data segment size
165	FLASH read failed	FW image read from FLASH failed. An attempt to access flash while busy
166	Out of modulo	The resulted user unit position is out of modulo range **XM[]**
168	Sped too large to start motor	Motor enable failed due to large speed. Refer to **XA[2]** command.
169	CPU peripheral busy	Time out using peripheral. (overflow or busy)
171	SFLASH read failed	Load parameters or read CRC from Serial FLASH read failed. Address is out of range.
172	SFLASH write failed	Write to serial FLASH failed
173	User program too large	User program code, variables and text size too large. Refer to User Program manual
174	No program loaded	No user program loaded
175	User program CS not erased	**CC** command fails because checksum is not erased and not equals to entered value

Error code	Error String	Description/Example/Remedy
176	User program code and variables too large	User program code and variables size too large. Refer to User Program manual
177	Analog encoder conversion error on Capture, Homing, Output Compare	Quad location is unknown. Move axis for some counts, then try Capture, Homing, Output Compare again.
178	CAN bus off	CAN bus off state was detected
179	Consumer HB event	Consumer heartbeat event detected. Refer to **Object 0x1016** description in the MAN-P-ADMINGUIDE.
180	Flow Error	Flow error. See **EE[22]**
181	Program FLASH write failed	User program write to serial FLASH failed
184	Capture already used	Can be removed from code and personality
185	Interpolation is active	Drive in Interpolated position mode, interpolation active state. Settings of the objects 0x60C0, 0x60C2, 0x60C4 is permitted in interpolation non-active state only.
186	Interpolation queue full	Drive in Interpolated position mode, interpolation buffer overflow event occurred. Refer to DS-402 manual, chapter "Interpolated position mode"
188	Gantry slave disabled	Cannot enable motor due to Gantry slave state
189	CAN message lost	CAN message lost due to HW or SW buffer overflow
190	Not chain command	The next commands can't be in command chaining : **DL**, **DF**, **UL**, **BH**, **DB##GS**
191	Out stream overflow	There is not enough space in Stream Out for the interpreter's answer
193	Input stream overflow	Processed command fails due to overflow in input stream
194	OS interpreter	Processed command failed in OS interpreter
195	Zero acceleration	Profiled position mode. New target cannot be added to buffer due to zero acceleration/deceleration
196	Group was not set	Group command failed because axis group was not set
197	Invalid motor simulation configuration	**CA[28]** motor simulation bit doesn't fit socket definition
198	Sensor not supported	Processed command failed due to invalid sensor settings. Check **S1…S8** commands
199	Sensor resource used	Sensor or Halls resource is used by another socket.
200	Main feedback error	Main feedback error. Refer to **EE[1]**

Error code	Error String	Description/Example/Remedy
201	Commutation sequence failed	Commutation sequence failed on motor enable process due to active HW stop switch
202	Bad Halls source	Hall sensors are defined via **H#[1]** to an unavailable halls source
210	In Slave Remote Mode	Attached slave drive cannot be enabled or disabled locally
211	In Remote mode	Drive in Remote mode. Refer to **AF[1]**
212	Sensor protocol error	Analog or serial sensor protocol error, configuration uses the same bit for different interpretations
213	Sensor serial commutation error	Motor enable process failed because serial communication sensor not ready
214	Sensor not defined	Sensor not defined
215	MF cannot be warning	One or more bits in **AE[4]** is defined as fault only, cannot be set as warning
216	Serial exclusive sensor not defined	Serial communication sensor is not defined at the slave drive
220	Position HW limit	Position target is out of HW limits (FLS,RLS,STOP)
222	Position SW limit	Position target is out of SW limits. Refer to **HL[3]**, **LL[3]**
225	Under voltage	Under voltage detected during motor enable
227	Over voltage	Over voltage detected during motor enable
229	Safety state	Safety state detected during motor enable
231	Sensor error detected	Check **W#[1]\W#[3]** of the sensor
233	Short	Short detected during motor enable
235	Amplifier over temperature	Amplifier over temperature detected during motor enable
236	Motor over temperature	Motor over temperature detected during motor enable
237	Additional abort	Additional abort switch active state during motor enable
239	No data to be uploaded	No data to be uploaded
240	Bad double DC motor configuration	Double DC motor configuration must be cleared (**ax2.ca[28]=0**)
242	Bad motor configuration	Motor type is not supported by drive hardware
244	Bad axis	The axis number is out of range
245	Bad recorder header buffer	Recorder header buffer **RH** was not built due to errors
246	Bad recorder trigger settings	Recorder bad setting of trigger function (Instantaneously cannot be set when use operation trigger)

Error code	Error String	Description/Example/Remedy
247	WB command failed	WB command failed. See specific error in **EE[8]**
248	OJ object access failed	OJ object access failed. See specific error in **EE[9]**
249	The number is infinite	The number exceeds the maximum or minimum integer's value
250	Axes group not valid	An attempt to set number of axis out of range or two similar axes in group or more than 6 axes in group
251	Database CRC error	Parameters load from FLASH failed due to database CRC error
252	Invalid data in FLASAH	Parameters load from FLASH failed due to incorrect signature or parameter size out of range
253	Bad command type	Attempt to process parameters with incorrect type
254	Set to **ST** failed	The value is not sent to **ST**
255	Debug error	Reserved for debug purposes
32768	ECAT fail	0x8000. ECAT general error
32770	ECAT NOP	0x8002. ECAT error. Nothing has been done
32771	ECAT invalid argument	0x8003. ECAT error. Function is called with one or more invalid arguments
32772	ECAT invalid pointer	0x8004. ECAT error. Attempt to use invalid pointer
32773	ECAT out of memory	0x8005. ECAT error. Memory limit (cannot allocate new memory)
32775	ECAT Invalid state	0x8007. ECAT error. Invalid state
32782	Slave not initialized	0x800E. ECAT error. Slave has not been initialized
33026	Not enough buffer size	0x8102. ECAT error. Buffer size is not enough
33285	Invalid access type	0x8205. ECAT error. OD access type is invalid
33287	Access to OD denied	0x8207. ECAT error. Access to OD denied due to Al status
33288	OD not created	0x8208. ECAT error. Object cannot be added to OD due invalid attributes or index range
33290	SDO list not supported	0x820A. ECAT error. Requested SDO list is not supported. Refer to EE[4] for details
33298	SDO service not supported	0x8212. ECAT mailbox error: in CoE header contains not supported CoE service. Refer to **EE[4]** for details
33299	Invalid SDO header	Ox8213. ECAT mailbox error: invalid CoE SDO header
33300	SDO length too short	0x8214. ECAT mailbox error: CoE SDO service with Length less than 10
33301	ECAT SDO invalid size	0x8215 ECAT mailbox error: SDO has invalid size. Refer to **EE[4]** for details

Error code	Error String	Description/Example/Remedy
33303	Value is incorrect	0x8217. ECAT error. Entry value is more than maximum or has incorrect format
33552	Cannot open file	0x8310. ECAT error. Can't open file
33554	Wrong file name	0x8312 ECAT error. Wrong file name
33557	File access denied	0x8315. ECAT error. File access denied. e.g. wrong FoE password
33559	Cannot write to file	0x8317. ECAT error. Can't write to file
33560	Cannot read from file	0x8318. ECAT error. Can't read from file
34048	Cannot open xml file	0x8500. ECAT error. Can't open XML file
34049	XML error	0x8501. ECAT error. Error while parsing XML data
34080	Safety disabled	0x8520. ECAT error. Functional safety is disabled in DI, 0x1700, 0x1B00 mapping is disabled
34081	ECAT Mailbox full	0x8521. ECAT mailbox error: Mailbox buffer is full and not available for new message. Refer to **EE[4]** for details
34082	ECAT incorrect service code	0x8522. ECAT mailbox error: CoE Header contains incorrect service code. Refer to **EE[4]** for details
34083	ECAT invalid opcode	0x8523. ECAT mailbox error: SDO Info service has invalid opcode. Refer to **EE[4]** for details
34084	ECAT mailbox state	0x8524. ECAT mailbox error: Mailbox error: Mailbox state is not fit to service request. Refer to **EE[4]** for details
34304	ECAT error already handled	0x8600. ECAT error already handled. Reserved to mark, that abort error was set in previous function. Refer to **EE[4]** for details

Sensor Error Codes and Warnings

This section describes the errors and warnings derived from the drive internal stats and from the sensor. The relevant Commands used to define the errors and warnings are:

W#[1]	Socket latched errors (cleared on motor on)
W#[2]	Socket warnings
W#[3]	Socket active errors. These errors are **mask able** via **S#[4]** command.
W#[4]	Recorded extended sensor's status errors. These errors r**eflect the error register from sensor's datasheet.**
W#[5]	Recorded extended sensor's status warnings. These warnings r**eflect the error register from sensor's datasheet.**

It should be noted that:

- The errors and warning details are extracted from the sensor datasheet. It is highly recommended to refer to the datasheet relevant to the specific sensor version.
- All sensor errors shown in **W#[3]** can be converted to warnings by setting its bit in the error mask parameter **S#[4]**. In addition, for all sensors, bit 31 in the of the Error register reports a *Hall Sensor* Error.

The extended sensor's status will be set to its buffers (**W#[4]/[5]**) automatically when an error or warning have been occurred, except Nikon that requires the command **T#[7]=1** in order to get the extended errors.

For any sensor during initiation the errors are automatically cleared. In EnDAT 2.2 setting bit 14 of S#[2] will prevent this behavior and initiation errors will remain until cleared by the user as described below.

For every sensor, after fixing the error/warning, it is required to send the command **T#[3] = 1 or T#[3]=2 depending on the sensor in** to reset sensor's errors/warnings. In some situations, a drive power OFF/ON is required which is described for the relevant sensor in the tables below.

For EnDAT 2.2 sensor, setting bit 13 of **S#[2]** will indicate any warning informed by the sensor automatically.

The following table shows **common warnings and errors.** These indications are relevant to all sensors in **W#[2]** and **W#[3]** commands:

Bit	Type	Title	Details	Resolution
0	Error	Sensor's General Error	General error bit	Check error bit cause in **W#[4]** command, then check the error code and its resolution in sensor's data-sheet or in this doc.
1-6		Reserved		
7	Warning	Sensor's General Warning	General warning bit	Check warning bit cause in **W#[5]** command, then check the warning code and its resolution

Bit	Type	Title	Details	Resolution
				in sensor's data-sheet or in this doc.
8	Error	CRC Error	CRC Communication error	Check sensor configuration parameters. Check sensor physical connection & shielding
9	Error	Encoder ID error	The encoder ID doesn't exist or isn't consistent with the assigned resolution **– Only for encoders supports sensor ID, NRZ (Tamagawa, Panasonic, Mitsubishi, Yaskawa, Nikon)**	1. Set the correct sensor resolution according to the sensor type. 2. Check if this sensor supported.
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	Increase serial communication time. Increase clock frequency. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialize.	If the error persists: Check sensor and connection. Check sensor configuration parameters
12-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	Run commutation wizard again. Check Halls connection

Sensor ID=1: Encoder Quad
W#[2] / W#[3] Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Pin A diagnostics	Quad pin A differential line error	Check encoder Physical connection
1	Error	Pin B diagnostics	Quad pin B differential line error	Check encoder Physical connection
2-6		Reserved		
7	Error	Quad states error	State machine – faulty transition	Check encoder Physical connection
8-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	Run commutation wizard again. Check Halls connection

Sensor ID=3: Analog Sin/Cos
W#[2] / W#[3] Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-4		Reserved		
5	Error	Amplitude Error	Analog signal amplitude exceeded allowed limits	Run sensor calibration again Check sensor
6		Reserved		
7	Error	Quad states error	State machine – faulty transition	Check encoder Physical connection
8-13		Reserved		
14	Warning	Amplitude low threshold warning	Analog signal amplitude is below threshold limit	Check sensor warning limits. Check sensor
15	Warning	Amplitude high threshold warning	Analog signal amplitude is above threshold limit	Check sensor warning limits. Check sensor
16-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	Run commutation wizard again. Check Halls connection

Sensor ID=4: Halls Only
W#[2] / W#[3] Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	Run commutation wizard again. Check Halls connection

Sensor ID=5: BiSS
W#[2] / W#[3] Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	BiSS error bit	General error bit	For details read out **W#[4]** – display error code. And compare with below table.
1-6		Reserved		
7	Warning	BiSS warning bit	General warning bit	For details read out **W#[5]** – display warning code. And compare with below table.

Bit	Type	Title	Details	Resolution
8	Error	CRC Error	CRC Communication error	Check sensor configuration parameters. Check sensor physical connection & shielding
9		Reserved		
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	Increase serial communication time. Increase clock frequency. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: Check sensor and connection. Check sensor configuration parameters
12-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	Run commutation wizard again. Check Halls connection

Sensor ID=5: BiSS cont.

W#[4] / W#[5] – Sensor's Extended Status Errors/Warnings code

Bit	Type	Title	Details	Resolution
0	Error	Digital photo-amplifiers in saturation	Illumination of at least one of the digital photo-diodes is too high, digital photocurrent-amplifier	
1	Error	LED current low	LED current smaller than 50% of the nominal current of square or sum control	
2	Warning	Temperature sensor not in steady state	Temperature sensor has not yet found the actual chip temperature	
3	Error	VDDIO voltage low	Voltage at VDDIO below threshold set by VDDIOSEL as defined in Elec. Char. P05	
4	Error	Interpolator error	Sinusoidal (analog) interpolator position is not within ±22.5°e of the digital interpolator position	
5		Reserved		
6	Error	ABZ-Interface not ready	The ABZ-Interface has not yet reached the current position.	

Bit	Type	Title	Details	Resolution
			Relevant at startup only. This error is not latched and automatically cleared when the position is reached.	
7	Error	UVW-Interface not ready	The UVW-Interface has not yet reached the current position. Relevant at startup only. This error is not latched and automatically cleared when the position is reached.	
8	Error	Position filter: alpha overflow	Acceleration register alpha exceeds an internally defined limit resulting in new filter startup (≈256 µSec)	
9	Error	Position filter: omega overflow	Velocity register omega exceeds an internally defined limit resulting in new filter startup (≈256 µSec)	
10	Error	Digital photo-amplifiers not in steady state	Digital photo-amplifiers did not yet startup successfully	
11	Error	PRC synchronization failed	PRC mismatching tolerance set in RAN_TOL exceeded. Most likely the PRC track was not sampled correctly. See chapter ADJUSTMENT DIGITAL for details on setting up the sampling.	
12	Error	Analog adjustment parameter at boundary value	Parameter COS_OFFS, SIN_OFFS, SC_GAINS or SC_PHASES reached minimum or maximum value, or dynamic part of one of these registers is unexpected high (±12.5% of the range)	
13	Error	Digital adjustment parameter at boundary value	Parameter AI_PHASES or AI_SCALES reached minimum or maximum value, or dynamic part of one of these registers is unexpected high (±12.5% of the range)	
14	Warning	Temperature limit 1	Chip temperature exceeds / falls below temperature limit defined by TEMP_L_1 and TEMP_LT_1	

Bit	Type	Title	Details	Resolution
15	Warning	Temperature limit 2	Chip temperature exceeds / falls below temperature limit defined by TEMP_L_2 and TEMP_LT_2	
16-19	Error	Multiturn error bit(s) read from slave(s)	Mapping of multiturn error bit(s) as set and transmitted by multiturn slave(s). Error bits of slaves 0..2 are individually mapped to diagnosis bits 16..18, while error bits of slaves 3..7 are reduced (OR-operation) to diagnosis bit 19.	
20	Error	Multiturn warning bit(s) read from slave(s)	Mapping of multiturn warning bit(s) as set and transmitted by multiturn slave(s). Warning bits of all slaves are reduced (OR-operation) to this single diagnosis bit	
21	Error	Multiturn position comparison failed	Mismatch of internally counted revolution counter (RC) with external (synchronized) RC from multiturn interface	
22	Error	Pin ADA stuck-at-0	SSI data line pin (ADA) is stuck-at-0 → No communication is possible	
23	Error	Pin ADA stuck-at-1	SSI data line pin (ADA) is stuck-at-1 → No communication is possible	
24	Error	System startup in progress	System is starting up and not yet ready	
25	Error	EEPROM CRC error	At least one CRC in the EEPROM is invalid (or EEPROM communication failed)	
26	Error	Pin GPIO(0) as error input	Pin GPIO(0) is used as external error input. Error is reported if GPIO(0)≠ GPIO0_DIAG	
27	Error	Pin GPIO(1) as error input	Pin GPIO(1) is used as external error input. Error is reported if GPIO(1)≠ GPIO1_DIAG	
28	Error	System startup aborted due to timeout	Timeout reached while waiting for functional block to become ready at system startup	

Bit	Type	Title	Details	Resolution
29-31	Warning	User available	User-defined diagnosis register bits. Can be set and reset by any serial interface	

Sensor ID=6: Panasonic

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Panasonic General Error	Sensor outputs error indication	For details read out **W#[4]** – display error code. And compare with below table.
7	Warning	Panasonic General Warning	Sensor outputs warning indication	For details read out **W#[5]** – display warning code. And compare with below table.
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
9	Error	Encoder ID error	The encoder id doesn't exist or isn't consistent with the assigned resolution.	1. Set the correct sensor resolution according to the sensor type. 2. Check if this sensor is supported.
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=6: Panasonic cont.

W#[4] / W#[5] – Sensor's Extended Status Errors/Warnings code

Bit	Type	Title	Details	Resolution
0	Error	Over Speed	The encoder shaft revolves at a speed exceeding the value specified for Power-OFF Mode.	Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands
1	Error	Full Absolute - Status	The main Power switch is turned ON while the encoder shaft is revolving at more than 100r/min.	Set encoder shaft revolution speed less than 100r/min and wait until the error is automatically reset.
2	Error	Count error	Single turn data is shifted due to operation error or failure when the main Power switch is ON	Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands. Power switch OFF/ON
3	Error	Counter overflow	The multiple revolution counter overflowed	Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands
4	-	Reserved		
5	Error	Multiple revolution error	Fault in multiple revolution data occurred when the main Power switch was turned ON	Return to origin. Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands.
6	Error	System down	The voltage of the encoder's built-in capacitor became less than 2.5V while the main Power switch was OFF	"Reset multiple revolution data and errors" (**T#[2]** =1). Check or replacement of external battery power supply is required.
7	Warning	Battery alarm	The external battery power supply voltage became less than 3.1V	Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands. Check or replacement of external battery power supply is required.

Sensor ID=7: Panasonic Incremental

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Incremental Panasonic General Error	Sensor outputs error indication	For details read out **W#[4]** – display error code. And compare with below table.
7		Reserved		
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
9	Error	Encoder ID error	The encoder id doesn't exist or isn't consistent with the assigned resolution.	1. Set the correct sensor resolution according to the sensor type. 2. Check if this sensor is supported.
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection 2. Check sensor configuration parameters
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again 2. Check Halls connection

Sensor ID=7: Panasonic Incremental cont.

W#[4] – Sensor's Extended Status Errors code

Bit	Type	Title	Details	Resolution
0	Error	INIE error	Encoder internal initialization error	Execute either "Reset errors" (**T#[3]** =1) or Power switch OFF/ON
1	Error	Count error 1	Position data per revolution ABSA is shifted.	

Bit	Type	Title	Details	Resolution
2		Count error 2	Edges Number of A and B is 13 or lower between CS signal edges.	Execute either "Reset errors" (**T#[3]** =1) or Power switch OFF/ON
3	Error	Z signal error	Generation of internal Z signal is stopped. Applicable if PS logic turned from "1" to "0".	Execute either "Reset errors" (**T#[3]** =1) or Power switch OFF/ON
4	Error	CS Signal error	Logic value "1" if the logics of CS1,CS2 and CS3 are all "1" or all "0".	Execute either "Reset errors" (**T#[3]** =1) or Power switch OFF/ON

Sensor ID=8: Sine Virtual Sensor

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	Run commutation wizard again. Check Halls connection

Sensor ID=9: EnDAT

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	EnDAT error message 1	General sensor error	For details read out EnDAT sensor errors, use the following commands: 1. **W#[4]** – display errors * See detailed description below
1	Error	EnDAT error message 2	General sensor error	For details read out EnDAT sensor errors, use the following commands: 1. **W#[4]** – display errors * See detailed description below
2	Warning	EnDAT WRN bit	General sensor warning	To obtain a **warning indication** from Endat 2.1/.2, it is required to set **bit 14 in S#[2]** command. For details read out EnDAT sensor warnings, use the following commands:

Bit	Type	Title	Details	Resolution
				1. **W#[5]** – display warnings * See detailed description below
3	Warning	Additional info not supported	The additional info requested by the user isn't supported by this sensor.	
4	Error	Encoder doesn't support EnDAT 2.2	The sensor supports EnDAT 2.1 only	Select to use EnDAT 2.1 mode commands in sensor settings
5	Error	Incorrect resolution setting	The defined resolution doesn't match the setting in sensor	Set the correct sensor resolution parameters.
6	Warning	Additional Info Error	EnDAT Additional Information 1 CRC	
7	Warning	Additional Info Error	EnDAT Additional Information 2 CRC or bad command acknowledge	
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
9		Reserved		
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
12 - 15		Reserved		

Sensor ID=9: EnDAT cont.

W#[4] – Sensor's Extended Status Error Code

This register displays two indications of errors read from the -sensor - 1st in the lower two least significant 16 bits (bits 0-15) bytes and 2nd in higher two significant 16 bits (16-31) bytes.**1st two Bytes indicates the following errors:**

Bit	Type	Title	Details	Resolution
0	Error	Light source	Light source defective	
1	Error	Signal amplitude	Signal amplitude too low	
2	Error	Position	Erroneous position value	
3	Error	Over-voltage		
4	Error	Under-voltage		
5	Error	Overcurrent		
6	Error	Battery		Replace battery
7		Reserved		
8	Error	S System	Indicates that a singleturn related internal error occurred during the initialization phase of the encoder	Another power-up cycle should be tried. If this also fails, a hardware problem of the encoder is probable.
9		Reserved		
10	Error	M Pos 1		
11		Reserved		
12	Error	M System	Indicates that the multiturn scanning or the multiturn position.	
13	Error	M ALL Power Down	Indicates that both buffer battery voltage (UBAT) and main supply voltage (UP) have dropped below the specified limits	
14	Error	M Overflow/Underflow	Indicates that the specified (multiturn) counting range of the encoder has been exceeded	
15		Reserved		

2nd two Bytes indicates the following errors:

Bit	Type	Title	Details	Resolution
16	Error	S System	Indicates that a singleturn related internal error occurred during the initialization phase of the encoder	Another power-up cycle should be tried. If this also fails, a hardware problem of the encoder is probable.
17	Error	S All Power Down		
18	Error	M Pos1	The M Pos 1 error message indicates that the multiturn scanning or the multiturn position processing failed during the operation of the encoder.	
19	Error	M Pos2		
20	Error	M System	Indicates that the multiturn scanning or the multiturn position.	Another power-up cycle should be tried. If this also fails, a hardware problem of the encoder is probable. Additionally, the consistency of the revolution counter value has to be verified. If this is not possible, referencing of the axis has to take place
21	Error	M All Power Down	Indicates that both buffer battery voltage (UBAT) and main supply voltage (UP) have dropped below the specified limits	
22	Error	M Overflow/Underflow	Indicates that the specified (multiturn) counting range of the encoder has been exceeded	
23	Error	M Battery	The M Battery warning indicates either that external backup battery is not connected or that battery supply voltage (UBAT) dropped below the specified voltage limit for replacement (refer to	The external buffer battery must be replaced / connected. The M Battery warning has to be cleared after the exchange of the buffer battery

Bit	Type	Title	Details	Resolution
			product information D753512). Unless an M ALL Power Down error message occurs, the encoder remains fully functional (including position values).	
24	Error	Light source	Light source defective	
25	Error	Signal amplitude	Signal amplitude too low	
26	Error	S Pos1	Erroneous position value	
27	Error	Overvoltage		
28	Error	Undervoltage		
29	Error	Overcurrent		
30	Error	Temperature exceeded		
31	Error	S Pos2		

For example: W#[4] = 0x00800004 => 1st error code = 0x04 -> Bit 2 is high, "Position Error",

2nd error code = 0x80 -> Bit 31 is high, "S Pos 2" error.

Sensor ID=9: EnDAT cont.

W#[5] – Sensor's Extended Status Warning Code

Bit	Type	Title	Details	Resolution
0	Warning	Frequency collision		
1	Warning	Temperature exceeded		
2	Warning	Light source control		
3	Warning	Battery charge	Indicates either that external backup battery is not connected or that battery supply voltage (UBAT) dropped below the specified voltage limit	
4	Warning	Reference points		
5	Warning	Cyclic mode		
6	Warning	Limit position		

Bit	Type	Title	Details	Resolution
7	Warning	Readiness		
8	Warning	Diagnostics		
9-15		Reserved		

Sensor ID=10: Tamagawa

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Tamagawa General Error	Sensor outputs error indication	For details read out **W#[4]** – display error code. And compare with below table.
7	Warning	Tamagawa General Warning	Sensor outputs warning indication	For details read out **W#[5]** – display warning code. And compare with below table.
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
9	Error	Encoder ID error	The encoder id doesn't exist or isn't consistent with the assigned resolution.	1. Set the correct sensor resolution according to the sensor type. 2. Check if this sensor is supported.
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=10: Tamagawa cont.

W#[4] / W#[5] – Sensor's Detailed Error / Warning code

Bit	Type	Title	Details	Resolution
0	Error	Over Speed OS (Latched)	During the external battery drive after main power supply is turned off, logic "1" is generated when the shaft of Smart Abs is rotated over the specified speed of the Power-off mode in Paragraph 5.4.2 "Electrical Specification for Multi-tum Signal". After main power supply is turned on, it can be transmitted outside. But this flag should be its only purpose, as it may not detect in some cases.	Reset error.
1	Error	Full Absolute-Status FS (Non-latched)	Logic "1" is transmitted when main power supply is turned on while the shaft of Smart Abs is rotated at 100 min-1 or more. The accuracy of one revolution data is 5 bits while logic "1" is transmitted. When one revolution data is switched to the resolution of 17 bits, the flag is automatically released.	Set the rotational speed to slow down to less than 100 min-1 and wait until the flag is automatically released.
2	Error	Count Error CE	Logic "1" is transmitted in the case I, when one revolution data deviates by any malfunction or defect at main power-on. When the shaft of Smart Abs is rotated at 100 min-1 or more, an error is detected every 45°. In mechanical angle. The flag is automatically released at every 45° when the deviation of one revolution data is reduced to less than ±22.5° (typ.).	Immediate stop of servo system. Error will be automatically reset. Turn the power supply off and on.
3	Error	Counter Overflow OF (Latched)	Logic "I" is transmitted when the multi-tum counter is overflowed. In case where it is detected, it can be transmitted outside. The flag detected once is held until reset in spite of main power-on/off and counting value, but the multi-tum counter continues to operate as a cyclic counter of o- 65,535. When a Battery error occurs, Counter overflow is normally operated by resetting the multi-tum data.	Reset error
4			Undefined	
5	Error	Multi-turn error, ME (Latched during main power-on)	Logic "1" is transmitted when a bit-jump in multi-turn signal occurs during main power-on. During power-off it is not detected.	Return to the origin position. Reset error.

			Bit jumping check will be executed every 12.8µs.	
6	Error	Battery Error BE (Latched)	Logic "1" is generated when the external battery voltage is 2.5 ±0.2 V or less, and it can be transmitted after main power-on. When this flag occurs immediately after main power-on, the multi-tum data may be abnormal at the same time.	Reset error and reset multi-turn data. Check or replacement of external battery power supply is required.
7	Warning	Battery Alarm BA (Non-Latched)	In case external battery power supply voltage becomes less than 3.1±0.1V, and it can be transmitted after main power-on. Error is automatically released when the external battery voltage is returned to normal value.	Error status is automatically released. It is necessary to check or replace the external battery.

Sensor ID=11: Pulse & Direction

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Pin A diagnostics	Quad pin A differential line error	Check encoder Physical connection
1	Error	Pin B diagnostics	Quad pin B differential line error	Check encoder Physical connection
2-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=13: EnDAT 3.0

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	EnDAT 3 error message	General sensor error	For details read out **W#[4]** – display error code. And compare with below table.
1	Warning	EnDAT 3 LPF Warning	Undefined command	Check command parameters and try again
2	Warning	EnDAT 3 Invalid Data List	Unexpected data list number	Current command has failed, try to perform again

Bit	Type	Title	Details	Resolution
3	Warning	EnDAT 3 Specific Err BG Request failed	Error while reading specific encoder errors	Discard command results
4	Warning	EnDAT 3 Specific Warning BG Request failed	Error while reading specific encoder warnings	Discard command results
5	Warning	EnDAT 3 Waiting for BG Response	Command is performing	Wait for command to done
6		Reserved		
7	Warning	EnDAT 3 warning message	General Sensor Warning	For details read out **W#[4]** – display warning code. And compare with below table.
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
9		Reserved		
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
12-15		Reserved		

Sensor ID=13: EnDAT 3.0 cont.

W#[4] – Sensor's Extended Status Error Code

Bit	Type	Title	Details	Resolution
0	Error	Generation of data failed or encoder system failure	the standard error Message for serious failures. All measured values of the encoder are to be considered invalid.	Discard received measured values

Bit	Type	Title	Details	Resolution
1	Error	ST position faulty	The singleturn of POS1, POS2, and POS_ABS is to be considered invalid.	Discard Singleturn
2	Error	MT position faulty	The multiturn of POS1, POS2, and POS_ABS is to be considered invalid.	Discard Multiturn part
3 - 7		Reserved		
8	Error	Permissible ambient conditions exceeded	The encoder has detected that permissible ambient conditions, such as the temperature, have been Exceeded.	Ensure that the ambient conditions are maintained (e.g. temperature). Return encoder for servicing.
9	Error	Permissible electrical operating conditions exceeded (current and/or voltage)	The encoder has detected that permissible electrical operating limits have been exceeded (too low or too high).	Ensure that the electrical operating conditions are maintained. Return encoder for servicing.
10 - 11	Error	Reserved		
12	Error	Encoder-specific error message 0	Indication of the message number of the encoder-specific error message.	Send command **T#[7] = 2** and read the results from **WS[110]**
13	Error	Encoder-specific error message 1		
14	Error	Encoder-specific error message 2		
15	Error	Encoder-specific error message 3		

Sensor ID=13: EnDAT 3.0 cont.

W#[5] – Sensor's Extended Status Warning Code

Bit	Type	Title	Details	Resolution
0	Warning	Encoder status / Maintenance	Servicing of encoder is recommended	Exchange or perform maintenance (depending on the encoder model) at an appropriate time
1 – 7		Reserved		

Bit	Type	Title	Details	Resolution
8	Warning	Approaching limit of permissible ambient conditions	The encoder has detected that permissible ambient conditions, such as temperature, could soon be exceeded.	Ensure that the ambient conditions are maintained (e.g., temperature).
9	Warning	Approaching limit of electrical operating conditions	The encoder has detected that permissible electrical operating limits, such as over-/undervoltage, could soon be exceeded.	Ensure that the electrical operating conditions are maintained, such as the power supply.
10	Warning	Temperature warning threshold exceeded	The specified temperature warning threshold was exceeded	Depends on application
11	Warning	Almost down to minimum battery charge	The battery charge threshold was exceeded	Ensure that the electrical operating conditions are maintained Replace the battery soon
12	Warning	Encoder-specific warning message 0	Indication of the message number of the encoder-specific warning message.	Please send command **T#[7] = 3** and read the results from **WS[110]**
13	Warning	Encoder-specific warning message 1		
14	Warning	Encoder-specific warning message 2		
15	Warning	Encoder-specific warning message 3		

Sensor ID=15: Copy Main Profile

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=16: Analog Input

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=18: SSI

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	SSI error bit	General error bit	Check error bit cause in the sensor datasheet
1-6		Reserved		
7	Warning	SSI warning bit	General warning bit	Check warning bit cause in the sensor datasheet
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
12-30		Reserved		

Bit	Type	Title	Details	Resolution
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=19: Yaskawa

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Yaskawa General Error	Sensor outputs error indication	For details read out **W#[4]** – display error code. And compare with below table.
7	Warning	Yaskawa General Warning	Sensor outputs warning indication	For details read out **W#[5]** – display warning code. And compare with below table.
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
9	Error	Encoder ID error	The encoder id doesn't exist or isn't consistent with the assigned resolution.	1. Set the correct sensor resolution according to the sensor type. 2. Check if this sensor is supported.
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=19: Yaskawa cont.

W#[4] / W#[5] – Sensor's Extended Error / Warning code

Bit	Type	Title	Details	Resolution
0	Error	Backup Alarm	Backup power is down, internal data has been lost	Reset command
1	Error	Encoder Error	Error in encoder	Reset command
2	Warning	Battery Warning	Battery voltage drop	
3	Error	Absolute Error	Possible error in position data.	Re-feed power
4	Error	Overspeed	Speed at power feed too high.	Re-feed power
5	Error	Overheat	Temperature inside encoder is too high	
6	Warning	Reset Complete	Reset the encoder with a RESET command	Re-feed power

Sensor ID=20: Gantry Sensor

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-31		No Specific Errors		

Sensor ID=21: Serial Exclusive

W#[2] / W#[3] – Socket's Warning / Error Indication

Master In Three Slaves Configuration

(**GG[1]**.5-7 == 0/1/5) && (**GG[1]**.1 == 1)

Bit	Type	Title	Details	Resolution
0	Error	Serial Exclusive sensor error	Communication channel 1: cycle not completed	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 1 physical connection

Bit	Type	Title	Details	Resolution
1	Error	Serial Exclusive sensor error	Communication channel 1: CRC error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 1 physical connection
2	Error	Serial Exclusive sensor error	Communication channel 2: cycle not completed	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 2 physical connection
3	Error	Serial Exclusive sensor error	Communication channel 2: CRC error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 2 physical connection
4	Error	Serial Exclusive sensor error	Communication channel 3: cycle not completed	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 3 physical connection
5	Error	Serial Exclusive sensor error	Communication channel 3: CRC error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 3 physical connection
6	Error	Serial Exclusive sensor error	Communication channel 1: sync error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 1 physical connection

Bit	Type	Title	Details	Resolution
7	Error	Serial Exclusive sensor error	Communication channel 2: sync error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 2 physical connection
8	Error	Serial Exclusive sensor error	Communication channel 3: sync error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check channel 3 physical connection
9-11		Reserved		
12		N/A		
13-331		Reserved		

Master in single slave configuration (Gantry)

(**GG[1]**.5-7 == 2/3/4/6/7) && (**GG[1]**.1 == 1)

Bit	Type	Title	Details	Resolution
0	Error	Serial Exclusive sensor error	Communication cycle not completed	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check physical connection
1	Error	Serial Exclusive sensor error	Communication CRC error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check physical connection
2-5		Reserved		
6	Error	Serial Exclusive sensor error	Communication sync error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check physical connection
7-11		Reserved		
12		N/A		
13-31		Reserved		

Slave

(**GG[1]**.5-7 == 2/3/4/6/7) && (**GG[1]**.1 == 0)

Bit	Type	Title	Details	Resolution
0	Error	Serial Exclusive sensor error	Communication cycle not completed	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check physical connection

Bit	Type	Title	Details	Resolution
1	Error	Serial Exclusive sensor error	Communication CRC error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check physical connection
2-5		Reserved		
6	Error	Serial Exclusive sensor error	Communication sync error	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check physical connection
7-11		Reserved		
12		TS trigger sync error	Slave TS is not synchronized to master TS	1. Check master and slave Serial Exclusive sensors configuration. 2. Make sure that **TS** of master and slave are equal. 3. Check physical connection
13-31		Reserved		

Sensor ID=22: Resolver

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-4		Reserved		
5	Error	Amplitude Error	Analog signal amplitude exceeded allowed limits	1. Run sensor calibration again. 2. Check sensor
6-13		Reserved		
14	Warning	Amplitude low threshold warning	Analog signal amplitude is below threshold limit	1. Check sensor warning limits. 2. Check sensor

Bit	Type	Title	Details	Resolution
15	Warning	Amplitude high threshold warning	Analog signal amplitude is above threshold limit	1. Check sensor warning limits. 2. Check sensor
16-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=25: Nikon

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Nikon General Error	Sensor outputs error indication and summary bits of the error	For details read out Nikon sensor errors, use the following commands: 1. **t#[7]**=0 – read sensor errors 2. **W#[4]** – display errors
7	Warning	Nikon General Warn	Sensor outputs warning indication and summary bits of the error	For details read out Nikon sensor warnings, use the following commands: 1. **t#[7]**=1 – read sensor warnings 2. **W#[5]** – display errors
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding.
9	-	Reserved		
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection.

Bit	Type	Title	Details	Resolution
				2. Check sensor configuration parameters
12	Warning	Busy+ MemBusy	Logical sum output of the errors: (See bits #7 + #8 in the table below)	Execute "Reset errors" (**T#[3]**=1) to read the detailed error status
13	Warning	Batt	The external battery voltage is 3.0V (TYP) or less.	Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands. Check or replacement of external battery power supply is required.
14	Error	OvSpd+MemErr+OvTemp+OvFlow	Logical sum output of the errors: (See bits #3 + #4 + #9 + #2 in the table below)	Execute "Reset errors" (**T#[3]**=1) to read the detailed error status
15	Error	STErr+PSErr+MTErr+IncErr	Logical sum output of the errors: (See bits #5 + #6 + # in the table below)	Execute "Reset errors" (**T#[3]**=1) to read the detailed error status
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=25: Nikon cont.

W#[4] / W#[5] – Sensor's Extended Error / Warning code

Bit	Type	Title	Details	Resolution
0	Warning	Batt	The external battery voltage is 3.0V (TYP) or less	Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands. Check or replacement of external battery power supply is required.
1	Error	MTErr	The voltage of the built-in backup capacitor or the external battery became 2.45V (TYP) or less.	"Reset multiple revolution data and errors" (**T#[2]** =1) . Check or replacement of external battery power supply is required.

Bit	Type	Title	Details	Resolution
2	Error	OvFlow	The multiple revolution counter overflowed	Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands.
3	Error	OvSpd	Overspeed fault occurred	1. Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands. 2. Power switch OFF/ON.
4	Error	MemErr	EEPROM access error	1. Execute either "Reset errors" (**T#[3]** =1) or "Reset multiple revolution data and errors" (**T#[2]** =1) commands. 2. Power switch OFF/ON.
5	Error	STErr	Single turn position data error	Power switch OFF/ON
6	Error	PSErr	Position data error - Inconsistence in the position data acquired by the MT vs ST modules	Power switch OFF/ON
7	Warning	Busy	Calculation ST position value	Automatically turned off
8	Warning	MemBusy	Sensor's EEPROM is currently being accessed	Automatically turned off
9	Warning	OvTemp	Encoder's temperature has exceeded the threshold	Automatically turned off
10	Error	IncErr	An error in the incremental signal was detected	Power switch OFF/ON
11-15	-	Reserved		

Sensor ID=28: Hiperface

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Hiperface general error	Hiperface sensor reports an error	See **W#[4]** for details (in order to interpret the read error status, refer to the manufacturers manual)
1 - 2		Reserved		
4	Error	Hiperface Init	Hiperface is in init procedure	1. Wait for power up procedure to finish. 2. Check serial data line connection. 3. Check sensor Power supply
5	Error	Amplitude Error	Analog signal amplitude exceeded allowed limits	1. Run sensor calibration again. 2. Check sensor
6	Error	Hiperface communication error	Timeout or faulty checksum received during “get status” command	1. Check sensor. 2. Check serial data line connection
8-13		Reserved		
14	Warning	Amplitude low threshold warning	Analog signal amplitude is below threshold limit	1. Check sensor warning limits. 2. Check sensor
15	Warning	Amplitude high threshold warning	Analog signal amplitude is above threshold limit	1. Check sensor warning limits. 2. Check sensor
16-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=30: Velocity Observer IR

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-30		Reserved		

Bit	Type	Title	Details	Resolution
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=34: Motor Simulation

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=36: BEMF Observer

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	BEMF Observer sensor error	Quality of estimated feedback is low	1. Check sensor parameters. 2. Tune commutation parameter **SC[15]**
1-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=38: Generic Serial IMU/Gyro

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Header value error	Received header value does not match expected value	1. Check header value. 2. Check physical connection
1	Error	Length error	Received data length is different than expected	1. Check length configuration. 2. Check baud rate configuration. 3. Check physical connection
2	Error	Checksum error		1. Check checksum configuration.

Bit	Type	Title	Details	Resolution
				2. Check baud rate configuration. 3. Check physical connection
3	Error	Data counter error		Check physical connection
4-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=39: Mitsubishi

W#[2] / W#[3] – Socket's Warning / Error Indication

Bit	Type	Title	Details	Resolution
0	Error	Mitsubishi General Error	Sensor outputs error indication	For details read out **W#[4]** – display error code. And compare with below table.
7	Warning	Mitsubishi General Warning	Sensor outputs warning indication	For details read out **W#[5]** – display warning code. And compare with below table.
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
9	Error	Encoder ID error	The encoder id doesn't exist or isn't consistent with the assigned resolution.	1. Set the correct sensor resolution according to the sensor type. 2. Check if this sensor is supported.
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection.

Bit	Type	Title	Details	Resolution
				2. Check sensor configuration parameters
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=39: Mitsubishi cont.

W#[4] / W#[5] – Sensor's Extended Error / Warning code

Bit	**Type**	**Title**	**Details**	**Resolution**
0	Error	CPU alarm	Encoder internal data damaged (alarm hold)	Stop operation
1	Error	Angle calculation alarm	Data per revolution calculation error	Stop operation
2	Error	Data alarm	Data per revolution error	Stop operation when received consecutively
3	Error	Encoder thermal alarm	Encoder section hot (115±5°C)	Stop operation
4	Warning	Encoder thermal warning	Encoder section hot (96±5°C)	
5	Warning	Multi-revolution alarm	Multi-revolution count data error	Note at next power ON, multi-revolution may be deviated
6	Error	ABS lost alarm	Multi-revolution backup data damaged (alarm hold)	Stop operation. Clear with request

Sensor ID=40: Acurolink

W#[2] / W#[3] – Socket's and Sensor's warning / error indication

Bit	Type	Title	Details	Resolution
0	Error	Standard Position Error	Sensor Error	For details read out **W#[4]** – display error code. And compare with below table.
1	Error	OEM Read Error	Sensor Command Error	Read **W#[4]** for more details
2	Error	OEM Write Error	Sensor Command Error	Read **W#[4]** for more details
3	Error	Diagnose Services Error	Sensor Error	Read **W#[4]** for more details

Bit	Type	Title	Details	Resolution
4	Error	Operational Mode Change Error	Sensor Error	Read **W#[4]** for more details
5	Error	Encoder Reset Error	Sensor Error	Read **W#[4]** for more details
6	Error	Invalid Operational Mode Error	Sensor Command Error	Read **W#[4]** for more details
7	Error	Invalid Command Error	Sensor Command Error	Read **W#[4]** for more details
8-9		Reserved		
10	Error	Communication Error	Drive Error - Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	Drive Error - The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
12-14		Reserved		
15	Error	Tx MRF to Rx DTF Timeout Error	Sensor Communication Error - Communication or sensor error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
16	Error	Invalid Rx DTF Type Error	Sensor Communication Error	Read W#[4] for more details
17	Error	Invalid Rx DTF Addr/Data Error	Sensor Communication Error	Read W#[4] for more details
18	Error	Invalid Rx DTF Encoder State Error	Sensor Communication Error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
19	Error	Invalid Rx DTF CRC Error	Sensor Communication CRC error	1. Check sensor configuration parameters.

Bit	Type	Title	Details	Resolution
				2. Check sensor physical connection & shielding
20-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=40: Acurolink cont.

W#[4] / W#[5] – Sensor's Extended Error / Warning code

Value	Title	Details	Resolution
1	NOT_EXPECTED_CMD	Invalid Command received (wrong state, wrong parameter etc.)	Encoder in invalid operation state. Send reset command and Re-initialize.
2	MRF_START_SEQUENCE_ERR	Constant synch byte in MRF not received correctly	Data transmission error. Retry transmission.
3	MRF_CMD_UNKNOW	Undefined command byte received	Check implementation or data transmission error. Retry transmission.
4	MRF_DEF_SEQ_BYTE1_ERR	Constant byte 1 in MRF not correctly received	Data transmission error. Retry transmission.
5	MRF_DEF_SEQ_BYTE2_ERR	Constant byte 2 in MRF not correctly received	Data transmission error. Retry transmission.
6	MRF_CRC_ERR	CRC Error nm MRF asynchronous	Data transmission error. Retry transmission.
7	SYNC_ERR	Synchronization error MRF occurred (received too early, too late)	Check master timing sequence.
8	SYNC_SAFETY_CB_ERR	New safety request during already running safety processing received	Check safety request implementation. Wait for current Safety cycle result.
9	SYNC_SAFETY_T2_ERR	Error creating safety Telegram part 2	Internal encoder error. Reset and restart.
10	SYNC_SAFETY_CRC32_ERR	Error creating safety CRC32	Internal encoder error. Reset and restart.

Value	Title	Details	Resolution
11	SYNC_MRF_CRC_ERR	CRC error in MRF synchronous	Data transmission error. Retry transmission.
16	SYS_RAM_ERR	Error RAM Test	Exchange encoder
17	SYS_FLASH_ERR	Error ROM Test	Exchange encoder
18	SYS_PROCESS_ERR	Error internal state machine	Internal encoder error. Reset and restart.
19	SYS_UART_BAUDRATE	Invalid baud rate configured	Check configuration.
20	LG_CONF_ERR	Error initializing opto ASIC	Internal encoder error. Reset and Restart.
21	PM_SYNC_ERR	Error initializing multiturn	Internal encoder error. Reset and Restart.
22	PM_INIT_N_ERR	Error initializing Wiegand ASIC	Internal encoder error. Reset and Restart.
23	MRF_UART_RX_FRAME_URXDA	Reception error hardware UART	Data transmission error. Retry transmission.
24	MRF_UART_RX_FRAME_GREATER4BYTE	MRF > 4 Bytes	Data transmission error. Retry transmission.
25	MRF_UART_RX_FRAME_NOT4BYTE	MRF < 4 Bytes	Data transmission error. Retry transmission.
26	MRF_UART_IRQ_ERR	Error interrupt Hardware UART (Framing, Overrun Error)	Data transmission error. Retry transmission.
27	BISS_ACK_ERR		
28	BISS_START_ERR		
29	OFFSET_ERR	Error initializing multiturn	Internal encoder error. Reset and Restart.
31	TRAP_IRQ	Internal CPU error	Internal encoder error. Reset and Restart.
32	SYS_PRESET_ERR	Error at Preset	Internal encoder error. Reset and Restart.
33	BISS_READ_CRC_ERR	BiSS CRC error (Register R/W, only asynchronous)	

Value	Title	Details	Resolution
34	I2C_TIMEOT_ERR	Timeout when reading via I2C	
35	SPI1_COMM_TIMEOUT_ERR	Timeout when reading directly via SPI 1	
36	SPI1_DMA_TIMEOUT_ERR	Timeout when reading via SPI1 / DMA	
37	SPI2_COMM_TIMEOUT_ERR	Timeout when reading directly via SPI 2	
38	SPI2_DMA_TIMEOUT_ERR	Timeout when reading via SPI2 / DMA	
39	DMA_OEM_TIMEOT_ERR	Timeout while reading from the OEM memory via DMA	
40	DMA_UART_ERR	DMA Error writing via UART	
41	T1_OVERFLOW_ERR	Flow in Timer1 Interrupt (cycle) took too long	
42	PROCESS_ERR	Error Handling of Background Processes	
43	WIEGAND_ERR	Error in the Wiegand calibration	
44	ICLG_CRC16	Only relevant for Unit Test Mode	
45	AD_WANDLER_ERR	Error during initialization of the ADC	
46	POWER_FAIL_ERR	Error in the voltage monitoring	
47	SAFETY_SIN2COS2PLAUSI_ERR	Error during $\sin^2 + \cos^2$ monitoring (outside tolerance)	
48	SAFETY_CH2_PLAUSI_ERR		
49	CPU_ERR_SELFTEST	Error CPU self-test	
50	OVERVOLTAGE	Overvoltage detected	

Sensor ID=41: Openlink

W#[2] / W#[3] – Socket's and Sensor's warning / error indication

Bit	Type	Title	Details	Resolution
0	Error	Standard Position Error	Sensor Error	For details read out **W#[4]** – display error code. And compare with below table.
1	Error	OEM Read Error	Sensor Command Error	Read **W#[4]** for more details
2	Error	OEM Write Error	Sensor Command Error	Read **W#[4]** for more details
3	Error	Diagnose Services Error	Sensor Error	Read **W#[4]** for more details
4	Error	Operational Mode Change Error	Sensor Error	Read **W#[4]** for more details
5	Error	Encoder Reset Error	Sensor Error	Read **W#[4]** for more details
6	Error	Invalid Operational Mode Error	Sensor Command Error	Read **W#[4]** for more details
7	Error	Invalid Command Error	Sensor Command Error	Read **W#[4]** for more details
8-9		Reserved		
10	Error	Communication Error	Drive Error - Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	Drive Error - The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
12-14		Reserved		
15	Error	Tx MRF to Rx DTF Timeout Error	Sensor Communication Error - Communication or sensor error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
16	Error	Invalid Rx DTF Type Error	Sensor Communication Error	Read W#[4] for more details

Bit	Type	Title	Details	Resolution
17	Error	Invalid Rx DTF Addr/Data Error	Sensor Communication Error	Read W#[4] for more details
18	Error	Invalid Rx DTF Encoder State Error	Sensor Communication Error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
19	Error	Invalid Rx DTF CRC Error	Sensor Communication CRC error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
20-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=41: Openlink cont.

W#[4] / W#[5] – Sensor's Extended Error / Warning code

Value	Title	Details	Resolution
1	NOT_EXPECTED_CMD	Invalid Command received (wrong state, wrong parameter etc.)	Encoder in invalid operation state. Send reset command and Re-initialize.
2	MRF_START_SEQUENCE_ERR	Constant synch byte in MRF not received correctly	Data transmission error. Retry transmission.
3	MRF_CMD_UNKNOW	Undefined command byte received	Check implementation or data transmission error. Retry transmission.
4	MRF_DEF_SEQ_BYTE1_ERR	Constant byte 1 in MRF not correctly received	Data transmission error. Retry transmission.
5	MRF_DEF_SEQ_BYTE2_ERR	Constant byte 2 in MRF not correctly received	Data transmission error. Retry transmission.
6	MRF_CRC_ERR	CRC Error nm MRF asynchronous	Data transmission error. Retry transmission.
7	SYNC_ERR	Synchronization error MRF occurred (received too early, too late)	Check master timing sequence.

Value	Title	Details	Resolution
8	SYNC_SAFETY_CB_ERR	New safety request during already running safety processing received	Check safety request implementation. Wait for current Safety cycle result.
9	SYNC_SAFETY_T2_ERR	Error creating safety Telegram part 2	Internal encoder error. Reset and restart.
10	SYNC_SAFETY_CRC32_ERR	Error creating safety CRC32	Internal encoder error. Reset and restart.
11	SYNC_MRF_CRC_ERR	CRC error in MRF synchronous	Data transmission error. Retry transmission.
16	SYS_RAM_ERR	Error RAM Test	Exchange encoder
17	SYS_FLASH_ERR	Error ROM Test	Exchange encoder
18	SYS_PROCESS_ERR	Error internal state machine	Internal encoder error. Reset and restart.
19	SYS_UART_BAUDRATE	Invalid baud rate configured	Check configuration.
20	LG_CONF_ERR	Error initializing opto ASIC	Internal encoder error. Reset and Restart.
21	PM_SYNC_ERR	Error initializing multiturn	Internal encoder error. Reset and Restart.
22	PM_INIT_N_ERR	Error initializing Wiegand ASIC	Internal encoder error. Reset and Restart.
23	MRF_UART_RX_FRAME_URXDA	Reception error hardware UART	Data transmission error. Retry transmission.
24	MRF_UART_RX_FRAME_GREATER4BYTE	MRF > 4 Bytes	Data transmission error. Retry transmission.
25	MRF_UART_RX_FRAME_NOT4BYTE	MRF < 4 Bytes	Data transmission error. Retry transmission.
26	MRF_UART_IRQ_ERR	Error interrupt Hardware UART (Framing, Overrun Error)	Data transmission error. Retry transmission.
27	BISS_ACK_ERR		
28	BISS_START_ERR		

Value	Title	Details	Resolution
29	OFFSET_ERR	Error initializing multiturn	Internal encoder error. Reset and Restart.
31	TRAP_IRQ	Internal CPU error	Internal encoder error. Reset and Restart.
32	SYS_PRESET_ERR	Error at Preset	Internal encoder error. Reset and Restart.
33	BISS_READ_CRC_ERR	BiSS CRC error (Register R/W, only asynchronous)	
34	I2C_TIMEOT_ERR	Timeout when reading via I2C	
35	SPI1_COMM_TIMEOUT_ERR	Timeout when reading directly via SPI 1	
36	SPI1_DMA_TIMEOUT_ERR	Timeout when reading via SPI1 / DMA	
37	SPI2_COMM_TIMEOUT_ERR	Timeout when reading directly via SPI 2	
38	SPI2_DMA_TIMEOUT_ERR	Timeout when reading via SPI2 / DMA	
39	DMA_OEM_TIMEOT_ERR	Timeout while reading from the OEM memory via DMA	
40	DMA_UART_ERR	DMA Error writing via UART	
41	T1_OVERFLOW_ERR	Flow in Timer1 Interrupt (cycle) took too long	
42	PROCESS_ERR	Error Handling of Background Processes	
43	WIEGAND_ERR	Error in the Wiegand calibration	
44	ICLG_CRC16	Only relevant for Unit Test Mode	
45	AD_WANDLER_ERR	Error during initialization of the ADC	
46	POWER_FAIL_ERR	Error in the voltage monitoring	

Value	Title	Details	Resolution
47	SAFETY_SIN2COS2PLAUSI_ERR	Error during $\sin^2 + \cos^2$ monitoring (outside tolerance)	
48	SAFETY_CH2_PLAUSI_ERR		
49	CPU_ERR_SELFTEST	Error CPU self-test	
50	OVERVOLTAGE	Overvoltage detected	

Sensor ID=42: Hiperface DSL

W#[2] / W#[3] – Socket's and Sensor's warning / error indication

Bit	Type	Title	Details	Resolution
0	Error	DSL error bit	General error bit	For details read out **W#[4]** – display error code. And compare with below table.
1-6		Reserved		
7	Warning	DSL warning bit	General warning bit	For details read out **W#[5]** – display warning code. And compare with below table.
8	Error	CRC Error	CRC Communication error	1. Check sensor configuration parameters. 2. Check sensor physical connection & shielding
9		Reserved		
10	Error	Communication Error	Communication cycle didn't complete in the defined communication time	1. Increase serial communication time. 2. Increase clock frequency. 3. Check sensor and connection
11	Error	Sensor is being initialized	The error is indicating the sensor is being initialized.	If the error persists: 1. Check sensor and connection. 2. Check sensor configuration parameters
12-30		Reserved		
31	Error	Halls Error	Wrong Hall state or wrong transition	1. Run commutation wizard again. 2. Check Halls connection

Sensor ID=42: Hiperface DSL cont.

W#[4] – Sensor's Extended Status Error Code

- **W#[4]** command of the DSL sensors contains up to 2 indications of errors in the following structure:
  - Group: indicates from which group the error is derived.
  - Error num: the actual number of the error

  Based on this information, the error detailed from the sensor datasheet.

The following figure decries the **bit structure** of the command. Each error is 12 bits from 0 to 11 and from 12 to 23:

2nd Sensor Error Indication		1st Sensor Error Indication	
(MSB) 23......................20	19 12	11..................................8	7..................................0 (LSB)
2nd Error Group Number	2nd Error Code (bit indication)	1st Error Group Number	1st Error Code (bit indication)

- **W#[4]** command of the DSL sensors contains up to 2 indications of errors in the following structure:
  - Group: indicates from which group the error is derived.
  - Error num: the actual number of the error

  Based on this information the error detailed from the sensor datasheet.
- It is possible to have multiple errors from the same group.
- When the error is cleared, **W#[4]** indications will also be cleared.
- For example: **W#[4]** = 0x0103 => Error Group = 1, Error Code = Bit 1 & Bit 2

The following table was copied from the DSL manufacturer datasheet. Please review the updated datasheet per your relevant sensor and any required troubleshooting and optional correction.

Group	Bit	Title	Details	Resolution
0	7	Position Synchronization Error	Position sensor subsystems could not match their signals due to shocks, hardware faults or too high speed during power-up. Synchronization of the counted position and a new absolute position was not possible.	If this error is accompanied by the following (position synchronization related) critical errors, the system must be placed in a safe state: • Counter Error (ENC_ST0) • Position Cross-check Error (ENC_ST1) Meaning, it becomes a critical severity error. In any other case, it can be handled as a minor severity error and no action required.
	6	Counter Error	Position quadrant counter has detected an invalid sequence of signals Occurrence during	The drive system must be placed in a safe condition. Action: SW reset of the encoder.

Group	Bit	Title	Details	Resolution
			operation leads to a permanent error.	
	5	Position Vector Length Error	The Vector Length of the sampled and measured Sin-Cos signals has exceeded its boundaries. Possible causes: HW or supply faults, mechanical shocks.	When this error occurs, the IP Core advances the fast position in linear fashion, by turning its estimator ON (estimator_ON = 1), until valid values are present again. Dependent upon the application, one or more vector length errors in succession can be tolerated. The tolerable number of errors can be determined from a calculation of the Maximum deviations in each occurring error. Recommendation: monitor (count) the occurrence of this error. If the maximum tolerable number of errors is exceeded, the drive system must be placed in a safe state, meaning a critical severity. If necessary, the safe position can be used as an alternative to position measurement, if this error arises for the fast position. In this case the significantly slower refresh cycle of the safe position must be considered. If the error persists, there is probably a general hardware or mechanical failure. Inform customer service.
	4	Position Tracking Filter Error	The sine/cosine-signals got a big distortion, and the signal tracking (interpolation) got lost. Occurs, when check is enabled, and tracking is out of limit. Not considered safety relevant. Affecting only Safety Channel 1.	No action required. Encoder recovers automatically.
	3	Plausibility Error	Potential cause: faults affecting the sensor, the analog input chain, the excitation-signals. Also possible: sensor- signal distortion due to electrical burst or mechanical shock. The sensor-signal behavior is not as expected: plausibility result is	The drive system must be placed in a safe condition. Action: SW reset of the encoder.

Group	Bit	Title	Details	Resolution
			greater than the allowed threshold. Will lead to a VPOS-error.	
	2	Test Running	Available only for safety encoders. Indicates that a diagnostic test has been requested by application.	Action required as detailed in the safety implementation manual see table 56.
	1	Acceleration Over- flow Error	Two subsequent fast position values were too far apart for valid DSL transmission; too high shaft acceleration or shaft was turned while the link was not active.	No action required. Fault is not permanent, and encoder recovers automatically. This error will likely lead to other errors. Possible causes: mechanical shocks or hardware faults. If the error persists, there is probably a general hardware or mechanical failure.
	0	Protocol Reset Indication	Indicates that the DSL protocol has been reset and a new connection was established. Serves a base indication for the start of the communication (like LINK). If not set after an encoder reset, no status registers can be relied upon!	Action: successful clearing is required. If this indication persists (bit is set), it is likely that the encoder was again reset. Check connection or supply of the encoder. Before the successful clearing of this indication, no other ENC_ST information can be relied upon!
1	4	Position Cross-check Error	Only for safety encoders and HF2DSL coupler. Internal cross-check of safe absolute position has failed. In case of non-safe encoders, this bit is always ‘0’. Potential cause(s): hardware faults; for linear encoders: alignment fault read / measure.	
	3	Multiturn Vector Length Error	Multi-turn sensor synchronization was erroneous. Potential cause(s): encoder reached end of lifetime, hard-ware faults, magnet (clip) faults or gear wear-out. NOTE: A multi-turn synchronization diagnostic cannot reliably detect gear synchronization faults over extended time. After first indication of this fault gear	The drive system must be placed in a safe condition. Action: immediate replacement of encoder

Group	Bit	Title	Details	Resolution
			synchronization can wear out even more and result in undetected false position output.	
	2	Multiturn Sync Error	Multi-turn sensor synchronization was erroneous. Potential cause(s): encoder reached end of lifetime, hard-ware faults, magnet (clip) faults or gear wear-out. NOTE: A multi-turn synchronization diagnostic cannot reliably detect gear synchronization faults over extended time. After first indication of this fault gear synchronization can wear out even more and result in undetected false position output.	The drive system must be placed in a safe condition. Action: immediate replacement of encoder
	1	Multiturn Amplitude Error	Invalid multi-turn sensor signal amplitude. Potential cause(s): magnetic disturbances, multi-turn sensor defects, loss of multi-turn magnets or errors in multiturn parameters.	The drive system must be placed in a safe condition. Action: immediate replacement of encoder
	0	Singleturn Error	Initialization of single-turn sensor was erroneous, or the encoder could not measure absolute single-turn position. This results in an overall invalid encoder position. Potential causes: hardware failures or external shocks.	The drive system must be placed in a safe condition. Action: SW-reset of the encoder
2	6	Internal System Error	After startup, an internal electronic error was identified. (Encoder initialization time-out; FW CRC error; internal transmission faults (SPI) of safety-relevant parameters). Potential cause(s): uC faults, hardware faults; EMC noise injection	The drive system must be placed in a safe condition. Action: SW-reset of the encoder

<table>
<tr><th>Group</th><th>Bit</th><th>Title</th><th>Details</th><th>Resolution</th></tr>
<tr><td rowspan="6"></td><td>5</td><td>Internal Communication Error 2</td><td>During startup an internal communication error has occurred (I2C).</td><td>No action required. If error persists, potential EEPROM hardware failures are present.</td></tr>
<tr><td>4</td><td>Internal Communication Error 1</td><td>During or after startup an internal communication error has occurred (SPI).</td><td>No action required. Encoder recovers automatically. If error persists, potential hardware failures are present.</td></tr>
<tr><td>3</td><td>Standard Parameter Error</td><td rowspan="2">There were errors in the internal encoder EEPROM parameter or diagnosis that could not be rectified.</td><td>The drive system must be placed in a safe condition.
Action: SW-reset of the encoder</td></tr>
<tr><td>2</td><td>Safety Parameter Error</td><td>The drive system must be placed in a safe condition.
Action: SW-reset of the encoder If error persists, potential EEPROM hardware failures are present. Encoder replacement is required!</td></tr>
<tr><td>1</td><td>Safety Parameter Warning</td><td>There were errors found in the safety parameters which were rectified.</td><td>No action required.</td></tr>
<tr><td>0</td><td>Power-on Self-test Conducted</td><td>Only for safety encoders. Mandatory to read this bit at startup for all safety encoders.</td><td>Action: successful clearing is required.
If this indication persists (bit is set), it is likely that the encoder was again reset. Check connection or supply of the encoder. Before the successful clearing of this indication, no other ENC_ST information can be relied upon!</td></tr>
<tr><td rowspan="2">4</td><td>3</td><td>File access error</td><td>Error when accessing a file in EEPROM.
Potential cause: may indicate EMC problems or corrupted file system due to previous power-down during write access or just simply incorrect access parameters.</td><td>Check and retry command.</td></tr>
<tr><td>2</td><td>Resource access error</td><td>Error when accessing an internal resource.
Potential cause: may indicate EMC problems, excessive number of EEPROM write cycles or wrong timing of power cycle vs. EEPROM write access or just simply incorrect access parameters.</td><td>Check and retry command; check resource access sequence.</td></tr>
</table>

Group	Bit	Title	Details	Resolution
	1	Access denied	Access to a resource was denied.	Check and retry command; check access user level.
	0	Invalid access	Invalid argument / command of a resource access. Potential cause: drive firmware fault or internal signal transmission fault (due to EMC) or just simply incorrect access parameters.	Check and retry command; check drive firmware.
7	0 - 7	User-defined Warning	Depending on user customization (see chapter 9.8.5).	Depending on configuration.

Sensor ID=42: Hiperface DSL cont.

W#[5] – Sensor's Extended Status Warning Code

- W#[5] command of the DSL sensors contains 1 indication of warning in the following structure:
  - Group: indicates which group the error derived from (Group number 3)
  - Warning num: the actual number of the warning

  Based on this information, the error detailed from the sensor datasheet.

The following figure decries the **bit structure** of the command. The warning is 12 bits from 0 to 7 – warning code, and 8 to 11 – warning group (equals 3):

Warning Indication	
(MSB) 11………………………….8	7…………………………..0 (LSB)
1st Error Group Number	1st Error Code (bit indication)

- It is possible to have multiple warnings.
- When the warning is cleared, **W#[5]** indications will also be cleared.

Group	Bit	Title	Details	Resolution
3	6	Internal Monitoring Error	During monitoring of an internal electronic, an error was identified.	No action required. Encoder might recover automatically. If error persists even after clearing, potential hardware failures are present and immediate replacement of the encoder is required!
	5	Counter Overflow	User counter has overrun.	Depending on counter use case.

Group	Bit	Title	Details	Resolution
	4	Critical Acceleration	Acceleration was out of specification.	No action required; check application, reduce acceleration or check for shock.
	3	Critical Rotation Speed	Rotation speed was out of specification.	No action required. Check application for higher speeds than specified. Possibly reduce speed.
	2	Critical Supply Voltage	Supply voltage was out of specification.	No action required. Check encoder power supply conditions. If error persists after ensuring the input power supply, potential hardware failures are present and immediate replacement of the encoder is required.
	1	Critical LED Current	Monitored sensor behavior or code disc position out of specification. Potential cause: encoder sensor near end of lifetime, sensor is polluted, encoder was used outside of its specification.	Action: SW-reset of the encoder Check application for potential pollution (dust/ brake dust). If error persists even with correctly set rotor position, replacement of the encoder is required.
	0	Critical Temperature	Encoder temperature was out of the specification	No action is required. Check application for potential motor/encoder cool down

References

EC[], **EE[]**

Sensor Status Flag Functions and Terms

Sanyo Denki/Nikon Status Flag Function Descriptions

Name	Function	Detection timing	Output	Switching OFF
Busy flag (BUSY)	This flag is turned on during the process to determine a single turn absolute value. To be more specific, if M is sequential scan data is logically incorrect, or if scan data which are detected three times do not match, "1" is output, and all the position data during that period becomes "0" fixed. After the determining process is completed, the flag returns to "0".	During normal operation (when the main power is turned ON)	Non-latch	Automatically turned off (Set the turn speed at 250min-1 or less)
PS error (PSERR)	To monitor the conformance between the "multiple turn calculation block'' and the "single turn calculation block". To be more specific, (I) Multiple turn calculation value calculated by the magnetic encoder; (2) Multiple turn calculation pseudo value calculated by carry/borrow of the single turn absolute value. When comparing (I) and (2), if (difference between (I) and (2)) > (single turn), an alarm is turned on.	During normal operation	Latch	The main power is turned ON again
ST error (STERR)	To monitor the conformance between the "ABS block'' and the ''INC block''. To be more specific, (I) Certain 1 bit of the absolute sensor (2) Certain 1 bit of the shift register in ASIC (generated by the M sequential polynomial equation) When comparing (1) and (2), if the data do not match, an alarm is turned on.	During normal operation	Latch	The main power is turned ON again

Name	Function	Detection timing	Output	Switching OFF
Over speed alarm (OVSPD)	When the turn speed exceeds 7700min-1 an alarm is turned on.	During normal operation	Latch	A command (CDF8 or CDF10) is entered
Battery warning (BW)	When the external battery voltage is 3.2V(TYP) or less, an alarm is turned on.	During normal operation	Non-latch	Automatically turned off
Battery alarm (BA)	When the encoder enters the non -operation state and it is unable to serve as an encoder, an alarm is turned on. To be more specific, the higher voltage between main power or the external battery becomes 2.76V(TYP) or less	During back-up operation	Latch	A command (CDF10) is entered
MEMBUSY flag (MEMBUSY)	This flag shows that access to the EEPROM in the encoder is under way. After the access is completed, the flag returns to "0".	During normal operation (data read, data write)	Non-latch	Automatically turned off (Turned off after a certain period of time passes)
MEM error (MEMERR)	This flag is turned on when an error occurs while accessing the EEPROM in the encoder. To be more specific, an error occurs in cases described below: (I) Response failure in a data write completion signal during data write, or unmatched data (2) Check failure during data read	During normal operation (data read, data write)	Latch	A command (CDF8 or CDF10) is entered or the main power is turned ON again
Temperature alarm (OVTEMP)	When the encoder temperature is 95°C ±4°C or more, an alarm is turned on.	During normal operation	Non-latch	Automatically turned off

Panasonic Status Flag Function Descriptions

<table>
<tr><th rowspan="2">Name</th><th colspan="2" rowspan="2">Function</th><th colspan="2">Battery
Error output when power switch is turned ON (refer to Note below)</th><th rowspan="2">Remedy</th></tr>
<tr><th></th><th></th></tr>
<tr><td>Over Speed
OS (Latched)</td><td colspan="2">In case Encoder shaft revolves at speed exceeding the value specified for Power-OFF Mode when Encoder is driven with battery power supply after the main Power switch is turned OFF, the logic "1" will be generated. External transmission will be enabled when the main Power is turned ON.
However, there are cases in which this detection is failed. Use this error as reference. (Note 9)</td><td>←</td><td>Undefined</td><td>Reset error.</td></tr>
<tr><td>Full Absolute-Status
FS (Non-latched)</td><td colspan="2">In case the main Power switch is turned ON while the Encoder shaft is revolving at more than 100 r/min, the logic "1" will be output.
Accuracy of one revolution data while the logic "1" is output will be 5-bit.
When one revolution data is switched to 17-bit resolution, the error will be automatically reset.</td><td>←</td><td>←</td><td>Set Encoder shaft revolution speed less than 100r/min and wait until the error is automatically reset.</td></tr>
<tr><td rowspan="3">Count Error
CE</td><td colspan="2">In case one revolution data is shifted due to operation error or failure when the main Power switch is ON, the logic "1" will be output by the following I, II OR.</td><td></td><td></td><td>Immediate stop of servo system.</td></tr>
<tr><td>I (Non-latch)</td><td>In case Encoder shaft is revolving at more than 100 r/min, error detection will be conducted at every 45° mechanical angle.
When shift of one revolution data is reduced to the mechanical angle ±22.5° (TYP.), the error will be automatically reset.</td><td>←
←</td><td>←
←</td><td>Error will be automatically reset. Power switch OFF/ON</td></tr>
<tr><td>II (Latch)</td><td>In case Encoder shaft is revolving at less than 100r/min, error detection will be conducted all the time. When shift of one revolution exceeds the</td><td>←</td><td>←</td><td>Reset error.
Power switch OFF/ON</td></tr>
</table>

Name	Function	Battery Error output when power switch is turned ON (refer to Note below)		Remedy
	mechanical angle ± 0.7° (TYP.), the logic "1" will be output.			
Multiple Revolution Error ME (Latched)	In case the abnormalities in multiple revolution data occurs when the main Power switch is turned ON, the logic "1" will be output. (This does not operate when the main Power switch is turned OFF.) Bit jumping check will be executed every 12.8µs.			Origin return. Reset error.
Counter Overflow OF (Latched)	In case the multiple revolution counter overflows, the logic "1" will be output. (However, in case this error is detected when the main Power switch is turned OFF, external output will be enabled when the main Power switch is turned ON.) Once this error is detected, the error will be sustained until it is reset but the multiple revolution counter will continuously operate as cyclic counter for 0 - 65535.	←	Undefined	Reset error.
Battery Alarm BA (Latched)	In case external battery power supply voltage becomes less than 3.1±0.1V (Refer to 4.4 Common Electric Specifications) irrespective of power supply ON/OFF, the logic "1" will be generated and external transmission will be enabled when the main Power switch is turned ON. Once this error is detected, the error will be sustained until it is reset.	←	"1"	Reset error. Check or replacement of external battery power supply is required.
System Down SYD (Latched)	In case voltage of Encoder built-in capacitor becomes less than 2.5±0.2V (Refer to 4.4 Common Electric Specifications). When the main Power switch is turned OFF, the logic "1" will be generated and external transmission will be enabled when the main Power switch is turned ON. When this error occurs, it is possible that multiple revolution data error is also occurring.			Reset error and reset multiple revolution data. Check or replacement of external battery power supply is required.

Note:

If no battery is connected, only Normal Mode functions are available.

Even if external battery is connected in case external battery power supply voltage becomes less than 2.5 - 3.5V(TYP.) and System Down(SYD) occurs, error flags operate similarly as in the case when the external battery is unconnected.

Note 9:

Over Speed Error occurring condition.

If the input shaft revolves exceeding the responding revolution speed, "1" will be output. In case the input shaft revolves exceeding the follow-up speed of the internal circuit of Encoder during Power-OFF operation, the output will be as shown in the following Table 1.

Encoder revolution speed	Over Speed Error
0 - 6000r/min	"0"
6000 - 14000r/min: Calculated value	"0": Detection of multiple revolution data is normal. "1": Detection of multiple revolution data is wrong.
14000r/min or more: Calculated value	Undefined

Table 1 Relations between revolution speed and Over Speed Error

If the revolution speed is 0 - 14000r/min (calculated value) and the Over Speed Error logic is "0", detection of multiple revolution data is normal and multiple revolution data is normal.

If the logic is "1", it is possible that multiple revolution data is shifted and resetting operation will be required. Therefore, the use in the range of 0 - 6000r/min is recommended.

During Power-OFF operation, even if revolution speed is lower than responding revolution speed, the output will be as shown in the following Table b if the angular acceleration exceeds 4000 rad/s^2.

Angular acceleration	Over Speed Error
0 - 4000 rad/s2	"0"
4000 – 28000 rad/s2: Calculated value	"0": Detection of multiple revolution data is normal. "1": Detection of multiple revolution data is wrong.
28000 rad/s^2 or more: Calculated value	Undefined

Table 2 Relations responding revolution speed and Over Speed Error

If the angular acceleration is 0~28000 rad/s^2 (calculated value) if Over Speed Error logic is "0", detection of multiple revolution data is normal and multiple revolution data is normal.

If the logic is "1", it is possible that multiple revolution data is shifted and resetting operation will be required. Therefore, the use in the range of 0~4000 rad/s^2 is recommended.

Tamagawa Status Flag Function Descriptions

Name	Function	Battery Error output depending on connecting battery during main power-on		Remedy
Over Speed OS (Latched)	During the external battery drive after main power supply is turned off, logic "1" is generated when the shaft of Smart Abs is rotated over the specified speed of the Power-off mode in Paragraph 5.4.2 "Electrical Specification for Multi-tum Signal". After main power supply is turned on, it can be transmitted outside. But this flag should be its only purpose, as it may not detect in some cases.	←	Undefined	Reset error.
Full Absolute-Status FS (Non-latched)	Logic “1” is transmitted when main power supply is turned on while the shaft of Smart Abs is rotated at 100 min-1 or more. The accuracy of one revolution data is 5 bits while logic “1” is transmitted. When one revolution data is switched to the resolution of 17 bits, the flag is automatically released.	←	←	Set the rotational speed to slow down to less than 100 min-1, and wait until the flag is automatically released.
Count Error CE	Logic “1” is transmitted in the case I, when one revolution data deviates by any malfunction or defect at main power-on.			Immediate stop of servo system.
	I (Non-latch) — When the shaft of Smart Abs is rotated at 100 min-1 or more, an error is detected every 45°. In mechanical angle. The flag is automatically released at every 45° when the deviation of one revolution data is reduced to less than ±22.5° (typ.).	←	←	Error will be automatically reset.. Turn the power supply off and on.
Multi-tum error, ME (Latched during main power-on)	Logic "1" is transmitted when a bit-jump in multi-turn signal occurs during main power-on. During power-off it is not detected. Bit jumping check will be executed every 12.8µs.	←	←	Return to the origin position. Reset error.

Name	Function	Battery Error output depending on connecting battery during main power-on		Remedy
Counter Overflow OF (Latched)	Logic "I" is transmitted when the multi-tum counter is overflowed. In case where it is detected, it can be transmitted outside. The flag detected once is held until reset in spite of main power-on/off and counting value, but the multi-tum counter continues to operate as a cyclic counter of o- 65,535. When a Battery error occurs, Counter overflow is normally operated by resetting the multi-tum data.	←	Undefined	Reset error.
Battery Alarm BA (Non-Latched)	In case external battery power supply voltage becomes less than 3.1±0.1V, and it can be transmitted after main power-on. Error is automatically released when the external battery voltage is returned to normal value.	←	←	Error status is automatically released. It is necessary to check or replace the external battery.
Battery Error BE (Latched)	Logic "1" is generated when the external battery voltage is 2.5 ±0.2 V or less, and it can be transmitted after main power-on. When this flag occurs immediately after main power-on, the multi-tum data may be abnormal at the same time.	←	←	Reset error and reset multi-turn data. Check or replacement of external battery power supply is required.

Note:
Even if external battery is connected in case external battery power supply voltage becomes less than 2.5 - 3.5V(TYP.) and Battery Error(BE) occurs, error flags operate in similarly as in the case when the external battery is unconnected.

Dual Use Compliance

In order to comply with the EU Dual Use Regulation 428/2009 & the US Dual-use regulation EAR ECCN# 3A225 the electric frequency should be limited to 599 Hz.

As from this version, drives with Part # suffix "Q" complies to this requirement by internally limiting the Max Electrical Freq. to 599 Hz.

Any previous firmware loaded to a "Q" drive will fail to enable the servo motor.

Drive Communication & Interpreter

Interpretation of commands as described in this document is derived from the following communication channels:

- RS232
- USB
- Ethernet: UDP or TCP\IP
- EtherCAT: EoE

These channels are considered as functional channels where the interpretation and activation of the command have no time restriction and are performed in the background task of the drive. The drive supports real time channels via fieldbus communication that include EtherCAT CoE and CANopen. The method of addressing these channels is not in the scope of this document.

The commands described in this document can be also used via the User Program of the drive. The User program allows the user to download any desired code to the drive and run the code via a command or automatically after power up.

The drive interpreter evaluate string which is a sequence of characters, terminated by semicolon ';'. The interpreter will process the command once the terminator is detected.

The Platinum drive maintain the legacy format of the 2 letters command with enhancement that the format of [letter][number] can also be used. The numbers are from 1 to 9 and the letters from A to Z. The first command is **A1** and the last command is **ZZ**. The interpreter uses also prefix to the command. The prefix are described in the following chapter.

The maximum length of the line can be 1024 characters.

The characters that are sent to the drive are echoed back in any of the channels. **EO** command can be used to deactivate the echo.

The interpreter can process basic mathematics operation allowing the user to manipulate commands for example:

Command Line	Result	Remarks
3+4	7	
UI[1]=10; UI[1]-3	7	UI[1] is set to 10, and 3 is then subtracted.
(3.2+4)/2	3.600000	

Command Format: Scalar vs Array

Two types of commands are used:

Scalar commands: **AC**, **BG** etc.

Array commands: **CA[1]**, **KP[3]** etc.

The Platinum drive handles all commands as an array where sub-index 0 of the array is the number of last index of the array.

For compatibility the legacy scalar commands can also be addressed as array with first index for example:

VR & **VR[1]** result in the same reply from the drive.

Command & Data types

The term “command” include the following interpretation options:

Commands: *Write only* commands that performs and action such as “BG” to start motion or “ST” to stop motion.

Parameters: Commands that includes data and have the following attribute:

- read only (e.g. “VR” for firmware version),
- write only (e.g. **TW[16]** to manually control the drive LEDs) or
- read\write (e.g. **UI[1]** general user parameter that can be freely used) that sets or get a parameter from the drive.

The parameters can be volatile (e.g. “PA” for Position Absolute reference) or non-volatile (“AC” profile acceleration).

Parameters have a defined data type that is described in this document. The commands have no data type.

The following data types are supported:

Type	Bytes	Range
char	1	-128 … +127
Unsigned char	1	0 … 255
short	2	-32768 … +32767
unsigned short	2	0 … 65535
long	4	-2,147,483,648 … +2,147,483,647
unsigned long	4	0 … 4,294,967,295
float	4	-1e37 … 1e37
double	8	-1.0e307 … +1.0e307
long long	8	-9,223,372,036,854,775,807 … 9223372036854775807

Bit-field format

Parameters can include bit-field forma. The bit-field means that each bit or group of bits within the same parameters have an individual meaning (indication). In this format there is no meaning to the decimal presentation of the number. Typically, these types of variables are defined as unsigned type. It is useful to receive the values using the hex decimal format to simplify the visual interpretation. The attribute of the command in this document indicates a bit-field type.

Hexadecimal format

For integer values, typically bit-field type, it is useful to request the drive to transmit the value in a hexadecimal format. Setting **HX**=1will convert any integer to Hex value with the 0x prefix. Setting an integer parameter can be done in decimal format or by using the prefix '0x' for a hexadecimal format. There is no need to set **HX=1** for the "set" of a parameter.

Note that the interpreter reply for mathematical operation when Hex format is active is always in long long format. See the example below.

The following examples are a reply from the drive for a "get" request:

Command	Dec (HX=0)	Hex (HX=1)
SR	2105351	0x00202007
UI[1]=0x1234	4660	0x1234
UI[1]+0xA	4670	0x000000000000123E (math operation is long long)

Mathematic and logical operation

The following math and logic operations are available including a simple example:

Operator/ Description	Symbol	Examples
Arithmetic addition	+	4+5=9 3.45+2.78=6.23
Arithmetic subtraction	-	3.45-2.78=0.67
Arithmetic multiplication	*	UI[1]=UI[1]*2 multiple UI[]1 by 2 5*4=20 1.5*2=3.0
Arithmetic division	/	20/4=5 3/1.5=2.0
Remainder after the division of two integers	%	20%4=0 23%4=3
Bitwise NOT	~	~3 is 0x00000000fffffffc, (if HX=1, note: -4.)
Bitwise OR	|	UI[1]=0x2|0x5 is equivalent to UI[1]=7.
Bitwise AND	&	UI[1]=0x7&0x3 is equivalent to UI[1]=3.
Logical equality	==	If UI[1] = 3 and UI[2] = 3, then UI[1]==UI[2] yields 1. If UI[1] = 3 and UI[2] = 5, then UI[1] == UI[2] yields 0.
Logical inequality	!=	If UI[1] = 3 and UI[2] = 3, then UI[1]!= UI[2] yields 0. If UI[1] = 3 and UI[2] = 5, then UI[1]!= UI[2] yields 1.

Operator/ Description	Symbol	Examples
Logical greater than	>	If UI[1] = 3 and UI[2] = 3, then UI[1]>UI[2] yields 0. If UI[1] = 3 and UI[2] = 2, then UI[1]>UI[2] yields 1. If UI[1] = 1 and UI[2] = 2, then UI[1]>UI[2] yields 0.
Logical greater than or equal to	>=	If UI[1] = 3 and UI[2] = 3, then UI[1]>=UI[2] yields 1. If UI[1] = 3 and UI[2] = 2, then UI[1]>=UI[2] yields 1. If UI[1] = 1 and UI[2] = 2, then UI[1]>=UI[2] yields 0.
Logical less than	<	If UI[1] = 3 and UI[2] = 3, then UI[1]<UI[2] yields 0. If UI[1]=3 and UI[2]=2, then UI[1]<UI[2] yields 0. If UI[1] = 1 and UI[2] = 2, then UI[1]<UI[2] yields 1.
Logical less than or equal to	<=	If UI[1] = 3 and UI[2] = 3, then UI[1]<=UI[2] yields 1. If UI[1] = 3 and UI[2] = 2, then UI[1]<=UI[2] yields 0. If UI[1] = 1 and UI[2] = 2, then UI[1]<=UI[2] yields 1.
Logical AND: The result is 1 if both arguments are non-zero, and 0 if one is zero.*	&&	1&&5 yields 1. 0&&2 yields 0.
Logical OR The result is 1 if one argument is non-zero, and 0 if both are zero.*	||	1||0 yields 1. 0||0 yields 0.
Logical NOT: The result is 1 if the argument is zero; otherwise, it is 0.*	!	!4 yields 0. !0 yields 1.
Unary minus: The result is negative if the argument is positive, and vice versa.*	-	-4.5 yields -4.5. -4 yields -4. (-4) yields -4. -5+5 yields 0.
Bitwise left shift: Shifts the first operand to the left by the number of positions specified in the second operand.*	<<	8<<2 yields 32.

Operator/ Description	Symbol	Examples
Bitwise right shift: Shifts the first operand to the right by number of positions specified in the second operand.*	>>	8>>2 yields 2.

* The arguments are truncated to integers before evaluation.

Further Note:

When a float is used than the result will be casted to float.

Division of integer will be casted to integer (floor operation).

Scientific notation can be used for float presentation and settings:

Example 1.e1 is 10.0 and 1e-2 is 0.01

The **DP** command can be used to determine the presented number precision. By default, the value is -1 using the max available range. The value of float number will rounded to the nearest integer – see example below. Note that internal the value is not modified and the **DP** only effects the presentation.

Examples:

Command	DP=-1	DP=2	DP=3
1.1239E4 * 0.0001	1.123900e0	1.12	1.124 (near integer)

The drive supports build in function operation as follows – note function names are case-sensitive.

Function	Description	Returns
sin	UF[1]=sin(x) x [rad] For example, sin (3.1415 / 2) is 1.00 Complex expression can be used: sin(2*3.1415+(3.1415/2) which is sin (2 * π + π/2) is 1.000	double
cos	UF[1]==cos(x) x [rad] For example, the angle 60° is approx. 1.0472 radians (60 * (2 * π / 360)). The result of cos(1.0472) is 0.500	double
sqrt	y=sqrt(x) Square root of the argument is negative result in “nan” For example, the results for sqrt(16) is 4.00, the result of sqrt(-81) is “nan”.	double
fix	y=fix(x)	long long

Function	Description	Returns
	Truncates the input argument to an integer. For example, the result of fix(3.8) is 3, and the result of fix(-3.8) is –3. **Note:** If the input argument exceeds the long value range, it will be limited to the maximum long value (for positive numbers) or the minimum long value (for negative numbers).	
rnd	y=rnd(x) Truncates the input argument to the nearest integer. For example, the result of rnd(3.8) is 4, the result of rnd(-3.8) is –4, and the result of rnd(3.4) is 3. **Note:** If the input argument exceeds the long value range, it will be limited to the maximum long value (for positive numbers) or the minimum long value (for negative numbers).	long long
sign	y=sign(x) Returns the sign of the input argument: **-1** for negative numbers, **1** for positive numbers, **0** for a zero. For example, the result of sign (-3.8) is -1, the result of sign(3.8) is 1, and the result of sign(0) is 0.	long long
real	y=real(x) Converts an integer to a float. If the argument is a floating-point number, the function does nothing: The result of 5/2 is 2, the result of real(5)/2 is 2.5, and The result of 5/real (2) is 2.5.	double

Assign and retrieve parameters

Assign a parameter is done with the '=':

UI[1]=1000;

AC=UI[1] * 5 ;

Retrieve a value is done by a direct request as follows:

UI[1];

AC;

The return of the parameter will be according to the data type of the parameter. Data types are described in the attribute of each parameter in this document.

Commands Prefix

Enhancement for the interpretation is done via command prefix allowing to extend the functionality of the command. The prefix are set of characters that are sent as part of the command as follows:

[Prefix][.][Command].

The rules of the assignment or retrieve commands with a prefix are the same as a command w/o a prefix. The interpreter analyze the prefix and do required the functionality accordingly.

The Prefix are:

- **AX** to address a specific axis
- **GR** to address a predefined group
- **OJ** to address an object in decimal format
- **OH** to address an object in Hex-Binary format.

Refer to the section **Interpreter Prefixes**.

Elmo commands alias Objects

Objects are commands that are used in fieldbus protocol and are interpreted via an Object dictionary. Each object gets the relevant functionality from the dictionary. Each object can have sub-index of up to 254 entries. In this situations that the object is from array (or struct.) type, sub index 0 will inform the number of the last index in the array.

For example, the command to set the protocol profile acceleration is done via object 0x6083 sub index 0 (0x6083.0). More details about objects are described in the CANopen and EtherCAT DS-402 & DS-301 manual.

The drive uses set of objects to allow the fieldbus to address Elmo's legacy commands. The set of objects that is used for this purpose starts from object 0x3000 until 0x5FFF. The protocol stack converts the object 0x3000 to the A1 command and the 0x3001 to A2 command etc. In this example if the host want to address **UI[1]** it will send 0x32E2.1 and for **UI[10]** it will send 0x32E2.10.

All the objects can be addressed via the drive interpreter as well. The method of doing that is via the **OJ** prefix and the **OH** prefix. Details about the **OJ\OH** format are in the chapter "Interpreter Prefix"

The alias object for each command is written in the command description.

The Multi-axes

The Platinum drive include multi axes functionality. Depending on the product, several axes can be supported at the same time. In order to address the relevant axis the prefix AX is used. Details about the AX prefix are in the "Interpreter Prefix" chapter.

The axis always treat the basic format as if the first axis was addressed.

For example:

PX retrieves the main position of the first axis. This is the same as writing AX1.PX; or AX1.PX[1];

Addressing the position of axis 2 the **AX2** prefix MUST be used and in this example: AX2.PX;

In order to address the object of the second axis an offset of 0x800 is used. For example:

As mentioned above the UI command alias object is 0x32E2. The UI of the second axis is 0x800 from the first axis hence: 0x32E2 + 0x800 = 0x3AE2 To set **UI[3]**=10 for axis 2 the following options can be used:

AX2.UI[10]=10;

OJ.3AE2[10] = 10;

Not all the parameters are axis related. Consult the relevant command attribute for commands which are related to the device. Note that in this situation addressing the command via different **AX** will result the same value for example the **AN[6]** retrieves the instantaneous bus voltage. The drive includes a single voltage entry and retrieving the value from **AX1** or **AX2** results in the same value.

BIT Status indications

The drive includes a built In test (BIT) each time it starts. The **BZ[1]** command includes the result of the BIT. It is advisable to retrieve the value of the **BZ[1]** each power up.

Status LED blinks

The Status LED blinks in the following scenarios:

- FW download is performed
- Drive FW is burnt
- BIT error (**BZ[1])**

LED \ Time	Meaning
Green constant	Power stage ready for motor enable
Red Constant	Power state error: over\under voltage, over temperature etc.
Alternate: Red 600, Off 200	Drive in Safety error (**BZ[2]\BZ[3]**)
Alternate: Red 200, Off: 200	Parameter process failed during power up (**CD** command)
Alternate: Red 200, Green 200 Red 600, Green 200	Flashing GREEN/RED during burn, frequency depends on the stage of burning/validation and the CPLD/FPGA that is been burned

Interpreter Prefixes

Interpreter prefix are simple interpret commands that allows addressing certain functionality according to the prefix command. After the prefix is set either a command or an object must follow. A “dot” separate the Prefix and the command:

Prefix + . + Command\Object+[<sub index.]

Prefixes do not include sub-indices and cannot be sequentially used which means that each interpretation can contain a single prefix command hence: **AX1.OJ.6040[0]** is an illegal command.

The Prefixes are:

AX# - addressing am axis

GR# - addressing a group \ vector

OJ – addressing the object dictionary using decimal based interpretation.

OH – addressing the object dictionary using Hex-Binary interpretation.

Details are in the following sections.

AX. – Axis Related Commands (Single-axis)

AX. Refer a command for axis related.

CANopen/CoE

Elmo Alias Object:

Attributes

Attribute	Description
Type	Prefix

Remarks

AX. Single-axis related commands

Set command
AX+ < Axis number> +**.** + < Elmo Command > + [+ < sub index> +] + = + <Value to set> +

Get command
AX+ < Axis number> +**.** + < Elmo Command > + [+ < sub index> +] + ;

Where:

< Axis number> = is the number representing the axis number we are applying to. (1-6)

< Elmo Command > = Elmo functions.

< sub index > = sub index.

<Value to set> = In case of set command, this is the value to set.

Usage Example

Axis number ‘1’
Elmo Command: PX
sub index ‘1’

Description	Example
Write (set command)	‘AX1.PX[1] = 40000;’
Read (get command)	‘AX1.PX[1];’

Type	Description	Example
float	Set	‘AX1.PX[1] = 125;’
	Get	‘AX1.PX[1];125.00;’
Double	Set	‘AX1.PX[1] = 4503599627370495;’
	Get	‘AX1.PX[1];4503599627370495.00;’

Type	Description	Example
	Set	'AX1.PX[1];5.55;'
	Get	'AX1.PX[1];5.55;'
Short	Set	'AX1.PX[1] =-1;'
	Get	'AX1.PX[1];1;'
String	Get (no set)	'AX1.PX[1];

Notes

If an error occurs during **AX** operation, the **EC** command returns an error of the **AX** procedure failure, with the extended error appearing at the suitable **EE[x]** command. However, if the object's set/get function returned an error (Elmo error code), this is the error retrieved by the **EC** command.

References

GR. – Group related Commands (Multi-axis)

GR. Refer a command for a group of axes.

CANopen/CoE

Elmo Alias Object:

Attributes

Attribute	Description
Type	Prefix

Remarks

GR. To use the command first define groups of Axes by (AY command).

Set command

GR+ < Group number> +**.** + < Elmo Command > + [+ < sub index> +] + = + <Value to set> + Optional ,+ = + <Value to set>, + = + <Value to set>…;

Get command

GR+ < Group number> +**.** + < Elmo Command > + [+ < sub index> +] + ;

Where:

< Group number> = Group number can be: 1-8.

< Elmo Command > = Elmo functions.

< sub index > = sub index.

<Value to set> = In case of set command, this is the value to set.

Optional: set a different value for each axis.

Usage Example

Group number '3'
Elmo Command: PX
sub index '1'

Description	Example
Write (set command)	'GR3.PX[1] = 40000;'
Read (get command)	'GR3.PX[1];'

Type	Description	Example
float	Set	'GR3.PX[1] = 125;'
	Get	'GR3.PX[1];125.00;'

Type	Description	Example
Double	Set	'GR3.PX[1] = 4503599627370495;'
	Get	'GR3.PX[1];4503599627370495.00;'
	Set	'GR3.PX[1];5.55;'
	Get	'GR3.PX[1];5.55;'
Short	Set	'GR3.PX[1] =-1;'
	Get	'GR3.PX[1];1;'
String	Get (no set)	'GR3.PX[1];

Notes

If an error occurs during **GR** operation, the **EC** command returns an error of the **GR** procedure failure, with the extended error appearing at the suitable **EE[x]** command. However, if the object's set/get function returned an error (Elmo error code), this is the error retrieved by the **EC** command.

References

OH. – Access to Objects represented in 'Hex Binary representation'

OH. enables access to objects via interpreter. The data to be uploaded/downloaded is represented in 'Hex Binary representation' which is representation of the binary data in Hex decimal format.

CANopen/CoE

Elmo Alias Object:

Attributes

Attribute	Description
Type	Prefix

Remarks

To use the command either for SET or GET use the following syntax:

Set command

OH.+<Object index in Hex> + [+ <Object's sub index> +] + = + <Value to set (Binary data in Hex-Dec)> + ;

Get command

OH.+<Object index in Hex> + [+ <Object's sub index> +] + ;

Where:

<Object number in Hex> = Object's index in hexadecimal format (only 4 digit/letters **without** "0x").

<Object's sub index> = Object's sub index.

<Value to set> = In case of set command, this is the value to set. In case of the set command, this is the value to set (**Binary data in Hex format without "0x"**)

Usage Example

Type	Description	Example
float	Set	'OH.32C8[1] = 42FA0000;'
	Get	'OH.32C8[1];42FA0000;'
Double	Set	'OH.3097[13]=432FFFFFFFFFFFFE;'
	Get	'OH.3097[13];432FFFFFFFFFFFFE;'
	Set	'OH.3097[13]=4016333333333333;'
	Get	'OH.3097[13];4016333333333333;'
Short	Set	'OH.3085[1]=FFFF;'
	Get	'OH.3085[1];FFFF;'

Type	Description	Example
String	Get (no set)	'OH.200A[1];685773696C7420653230302E2E303030302E2032314 22037363065533270313037;'

Notes

The binary data (hex-decimal representation) retrieved by **OH** is presented in ***Little Endian*** byte order. The **Little Endian** byte order is valid for all numbers retrieved as binary data (hex-dec) and not only for strings. For example :

The string : 'Whistle'

'W' – 57 (hex decimal of the printable character 'W')

'h' – 68

'i' – 69

's' – 73

't' – 74

'l' – 6C

'e' – 65

The binary data to be retrieved for the string 'Whistle' is: '685773696C7465'

The conversion from floating point number of type 'float' or 'double' into binary data in hex-decimal representation is according to ***IEEE 754 binary format***.

If an error occurs during **OH** operation, the **EC** command returns an error of the **OH** procedure failure, with the extended error appearing at the suitable **EE[x]** command. However, if the object's set/get function returned an error (Elmo error code), this is the error retrieved by the **EC** command.

References

OJ. – Access to Objects represented in 'Decimal representation'

OJ. enables access to objects via interpreter. The data to be uploaded/downloaded is represented in 'Decimal representation' as ASCII printable characters.

CANopen/CoE

Elmo Alias Object:

Attributes

Attribute	Description
Type	Prefix

Remarks

OJ. enables access (read/write) to objects from the Object Dictionary via interpreter regardless of the configured communication (CAN/ECAT).

To use the command either for SET or GET use the following syntax:

Set command
OJ.+<Object index in Hex> + [+ <Object's sub index> +] + = + <Value to set> + ;

Get command
OJ.+<Object index in Hex> + [+ <Object's sub index> +] + ;

Where:

<Object number in Hex> = Object's index in hexadecimal format (only 4 digit/letters **without** "0x").

<Object's sub index> = Object's sub index.

<Value to set> = In case of set command, this is the value to set.

Usage Example

Object index: 0x2205 "Analog Input Object"

sub-index '1'

Description	Example
Write (set command)	'*OBJECT* 2205[1] = 5;'
Read (get command)	'*OBJECT* 2205[1];'

Type	Description	Example
float	Set	'OJ.32C8[1] = 125;'
	Get	'OJ.32C8[1];125.00;'
Double	Set	'OJ.3097[13]=4503599627370495;'

Type	Description	Example
	Get	‘OJ.3097[13];4503599627370495.00;’
	Set	‘OJ.3097[13]=5.55;’
	Get	‘OJ.3097[13];5.55;’
Short	Set	‘OJ.3085[1]=-1;’
	Get	‘OJ.3085[1];-1;’ (or : ‘OJ.3085[1];0xFFFFFFFF;’)
String	Get (no set)	‘OJ.200A[1];Whistle 02.00.00.02 B17 06Sep2017;’

Notes

If an error occurs during **OJ** operation, the **EC** command returns an error of the **OJ** procedure failure, with the extended error appearing at the suitable **EE[x]** command. However, if the object’s set/get function returned an error (Elmo error code), this is the error retrieved by the **EC** command.

References

AB[] – Axis Options

AB[] specifies the option parameters for various axis related algorithms.

CANopen/CoE

Elmo Alias Object: 0x300B

Attributes

Attribute	Description
Type	Unsigned char, Read/Write
Source	All
Restrictions	None
Range	NA
Index range	1 - 11
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

Indices

The following table describes the available options for **AB[]**.

Index	Description	Values		Comments
1	Motor OFF option code	0	Disable drive function, go to SWITCH ON DISABLED state	**Motor Off option code for command MO=0**
		1	Slow down on slow-down ramp and then disable the drive function, go to SWITCH ON DISABLED state	
		2	Slow down on quick-stop ramp and then disable the drive function, go to SWITCH ON DISABLED state	
2	Motor ON option code	Reserved		

Index	Description	Values		Comments
3	Number of points in Position FIR filter.	0 - 16		The motor must be off to change the setting. 0 and 1 disable the position FIR filter. See **GS[8]** for filter's enable modes.
4	Force high resolution PWM mode	0	**Do not force: set according to XP[2]**	The motor must be off to change the setting
		1	Force enable	
		2	Force disable	
5	Voltage saturation option	0	Scaling of voltage vector (preserve orientation)	The motor must be off to change the setting
		1	Preserve direct component, scale quadrature component (reactive current control is preserved)	
		2	Preserve quadrature component, scale direct component (active current control is preserved)	
6	Axis Related configuration bits	**AB[6]** bit field command contains axis related configuration bit. Configuration bits can be accessed with **AB[6]** or **Object 0x300B.** Bit placement see table below		This bit is relevant for DS-402 operation mode 6 "Profile Homing" only.
7	Voltage feed forward option	0	Disable	The motor must be off to change the setting.
		1	Enable equivalent inductance $L\omega i$ compensation	
		2	Enable BEMF feed forward	
		3	Enable BEMF feed forward and $L\omega i$ compensation	
8	Thermal protection method	0	Low Pass Filter thermal protection – compatibility only, not recommended for new implementations.	Default value = 1 (starting from firmware version 02.00.02.00 B212.
		1	I^2T thermal protection	

Index	Description	Values		Comments
9	**High friction algorithm - dead zone and zero FF[2]**	0	Dead zone & zero feed forward disable	**AB[9].0** - Dead zone bit enable/disable. **AB[9].1 - Zero feed forward enable/disable.** 1 – Enable 0 - Disable Default value = 0
		1	Dead zone enable	
		2	Zero feed forward **(FF[2])** enable	
		3	Dead zone & zero feed forward enable	
10	Field Weakening	0	No field weakening	
		1	N/A	
		2	User defined field weakening	The field weakening according to threshold velocity and weakening slope, parameters **XA[7]** and **XA[8]** respectively.
		3	Reactive current command being set from the SIL	
11	Hot plug in option code	0	Motor stops immediately	
		1	Motor continues to rotate or stop the motion according to speed threshold defined in **XA[11].**	

The following describes the **AB[6]** index details:

AB[6] bits	Description
0-3	Reserved
4	This bit is relevant for DS-402 operation mode 6 "Profile Homing" only. **1** – Meaning, store Home Attained bit (SW bit 12) when homing is once successfully performed. When set, the bit Home Attained is reset on power up and on a new initialization of DS-402 Homing procedure (CW bit 4 changed from 0 to 1 under Operation Mode "Profile Homing")
5…7	Reserved

References

AF[], MP[], XP[2], FF[2], XA[11]

AC[N] – Set Acceleration

AC[N] specifies the profile acceleration.

CANopen/CoE

AC[1] Profile_acceleration: 0x6083Elmo Alias Object: 0x300C

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	**Effective on the next call to BG[N]**
Range	1…10^{14}
Index range	1 to 4
Default	1,000,000,000
Unit modes	**UM = 2, 3, 5, 6, 7**
Non-Volatile	Yes
Axis Related	Yes

Indices

Refer to **PA[N]** command for more details.

Remarks

The command defines the maximum allowed acceleration during the operation of point-to-point (**PA[N]**, **PR[N]**) and jog (**JV, JP[N]**) profilers.

AC[N] commands (**N**=2,3,4) for the superimposed, phasing and socket profilers also define their deceleration.

The **AC[N]** command does not affect the present motion. It takes effect only on the next call to Begin Motion (**BG[N]**).

The **AC[1]** value is provided to **Object 0x6083**.

The **AC[N]** command does not affect time-dependent motion, such as Interpolated Position or Cyclic Synchronous Mode (See the MAN-P-ADMINGUIDE).

The acceleration **AC[1]** and deceleration **DC[1]** of the main profiler are subject to the limits of the **SD[1]** value. If the **AC[1]** value is higher than the **SD[1]** value, the **SD[1]** value is used, and the **AC[1]** value is ignored.

The **AC[N]** value can be given in user-defined units specified by the **FC** command.

References

DC, **SD**, **BG**, **PA**, **PR**, **JV**, **FC**

AD – Analog Input Dead-band

AD sets the value of the symmetric dead-band for analog input signals.

CANopen/CoE

Alias Objects: 0x300D

Attributes

Attribute	Description
Type	Float
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 2
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The **AD** command allows to define a symmetric dead-band for analog input signals.

While the analog input is in the dead band, the reading is ignored (considered zero).

The analog input reading will start from 0 at the dead band.

Indices

The following table details the **AD** entries.

Index	Description	Default	Values	Restrictions/Notes
1	Analog Input #1 symmetric dead-band	0	0 to 10	
2	Analog Input #2 symmetric dead-band	0	0 to 10	

References

AS[], AN[]

AE[] – Axis Elements

AE[] specifies the option parameters for the axis related configuration.

CANopen/CoE

Elmo Alias Object: 0x300E

Attributes

Attribute	Description
Type	Unsigned long, Read/Write
Source	All
Range	0…(2^{32}-1)
Index range	1 , 2, 3, 4
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

The command is non-volatile and saved in FLASH. The command is axis related.

AE[4] defines the fault events that shall be treated as warnings. Some faults cannot be treated as warning. When trying to set **AE[4]** with a fault bit that cannot be converted to a warning, the drive shall return an error (`error code 215`) and **AE[4]** will not change. The bits of motor fault that cannot be defined as warnings are marked as "Motor disabled immediately". Refer to the **MF** command description.

Indices

The following table describes the available options for **AE[]**.

Index	Description	Type	Values	Restrictions
1	Reserved			
2	Reserved			
3	Reserved			
4	Set faults as warning	Unsigned long	**The same bit field as MF command. Setting a bit to 1 means "Treat related fault as warning"**	

Note on Warning events:

If present, can be retrieved with the WE[] command

Do not disable the servo

These events are treated with non-latched policy, meaning WE[] bit is auto reset when the conditions do not match to this event anymore. References

References

MO[], SO[], AF[], WE[]

AF[] – Axis Function

AF[] specifies the drive's axes functionality in Remote/Local mode and non-axis related drive configuration.

CANopen/CoE

NA

Attributes

Attribute	Description
Type	Unsigned long, Read/Write
Source	All
Restrictions	None
Range	0…(2^{16}-1)
Index range	1, 2
Default	0 (unless other specified in index table)
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

By default, all drive's axes are in **Remote** mode, meaning that the state machine is controlled by an external field bus host (EtherCAT\CANopen) via **Objects 0x6040**. Refer to the Platinum Administrative Guide for further information about the DS402 state machine.

The axis is in the **Remote** mode if **AF[1]=1** and **AX#.AF[4]=1** (where **#** - axis number).

- In **Remote** mode the servo of the axis cannot be enabled via the legacy servo enable command (**AX#.MO=1**).
- In **Remote** mode the servo of the axis can be disabled via the legacy servo enable command (**AX#.MO=0).** In this situation the axis switches automatically to **Local** mode (**AX#.AF[4]=0** is set internally by the drive).

The axis is in the **Local** mode, if:

- **AF[1]**=0 or **AF[1]**=1 and AX#.**AF[4]**=0
- In Local mode the servo of the axis can be enabled or disabled via commands **AX#.MO**=1, **AX#.MO**=0. **Object 0x6040** is ignored.

The Actual Remote/Local status of the axis can be retrieved by the **AX#.WS[34]** command.

Indices

The following table describes the available options for **AF[]**.

Index	Description	Type	Values		Restrictions
1	Remote mode	Unsigned long	0	Local mode	Motor must be off.
			1	Remote mode (default)	
2	Reserved for internal use	No	**For internal use.**		**Do not modify the parameter**
3	Reserved for internal use	No	**For internal use.**		**Do not modify the parameter**
4	Axis Remote mode	Yes	0	Local mode	From version: 2.0.4.0 B01 Motor must be off.
			1	Remote mode (default)	

Notes

For compatibility reasons setting AF[1]=0 or AF[1]=1 will set the drive to Remote \ Local if AX#.AF[4] was not modified by the user.

From EAS version 3.0 the compatibility when using the AF[1] via the terminal will be kept by the EAS. User does not have to modify the AX#.AF[4] if only AF[1] is used.

References

MO[], SO[], AE[]

AN[] – Get Analog Input

AN[] reads values from the drive's analog inputs.

CANopen/CoE

Elmo Alias Object: 0x3017

Attributes

Attribute	Description
Type	Float, Read-only
Source	All
Restrictions	None
Range	See the table below.
Index range	1 to 11
Default	NA
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

The **AN[]** command reads analog values from the drive's analog-to-digital (A2D) converter. The values are converted to user units according to the Indices table below.

For **Platinum Quartet drives**, the Analog Input 2 is reserved for future use.

Indices

The following table describes the **AN[]** entries.

Index	Description	Units	Values	Restrictions
1	Reads analog input 1.	Volts	+/- 10 0	The analog input can be used as direct motion reference via the feedback socket. Refer to **S#** command.
2	Analog input 2 readout	Volts	+/- 10 0	
3	Reads the instantaneous current feedback from motor phase A	Amperes	NA	The range of values depends on the **MC** of the drive.

Index	Description	Units	Values	Restrictions
4	Reads the instantaneous current feedback from motor phase B	Amperes	NA	The range of values depends on **MC** of the drive
5	Reads the instantaneous current feedback from motor phase C	Amperes	NA	The range of values depends on the **MC** of the drive
6	Reads the bus voltage	Volts	0-BV	The range of values depends on the actual power voltage supplied to the drive
7	Reads the instantaneous current feedback from motor phase D	Amperes	NA	**The range of values depends on MC of the drive**
8	Reads the 5V digital voltage supply to the CPU	Volts	5.0	
9	Reserved			
10	Analog input 3 readout	Volts	+/- 10 0	For **Titanium drive** only
11	Analog input 4 readout	Volts	+/- 10 0	For **Titanium drive** only

References

AG[], **AS[]**, **BV**, **MC**

AP[N] – Analog Sensor Maximum And Minimum Amplitude

AP[N] is a read only command that returns the analog sensor maximum and minimum amplitude and related position.

CANopen/CoE

Analog sensor maximum and minimum amplitude **AP[N]**: Elmo Alias Object: 0x3019

Attributes

Attribute	Description
Type	Double, Read only
Source	All
Restrictions	NA
Range	According to array index
Index range	1 to 7
Default	none
Unit modes	All
Non-Volatile	Indexes 1…4, 6, 7 - No, Index 5 - Yes
Axis Related	Yes

Remarks

The analog sensor amplitude is checked every **HS*TS** uSec and are latched under the following conditions:

- They are latched in **AP[1]** if updated value is greater than previously stored in **AP[1]**

Or

- They are latched in **AP[2]** if updated value is lower than previously stored in **AP[2]**

In these situations, the actual position in user units is also stored in **AP[3]** or **AP[4]** accordingly.

The analog sensor amplitude signal is relevant to the Sine\Cosine Analog Feedback sensor, Hiperface and Resolvers. The amplitude is calculated as follows:

$\mathbf{Amp^2} = (\mathbf{Sin^2} + \mathbf{Cos^2})$.

The amplitude values stored in **AP[1]** and **AP[2]** are $\mathbf{Amp^2}$

Amplitude can be recorded via the EASII: "Analog Amplitude" from group "General"

Nominal value for the 1V p-p is TBD

The voltage is calculated from the amplitude as:

Amplitude (in Volts) = Sqrt(Amp^2) / (Nominal value for the 1V).

Indices

The following table describes the command indexes.

Index	Description	Units	Range	Default
1	Maximum analog sensor amplitude	AD counts	0…65535	
2	Minimum analog sensor amplitude	AD counts	0…65535	
3	Position on maximum analog sensor amplitude	Position user units	-2^{52} to $(2^{52}-1)$	
4	Position on minimum analog sensor amplitude	Position user units	-2^{52} to $(2^{52}-1)$	
5	Analog sensor socket selection	None	0…8	
6	Maximum analog sensor amplitude threshold	AD counts	0…65535	
7	Minimum analog sensor amplitude threshold	AD counts	0…65535	

References

TBD

AS – Analog Input Offset

AS compensates for the offset (bias) of the analog input signal.

CANopen/CoE

Alias Objects: 0x301C

Attributes

Attribute	Description
Type	Float
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 2
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The analog input of the drive is offset during production of the drive. However, sometimes the reference input is not adjusted to the input stage of the host.

The **AS** command should be tuned so that when a zero voltage is applied to analog input *N*, the measurement will also output a zero voltage.

A typical method for setting the offset value is to set **AS[1/2]** to zero, set the input to 0 V (i.e., short to ground), obtain the value of **AN[1/2]** multiple times (x10) and average the readings.

Analog input 1/2 can also be recorded using the EAS recorder. The signal is "Analog Input 1/2".

Note that both the recorder signal and the **AN[1/2]** value includes the values of **AS[1/2]** and **AD[1/2]**.

Indices

The following table details the **AS** entries.

Index	Description	Default	Values	Restrictions/Notes
1	Analog Input #1 offset	0	-10 to +10	
2	Analog Input #2 offset	0	-10 to +10	

References

AD[], AN[]

AY[N] – Define Groups Of Axes

AY[N] defines a group of axes as group number **N**.

CANopen/CoE

Elmo Alias Object: 0x3022

Attributes

Attribute	Description
Type	Unsigned long
Source	All
Restrictions	None
Range	0 to (2^{32}-1)
Index range	1 to 3
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

- Each cell in the **AY[N]** array is 32bit field integer. Which may contain representation of 6 axes numbers (each axis number is 4 bits).
- The total structure of one cell in the **AY[N]** array is:

Bits	Description
0:3	Number of first axis in the group
4:7	Number of second axis in the group
8:11	Number of third axis in the group
12:15	Number of 4th axis in the group
16:19	Number of 5th axis in the group
20:23	Number of 6th axis in the group
24:27	Reserved
28:31	Configuration of axis running the group vector profile. Affective when a synchronized (vector) motion of the group is enabled by command **EV** =1 or **EV**=2.

- For example, if the user wants to set group no.3 to be the axes: 1,2,5,7
  ‘**AY[3]** = 0X00007521;’ (hex)
  Or
  ‘**AY[3]** = 29985;’ (dec)

- If the user wishes to refer a command (**PX[1]** for example) and not to a single axis , but to a group of axes, for example group no.3 from the previous example, the following representation is required: '**GR3.PX[1]**' where
  '**GR**' – prefix represents the group of axes, a shortcut of the word : Group.
  In this way the user obtains a value of **PX[1]** that references axes : 1,2,5,7.
- **Set command of group of axes:**
  To set a value (40000 for example) into **PX[1]** referring to the same group of axes, the following representation is required:
  '**GR3.PX[1]** = 40000;'
- **Bits 28:31** of the AY[N]command define which axis runs vector profile:

  AY[N].bit28:31=K – vector profile of group #N is run by separate axis #K, not belonging to any group. N must be smaller or bigger than all group #N axes numbers. For example, if group #2 consists of the axes 2,3 and 4, the vector can be run by Axis #1 or Axis #5 defined by AY[2]=0x10000432 or AY[2]=0x50000432:
  If vector profile axis is selected to be after group axes (Axis #5), an additional delay of HS sample exists after profile calculation;
  If vector profile axis is selected to be before group axes (Axis #1 above), the group cannot contain axis #1;
  Vector profile axis has to be configured as virtual axis (AXK.CA[28]=0) and enabled (AXK.MO=1);
- **AY[N].bit28:31=0** – vector profile of group #N is run by the - axis having the smallest number. For example, if group #1 consists of the axes 3,4 and 5, the vector will be run by Axis #3, if group defined by AY[1]=0x345:
  This configuration should be used, if there is no "free" axis for running of the vector profile, for example, when all axes of the drive are used for motors control (for example, **Platinum Quartet drive** connected to 6 DC brush motors);
  Fault of any of one of the vector axes disables all remaining vector axes immediately;

Restrictions

- All axes numbers in the group must be in the range: 1 to **WS[1]**.
  If the user defines more axes than allowed, an error is returned: Drive Error 250: Group axis number is not valid.
  For example:
  If maximum number of axes defined in the system WS[1] is 2, the group definition AY[1] = 0x0000132 is illegal
- All axes numbers defined in one group should be different, one from another. If two identical axes numbers are defined, the error will be returned: Drive Error 250: Group axis number is not valid
  For example:
  The maximum number of axes defined in the system is 6 and the user entered: **AY[1]** = 0x00642511.
  The error is of defining twice the axis '1'.

References

EV

BF[] – Brake Fall Parameters

The **BF[]** specifies the parameters allowing to reduce \ eliminate the possible fall during the release of a brake.

The **BF[]** works with conjunction to **BP[2]** command that defines the brake release time.

CANopen/CoE

Elmo Alias Object: 0x3033

Attributes

Attribute	Description
Type	Float, Read\Write
Source	All
Restrictions	No
Range	According to array index
Index range	1 to 2
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

Setting the **BF** reduces and even prevents a brake fall due to the unbalanced force. The brake fall prevention requires gain & current compensation as described below.

The brake fall prevention is active when **BP**[2], **BF**[1] & **BF**[2] are not zero.

Indices

The following table describes the command indexes.

Index	Description	Range	Restriction
1	Brake release compensation algorithm gain	0 to 100	
2	Brake release current compensation	-PL[1] to PL[1]	

References

BP[]

BG[N] – Begin Motion

BG[N] starts the next profiled motion.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Command
Source	All
Restrictions	NA
Range	NA
Index range	1 to 4
Default	NA
Unit modes	**UM = 2, 3, 5, 6, 7**
Non-Volatile	No
Axis Related	Yes

Indices

Refer to **PA[N]** command for more details.

Remarks

The **BG[N]** command is used to activate the next profiled motion.

On the **BG[N]**command, the relevant motion target data (set point) is sent to the profiler, which then calculates the command for the control loop.

BG[1] affects the present motion mode by modifying the profiler and/or controller which are used by the mode.

The "Motion Mode" is determined by the "Elmo Motion Command," which should be the last effective command for the presently required motion (see the table below).

The Actual Motion Mode is presented by CANopen object 0x6061.

The effect of **BG[1]** on the motion mode is considered according to the next table.

Elmo Motion mode	UM value	Elmo motion command	DS-402 Motion (0x6061)	Relevant parameters considered
PTP	**UM = 5,** **UM = 3**	**PA, PR**	Profile Position, 1	**AC, DC, SP, SF, FS**
JV	**UM = 2**	**JV**	Profile Velocity, 3	**AC, DC, SF**

Elmo Motion mode	UM value	Elmo motion command	DS-402 Motion (0x6061)	Relevant parameters considered
JP	**UM = 3, 5, 6, 7**	**JP**	Profile Position, 1	**AC, DC, SF**

BG[1] is also used to convert Elmo's commands to DS-402 objects and switch between motion modes. For details, see the following table.

Elmo Command	Converts to CANopen Object	Action Performed	Note
JV	0x60FF	• Switch motion mode to Profile Velocity • Saturation to maximum speed	
JP	0x6081 0x6082	• Switch motion mode to Profile Position • Saturation to maximum speed	
-	0x607E	Set to 0 indicating: do not convert polarity	
PA/PR		Switch motion mode to Profile Position	

Note:
The command also affects the DS-402 control word (object 0x6040). The value of this object is determined according to the actual motion mode.

The **BG[N]** command removes any pending Halt from DS-402.

BG[N] should have no effect in torque modes or in a time-dependent mode, such as Synchronous Cyclic Position or Interpolated Position.

If the recorder is triggered by a Begin Motion command, **BG[1]** will start the recording.

If the User Program includes an Auto_BG routine, the routine should be called automatically in the next cycle after **BG[1]**.

References

UM, PA, JV, JP, SP, SF, FS, PO

BH – Upload Recorded Data

BH uploads the values recorded by the recorder to a host in Hex-Binary or Binary format.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Command, read-only
Source	All, except FoE
Restrictions	• Valid recorded data is ready (see **RR** command). • No other uploading sequence is performed (**UL**).
Range	Recorder buffer mode: 1 to 2^{16}, bit field format Recorder Real-time mode: The value is ignored
Index range	NA
Default	None
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

BH is a bit-field command, where every bit points to a specific recorded signal (e.g., "Position") that can be uploaded. The signals are requested by using the **RC** command in conjunction with the **RV[]** command. **BH** can only be used after the recorder data is valid see **RR** command, which launches and retrieves the status of the recorder.

Please refer to **RR** command for further information about the recorder procedure.

The **BH** command is designed to optimize data transfer from the drive to the host while allowing fetching of the data in a simple Binary format or Hex-Binary text format.

In buffered mode the BH command supports the following upload formats:

<table>
<tr><th>BH index</th><th colspan="2">Description/Format</th></tr>
<tr><td rowspan="6">1
(Same as BH without any index)</td><td colspan="2">Recording upload data format is according to communication type:</td></tr>
<tr><td>**Communication**</td><td>**Data Format**</td></tr>
<tr><td>RS232</td><td>Hex-Binary</td></tr>
<tr><td>UDP</td><td>Hex-Binary</td></tr>
<tr><td>USB</td><td>Hex-Binary</td></tr>
<tr><td>CAN</td><td>Binary</td></tr>
<tr><td>2</td><td colspan="2">**Recording data will be uploaded in Binary format.**</td></tr>
</table>

In Hex-Binary format (recorder buffer mode):

Each number is transferred in two printable characters. For example, the value 10 (0x0A) is transmitted in two chars: 0x30 and 0x41.

The uploaded stream consists of two main parts:

- Data header
- Data value

The data header contains the following information about the recorded data.

Byte	Description
0 to 7	**The length of data been actually uploaded from the drive in the current BH upload packet.** **For example: in case the number of bytes to uploaded that was set by EAS in RP[8], RP[9] is 100 but the actual remained number of bytes to upload from drive is 40, thus this field will contain the value 40 and this is the length of data that will be uploaded in the current packet.**
8 to transmit data length	Data of the specific uploaded signal

In Binary format (recorder buffer mode):

Each number is transferred as is.

The uploaded stream consists of two main parts:

- A data header
- A data value

The data header is exactly the same as in 'HexBinary' format (See section Hex-Binary format above).

Byte	Description
0 to 3	**The length of data been actually uploaded from the drive in the current BH upload packet.** **For example : in case the number of bytes to uploaded that was set by EAS in RP[8], RP[9] is 100 but the actual remained number of bytes to upload**

Byte	Description
	from drive is 40, thus this field will contain the value 40 and this is the length of data that will be uploaded in the current packet.
4 to transmit data length	Data of the specific uploaded signal

In Binary format (recorder real-time mode):

1. Each number is transferred as is
2. The uploaded stream consists of data only (i.e. no header).

All Recording Modes

During the uploading of data, the drive can receive and execute other commands. However, the drive will not be able to reply the commands unless it is used in a communication channel that is different from the channel used by the **BH** request. For example, if the recorded data is fetched from the RS232 communication channel, the USB communication channel can still be used to execute motion commands while the RS232 channel is uploading the data.

Recorded data can only be fetched one at a time. This means that if **BH** is already in process, other **BH** commands cannot be executed by any other communication line.

During the uploading the following commands cannot be executed:

- A **PP[1]** command for engaging new communication parameters
- An **HM[]/HF[]** command when the recorder buffer is used as a position capture buffer
- **FT[]**,**RC**, **RG**, **RL**,**RP[]**,**RR** and **RV[]** commands are used during the recorder setting.

The recorder allows the uploading of global variables from the User Program as well. See the **RR** command and the User Program manual for further information.

References

FT[], **RC**, **RG**, **RL**, **RP[]**, **RR**, **RV[]**

BP[] – Brake Parameters

BP[] specifies the time parameters for the logical brake. The command is axis related.

CANopen/CoE

Elmo Alias Object: 0x303D

Attributes

Attribute	Description
Type	Short, Read/Write
Source	All
Restrictions	**One of the digital outputs must be defined as a brake using OL[*N*].**
Range	According to the table below
Index range	1, 2
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Indices

Outputs are logically set/reset according to the following table:

Index	Description	Values	Restrictions
1	The delay time that is needed to engage the brake before the motor is actually stopped. **The delay time set by BP[1] is the time between the request to disable the motor and the actual time when the servo is off.**	0 to 1000 milliseconds	
2	The delay time that is needed to release the brake before the motor is actually started. **The delay time set by BP[2] is the time between the request to enable the motor and the actual time when the servo is on and the profiler can actually be used.** During this time profiler references (software set points & auxiliary) are ignored. **If auto-phasing commutation is required, it should be activated after the time set by BP[2].**	0 to 1000 milliseconds	
3	Reserved		

Index	Description	Values	Restrictions
4	PWM period during release	For 0 – 7: **Period[%]= 12.5+BP[4]*12.5** For 8 (default): PWM is disabled	Available only via Digital output 8 when the brake is connected between Output 7 & Output 8
5	**Time duration for 100% PWM when brake is released. After this time the BP[4] PWM period is used. This allows to energies the brake before the PWM starts.**	0 to 32000 milliseconds Default: 100	Available only via Digital output 8 when the brake is connected between Output 7 & Output 8.

Remarks

The drive allows the application to use a brake to hold the motor while (break is engaged) the servo is off (**SO** == 0). **BP[]** is used to define the times that are needed to engage and disengage the brake.

The brake will be activated only if one of the digital outputs is defined as a brake by the **OL[*N*]** command.

OL[] also defines the logic level by which the output is activated. Normally, the hardware connection to the brake is such that when the drive is powered off the brake is engaged (current does not flow though the brake windings).

Any digital output can be used as a brake logic output. Output 1 (**OL[1]**) also supplies current for the brake. Please refer to the specific drive's Installation Guide for more information about the current source for this purpose.

Digital output 8 can be used to release the brake by PWM signals. This feature allows to save energy during the release of the brake when the motor is enabled. The PWM is activated when **BP[4]** is set to != 8.

Disabling the servo

When the servo is disabled by setting **MO** = 0, the brake is engaged (current does not flow though the brake windings), **OP** bit 17 and **OB[18]** command are set to 1, and the corresponding indication (**SO** = 0) is received only after the time set by **BP[1]**. Please refer to the **SO** command.

Note: **OP** and **OB[]** command are not affected by **OL[]** command polarity bit (bit 0). Polarity bit has impact on physical output only.

When a DS-402 state machine is used, Switch On, Ready to Switch On or Switch On Disable should be indicated by the status word only after the time needed to engage the brake has elapsed.

Enabling the servo

When the motor is enabled by setting **MO** = 1, the brake is released/disengaged (current flows though the brake windings), **OP** bit 17 and **OB[18]** command are set to 0, the **SO** (Servo On) indication should be set to 1 only after **BP[2]** milliseconds under the assumption that the brake was released. Please refer to **SO** command.

When a DS-402 state machine is used to enable the motor, the Status Word object (object 0x6041) should indicate Operation Enabled only after the brake is released. The host should consider this when the time-out is calculated.

Fault reaction

The brake output is activated immediately when a motor fault occurs (**MF** > 0). Both the Motor On and Servo On indications (**MO** and **SO**) are set to 0. In the case of an amplifier fault (i.e., Overvoltage, Over temperature, Short Protection and Safety Active) there may be no drive controlling the servo before the brake is fully engaged.

Notes

- In cases in which the drive is in Stepper Mode (**UM** = 3) and **SC[8]** is used for automatic setting of the torque, the torque will be applied regardless of the state of the brake.
- The effect of **BP[2]** is considered only on the next motor on.
- The effect of **BP[1]** is considered only on the next motor off.
- The resolution for the brake output response is HS*TS in µs.

References

OL[], **OP**, **SO**,

BS[] – Bring Recorded Sample

BS[*N*] obtains the (*N* - 1) samples from a pre-selected recorded data vector, i.e. **BS[1]…BS[N-1]**, where N is actual recorder length that can be retrieved by **RA[1]** command.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	64-Bit (short, unsigned short, long, unsigned long, long long, unsigned long long, float, double), Read-only
Source	USB, TCP, EoE, RS232
Restrictions	Recorder ready (**RR == 0**); Valid vector selected (**RP[11]**)
Range	NA
Index range	1 to 16384
Default	NA
Unit modes	All
Non-volatile	No
Axis Related	No

Remarks

The **BS[]** command provides an interface which is used by the User Program to read variables previously recorded by the Recorder. **BS[]** returns a specified sample from a pre-selected recorded vector. The selected vector, from which the recorded signal samples are retrieved, is specified by **RP[11]** command.

An index range that starts from 0 allows simpler modulo operations.

N can have a maximum value of 16384 (the longest possible vector). However, *N* also depends on the number of recorded points. Refer to the **RL, RA** command for more details.

Indices

Outputs are logically set/reset according to the following table:

Index	Description	Type	Values	Restrictions
0	First sample	Any type		Always returned 0
1	Second sample	Any type		
…	…	…		
N-1	*N* th sample	Any type		

References

RR, RL, RP[11]

BU[N] – Set boot up mode

BU[*N*] command sets and enables boot up modes and dedicated for ELMO internal use only.

CANopen/CoE

Elmo Alias Object: 0x3042

Attributes

Attribute	Description
Type	Long, write only
Source	All
Restrictions	• The motor must be off, else the command returns error 57 "Motor must be off". • The user program must not be running (**PS !**= 1) else the command returns error 82 "Program is running".
Range	According to array index
Index range	1 to 2
Default	none
Unit modes	All
Non-Volatile	No
Axis Related	No

Indices

The following table describes the **BU[*N*]** entries.

Index	Mode	Value	Description
1	Power reset	0x1234 or 0x4321	Reboot drive
		Any other value : Incorrect password	The command is ignored
2	Signature FW	0x4321	When FW signature successful the command returns 0
			When signature was not successful the command returns possible errors: 165 "Flash read fail" , error: 88 "Flash distribution fail", error: 11 "Flash write fail"
		Any other value: Incorrect password	Error 21 "Out of range" returned

References

MO, PS

BV – Bus Voltage

BV gets the maximum drive bus voltage in volts.

CANopen/CoE

Elmo Alias Object: **0x3043**

Attributes

Attribute	Description
Type	Unsigned Long, Read-only
Source	All
Restrictions	None
Range	BV > 0 (The bus voltage depends on the drive and has no range.)
Index range	NA
Default	None
Unit modes	All
Non-Volatile	Constant
Axis Related	No

Remarks

The configured bus voltage is burned into each drive. The BV value is burned-in in Elmo during manufacturing and provides the voltage which is stated on the label plate.

Note that the factory under-voltage threshold (WS[25]) and the factory overvoltage threshold (WS[23]) are indicated in their specific parameters.

The actual voltage can be read by calling AN[6].

The voltage thresholds can be modified by calling XP[1] for the overvoltage and XP[5] for the under-voltage.

The actual values of the voltage thresholds can be read by calling WS[26] for the under-voltage and **WS[24]** for the overvoltage.

References

AN[], **WS[23]**, **WS[24]**, **WS[25]**, **WS[25]**, **XP[1]**, **XP[5]**

BZ – Software BIT

BZ retrieves the drive software BIT status.

CANopen/CoE

Elmo Alias Object: 0x3047

Attributes

Attribute	Description
Type	unsigned long, read-only
Source	RS232, USB, TCP, EoE, CoE
Restrictions	None
Range	None
Index Range	1 to 3
Default	NA
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

A bit in the **SR** command will indicate a BIT error (see **SR** command bit 27). If this bit is set, the **BZ[1]** command should be checked for the source of the failure.

Note:
If the BIT error is set, the drive status LED flashes and the motor cannot be enabled.

Indices

The following table describes the **BZ[*N*]** entries. Index values which are not indicated will return 0.

Index	Description	Format			
1	Power-up BIT	**Bit**	**Description**		**Cleared**
		0 - 3	DI error:		Power-up
			0	OK	
			1	Error: Incompatible HW Safe/Non-Safe version	
			2	Error reading **DI** from Flash (CRC, signature), Drive Info or Product Code	
			3	Error reading User Info-Table from Flash.	

Index	Description	Format			
			4	Error drive product code does not match any product in firmware	
			5	**Error drive info-table, and user info-table do not include the drive type (BV, MC)**	
			6	Error Incompatible Compilation Type	
		4	Post process error:		Power-up
			0	OK	
			1	The drive parameters process detected an error. The error can be retrieved via the **CD** command. **Note:** In versions up to 2.0.3.9 B07 the parameter database is reset to default. Starting from version2.0.3.9 B07 the parameters are no longer reset, allowing the present communication parameter to be retained.	
		5	Post process reset error:		Power-up
			0	OK	
			1	Error in setting parameters to default	
		6	CAN driver initialization error:		Power-up
			0	OK	
			1	Error	
		7	EtherCAT initialization error:		Power-up
			0	OK	
			1	Error	
		8	Safety status:		**BZ[1] & BZ[2] are both 0**
			0	Safety status OK	
			1	Safety status error. Refer to this command sub index 2/3	
		9	New FW Image distribution fail: See **EE[12]** and **EE[13]**		Power-up
			0	No new FW distribution, or distribution OK	
			1	New FW distribution failed	

Index	Description	Format			
		10	FW has no signature:		Power-up
			0	FW has signature	
			1	**FW has no signature. The user must sign the FW via the BU Command.**	
		11	Configuration pins error:		Power-up
			0	No error	
			1	ID pins not stable	
		12	Reserved		
		13 to 14	Ethernet Configuration:		Power-up
			0	No Ethernet	
			1	Ethernet on EtherCAT out port	
			2	Ethernet on additional PHY	
		15	Shared RAM error – in debug mode:		Power-up
			0	All are OK (firmware loaded by boot)	
			1	Debug mode (firmware loaded using debugger)	
		16	EtherCAT invalid EEPROM at power-up:		Power-up
			0	EEPROM valid	
			1	EEPROM invalid	
		17	EtherCAT invalid updated EEPROM (after user write or reload command):		Update EEPROM valid
			0	EEPROM valid	
			1	EEPROM invalid	
		18	Primary CPLD initialization status:		Power-up
			0	OK	
			1	Error	
		19	Secondary CPLD initialization status:		Power-up
			0	OK	
			1	Error	
		20	User Anan initialization status:		Power-up
			0	OK	
			1	**Error - see EE[14] for details**	

Index	Description	Format			
		21	EEPROM Revision not valid		Power-up
			0	OK	
			1	FW revision number differs from EEPROM revision number	
		22 to 31	Reserved		
2	Safety BIT (Primary safety CPU)	0	Safety memory test status:		Power-up
			0	OK	
			1	Error	
		1	Safety code CRC check power-up:		Power-up
			0	OK	
			1	Error	
		2	Safety code CRC periodic check status:		Power-up
			0	OK	
			1	Error	
		3	Safety parameters CRC periodic check status:		Power-up
			0	OK	
			1	Error	
		4	Watchdog Timer initialization status:		Power-up
			0	OK	
			1	Error – timeout waiting for WD reset	
		5	Watchdog Timer runtime error:		Power-up
			0	OK	
			1	Error – WD counter stopped working	
		6 to 7	Clock monitoring error:		Power-up
			0	No error	
			1	Counter overflow, no frequency is detected	
			2	Measured frequency out of pre-defined range	
		8	ADC initialization error:		Power-up
			0	No error	
			1	Error	
		9	ADC Run, FIFO overrun error:		Power-up

Index	Description	Format			
			0	No overrun	
			1	Overrun	
		10	ADC Run, FIFO underflow error:		Power-up
			0	No underflow	
			1	Underflow	
		11	Safety internal communication CRC:		Power-up
			0	OK	
			1	Error	
		12	Safety internal communication counter:		Power-up
			0	OK	
			1	Error	
		13 to 15	Safety manager configuration error:		Power-up
			0	OK	
			1	Unsupported object (mapped safety function)	
			2	Unsupported object size	
		16	FSoE communication diagnostics status:		Power-up
			0	OK	
			1	Error	
		17	Internal communication between CPLD and secondary safety CPU status:		Power-up
			0	OK	
			1	Error	
		18	**Safety function execution error (see logger for specific function and error 0x2300/0x2301):**		Power-up
			0	All safety functions are OK	
			1	At least one safety function has a diagnostic error	
		19	STO diagnostic status:		Power-up
			0	OK	
			1	There is an error in STO HW. STO error must be cleared by power-up only.	
		20	Safety sensor diagnostic status:		

Index	Description	Format			
			0	OK	SafeOp-OP or Power-up
			1	Error	
		21	Safety torque diagnostic status:		SafeOp-OP or Power-up
			0	OK	
			1	Error	
		22	Safety mapping during SafeOP to Operational state transition status:		SafeOp-OP or Power-up
			0	OK	
			1	Error	
		23	FSoE communication state machine status:		SafeOp-OP or Power-up
			0	OK	
			1	FSoE included incorrect state transition/ PDU CRC etc..	
		24	**Safety IO diagnostic error (see logger for specific IO and error 0x2300/0x2301):**		Power-up
			0	OK	
			1	Error	
		25	Safety auto-mode CRC error		Power-up
			0	OK (OR not in auto mode)	
			1	Invalid CRC	
		26	Voltage monitoring error		Power-up
			0	OK	
			1	Voltage monitoring error (see logger for detailed error description)	
		27	CPLD Error:		Power-up
			0	OK	
			1	CPLD Error	
		28-31	Reserved		
3	Safety BIT (Secondary safety CPU)	Same bit description as sub-index 2.			
131		0-15	Board PLL frequency		

Index	Description	Format	
	Compiled frequencies report	16-31	MPU frequency (CPU core)
132	Compiled numbers report	0-3	Number of compiled application axes
		4-7	Number of safety axes
		8-11	Primary CPLD axes
		12-15	Secondary CPLD axes
		16-19	Number of analog inputs
		20-23	Number of kinematics axes

References

SR, BU

CA[N] – Commutation Parameters

CA[] sets the commutation related parameters.

CANopen/CoE

Alias Objects: 0x3052

Attributes

Attribute	Description
Type	Long long
Source	All
Restrictions	According to index
Range	According to index
Index range	According to index
Default	According to index
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

It should be noted that modification of the command should be performed using the EAS Quick or Expert Tuning interface. Manual modification may result in ambiguous results from the sensor or function that uses it.

Indices

The following table details the **CA[]** entries.

Index	Description	Default	Values	Restrictions/Notes
1	**Stepper angle command source in stepper mode (UM=3,6)**		0 to 1	0 – main profile **1 – position reference (defined by SA[12])**
7	Phase offset relative to the absolute commutation sensor (Hall, Absolute Serial, Analog Hall and Resolver) in stepper units	0	-4096 to 4095	The motor must be off to change the setting. Changing the setting resets commutation.

<table>
<tr><th>Index</th><th colspan="3">Description</th><th>Default</th><th>Values</th><th>Restrictions/Notes</th></tr>
<tr><td rowspan="5">16</td><td colspan="3">Force commutation search for every motor on</td><td rowspan="3">0</td><td rowspan="3">0 to 1</td><td rowspan="3"></td></tr>
<tr><td>Bit</td><td colspan="2">Description</td></tr>
<tr><td rowspan="3">0</td><td colspan="2">Force Commutation search</td></tr>
<tr><td>0</td><td>Use commutation when known</td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>1</td><td>Force commutation search for every motor on</td></tr>
<tr><td rowspan="9">17</td><td colspan="3">Commutation method</td><td rowspan="9">1</td><td rowspan="9">1 to 9</td><td rowspan="9">The motor must be off to change the setting.
Changing the setting resets commutation.</td></tr>
<tr><td>1</td><td colspan="2">By Hall sensor</td></tr>
<tr><td>2</td><td colspan="2">By stepper, motor will move to a certain stepper position</td></tr>
<tr><td>3</td><td colspan="2">By binary search, minimal movement when commutation is not known</td></tr>
<tr><td>4</td><td colspan="2">By analog Hall sensor</td></tr>
<tr><td>5</td><td colspan="2">By serial absolute encoder</td></tr>
<tr><td>8</td><td colspan="2">By absolute & analog commutation (for Hiperface sensor only)</td></tr>
<tr><td>9</td><td colspan="2">By BEMF observer (sensorless)</td></tr>
<tr><td>18</td><td colspan="3">Feedback counts per electrical cycles.
The value is used to determine the number of counts per electrical cycles (MP[4]) of the commutation socket.
In rotary motion counts per revolution
In linear motion counts per single electrical cycle</td><td>10000</td><td>6 to (2^{52}-1)</td><td>The motor must be off to change the setting.
Changing the setting resets commutation.</td></tr>
</table>

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Default</th><th>Values</th><th>Restrictions/Notes</th></tr>
<tr><td>20</td><td colspan="2">Hall + encoder mismatch field angle, in stepper units</td><td>0</td><td>0 to 1365
(120 degree)</td><td>If CA[20] is zero, this function is disabled</td></tr>
<tr><td rowspan="3">25</td><td colspan="2">Reverse the direction of stepper angle, equivalent to switching the motor cables between M2 and M3</td><td rowspan="3"></td><td rowspan="3">0, 1</td><td rowspan="3">The motor must be off to change the setting.
Changing the setting resets commutation.</td></tr>
<tr><td>0</td><td>Do not reverse</td></tr>
<tr><td>1</td><td>Reverse the direction</td></tr>
<tr><td rowspan="9">28</td><td colspan="2">Configure motor connection</td><td rowspan="9">0x20000
000</td><td rowspan="9"></td><td rowspan="9">The motor must be off to change the setting.
Changing the setting resets commutation.</td></tr>
<tr><td>Bits</td><td>Description</td></tr>
<tr><td>0 to 23</td><td>Motor phases connected to axis</td></tr>
<tr><td>24</td><td>Axis controls attached external drive</td></tr>
<tr><td>25 to 26</td><td>Reserved</td></tr>
<tr><td>27</td><td>Linear motor connected to axis</td></tr>
<tr><td>28</td><td>Axis runs motor simulation</td></tr>
<tr><td>29</td><td>No Motor</td></tr>
<tr><td>30 to 31</td><td>Reserved</td></tr>
<tr><td>29</td><td colspan="2">Changing commutation offset on the fly, in stepper units</td><td>0</td><td>-4096 to 4095</td><td></td></tr>
</table>

Notes

CA[28]:

- The motor connections can be configured if there is an available hardware
- To configure the motor, it is required to set the motor phases connected to the drive (bits 0 to 23): bit 0 defines M1 power connector, bit 1 - M2 and so on. For example, AX1.CA[28]=0x7 configures brushless motor connected to M1, M2 and M3 power connectors.
- Phases of one motor must be successive (setting AX1.CA[28]=0xD is wrong).

  Example:
  The drive having hardware for 4 phases, can be set for:
  - One stepper motor: AX1.CA[28]=0xF.
  - One brushless motor: AX1.CA[28]=0x7 or AX1.CA[28]=0xE;
  - Two DC brush motors: AX1.CA[28]=0x3 and AX2.CA[28]=0xC.

  The drive having hardware for 12 phases, can be set for:

- Four brushless motors: AX1.**CA[28]**=0x7; AX2.**CA[28]**=0x38; AX3.**CA[28]**=0x1C0; AX4.**CA[28]**=0xE00;
- Three stepper motors: AX1.**CA[28]**=0xF; AX2.**CA[28]**=0xF0; AX3.**CA[28]**=0xF00.

- To run a motor simulation on the axis, set **CA[28]**.bit28 and define the relevant sensor (ID=34).
- Setting **CA[28]**=0 defines a Virtual axis - axis running the motion profile only. Virtual axis can be used for example, for generation of ECAM master command or Gantry Yaw command.

References

MO, **SC[]**, **SE[]**, **UM**

CC[N] – Compilation Checksum

CC command signs the user program and allows it to be executed after successfully completing the downloading procedure. The command is dedicated for ELMO internal use only

CANopen/CoE

Elmo parameters object: 0x3054

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All, excepts the user program
Restrictions	None
Range	None
Index range	1
Default	-1
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

In order to run the user program after it was downloaded, the host sends the **CC** command that indicates the user program checksum. The checksum is calculated by the drive during the downloading procedure and the drive compares it to the CC. If the value is correct the user program is processed and ready for XQ## command that executes it. If the value is not equal to the one calculated by the drive, an error will be reported.

If there is no program to load – the program will not return an error.

When a drive is powered up, if the user program exists (**PS=-1**) then the **CC** is called automatically to allow an updating of the user program without the need to resent the CC.

The existence of the user program increases the drive boot-up time.

The checksum calculated is the 2's complement of the 16-bit summation of the user program data (code + symbols table).

The value is indicated in CompileInfo.txt file which is located in the same folder as the "TestName".ehl file.

References

XQ, LP[*N*], CP, HP, KL, PS

CD – Upload Detailed Error

CD returns a detailed error for particular situations.

CANopen/CoE

Elmo Alias Object: 0x3055

Attributes

Attribute	Description
Type	Unsigned long long (64bit field), Read-only
Source	All.
Restrictions	None
Range	Maximum 64 bit number.
Index Range	Array 2 cells: indexes 0 – 1.
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

Index in CD	Description
1	Detailed error of RS/LD/SV procedures.
2	Detailed error of upload/download /upload FoE/download FoE parameters xml procedures.
3	SIL saved user parameters power-up processing detailed error – relevant when SIL error via **EE[18]** is 5.

The **CD** value will be set with a detailed error when one of the following processes returns an error upon failure:

- Parameters database process (SV/RS/LD):
  The drive includes parameter's database which are saved in the non-volatile memory and are loaded to the RAM at power up and when **LD** command is requested. The database is processed by the drive to evaluate the parameter validity. This process might return an error in case of a failure. This error is reported via **CD** command.
  The database is processed in the following situations:
  - During drive boot-up
  - During drive reset (**RS** command)
  - When parameters are saved (**SV** command)
  - When parameters are loaded to the RAM (after the **LD** command)

In this case the error of that process will be returned via **CD[1]**.

- Upload/ Download parameters xml
  When one of the following processes: upload/download parameters xml , upload/download FoE parameters xml returns an error in case of failure, the **CD[2]** command will return a detailed error.

Return Values

The structure of each bit field value of **CD[1] & CD[2]:**

Bits	Description	Type	Values Range
0-15	**Object Index** in which the error occurred.	unsigned short – 16bit	0x0000-0Xffff. **Note:** In case of axis related object, the index is a virtual index and will include the offset according to the axis number. The ranges of object indexes are described in the doc: “Object Dictionary Requirements and Detailed Design_vXX”
16-23	**Sub-index** in which the error occurred.	unsigned char - 8bit	0 – 254. According to DS 402.
24-39	**Elmo Error code** of the current error - EC.	Unsigned short – 16bit	0 – 65535.
40-63	Reserved		0

The structure of each bit field value of **CD[3]**:

Bits	Description	Type	Values Range
0-15	Parameter size in Bytes	unsigned short – 16bit	0x0000-0x0008
16-23	Parameter offset in user parameter structure	unsigned short – 16bit	0x0000-0xFFFF
31-47	Parameter ID	unsigned short – 16bit	0x0000-0xFFFF
48-63	Reserved		0

Notes

- In case of failure in LD/SV/RS, 'EC' indicates “Database reset see CD for more details" error.
- In case of failure in power-up post process, a flag indicating post process fail is set in **BZ** command. (bit xx in the **SR** command indicates that a system error occurred and the relevant information is in **BZ**)

References

LD, SV, RS, BZ, SR

CF[] – Capture Input Filter

CF[] defines the time period of the HW capture input filter with resolution of 20 nanoseconds. The command is non-axis related.

This command was originally added for the Platinum drive but is no longer relevant. It has been reserved for the Titanium drive.

CANopen/CoE

Elmo Alias Object: **0x3057**

Attributes

Attribute	Description
Type	Parameter, unsigned short
Source	All
Restrictions	None
Range	See table below
Index range	1 to 12
Default	0 (no filter)
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Indices

The following table describes the **CF[]** entries for both Platinum and Titanium drives.

Index	Description	Range	Comments
1, 2	**CF[1]**, **CF[2]** – filter for Capture 1, Capture 2 modules	0 to 4095	
3 - 8	Reserved		Write access has no effect and does not return error. Read access returns 0
9	**CF[9]** – filter for Time Capture 1 module	0 to 4095	
10 - 12	Reserved		Write access has no effect and does not return error. Read access returns 0

References

CR[], **GI[]**, **IF[]**

CG[] – Non-linear Current Feed-Forward

CG[*N*] configures and enables the mode of the non-linear current feed-forward (cogging compensation) by mechanical position.

CANopen/CoE

Manufacturer specific profile area objects: 0x3058

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 7
Default	According to array index
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

Non-linear current feed-forward by mechanical position is a method that can compensate:

- Cogging torque of electrical motors which is the torque due to the interaction between the permanent magnets of the rotor and the stator slots of a Permanent Magnet (PM) machine.
- Other mechanical problems that can be overcome by the current feed-forward.

Note, that additional method of cogging torque compensation by field angle is defined by **NF[3]-NF[7]** commands and **NT[]** table. Both these methods can be activated simultaneously.

CG[] command enables the user to set the non-linear current feed-forward parameters and to enable/disable this mode.

The mode can be activated using a linear table, where the axis main feedback position is taken as an absolute entry in the table. The mode can also be activated as a cyclic mode (modulo), where the position is calculated for each modulo value, as defined by the user, producing endless cyclic current command correction.

The table entries are the current corrections which need to be added to the current command at a specific feedback location. To edit the correction table, use the **ET[]** command.

The array **ET[N]** is used by non-linear current feed-forward, error mapping and ECAM algorithms. The user must make sure that there is no overlapping between the **ET** array portions which are used by different algorithms. This is not checked by the drive.

Indices

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Default</th><th>Range</th></tr>
<tr><td rowspan="4">1</td><td>**Value**</td><td>**Operation**</td><td rowspan="4">0</td><td rowspan="4">0 to 6</td></tr>
<tr><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>Enable linear mode</td></tr>
<tr><td>2</td><td>Enable cyclic (modulo) mode</td></tr>
<tr><td>2</td><td colspan="2">Mechanical position source selection</td><td>0</td><td>See Notes below</td></tr>
<tr><td>3</td><td colspan="2">Reserved</td><td></td><td></td></tr>
<tr><td>4</td><td colspan="2">First index of the correction table</td><td>1</td><td>1 to 1999</td></tr>
<tr><td>5</td><td colspan="2">Last index of the correction table</td><td>2</td><td>2 to 2000</td></tr>
<tr><td>6</td><td colspan="2">Correction table position gap</td><td>10</td><td>1<**CG[6]** <(2^{52}-1)</td></tr>
<tr><td>7</td><td colspan="2">Start position, in user-defined units.</td><td>0</td><td>-2^{52} to (2^{52}-1)</td></tr>
</table>

Notes

- Correction table position gap is a position distance between two consequent points in the table in user units
- The following bit order applies to **CG[2]** command:

<table>
<tr><th>Bits</th><th colspan="3">Details</th><th>Range</th><th>Notes</th></tr>
<tr><td rowspan="2">0-3</td><td>**If Bits 8-11 = 0**</td><td colspan="2">Socket number</td><td>0 to 8</td><td rowspan="2">0 – the axis feedback position will be used as a source position</td></tr>
<tr><td>**If Bits 8-11 = 1**</td><td colspan="2">Axis number</td><td>0 to **WS[1]**</td></tr>
<tr><td rowspan="5">4-7</td><td rowspan="4">**If Bits 8-11 = 0**</td><td colspan="2">**Socket signal** used for axis function</td><td>0 to 2</td><td></td></tr>
<tr><td>0</td><td>Main socket signal</td><td></td><td></td></tr>
<tr><td>1</td><td>Auxiliary 1 socket signal</td><td></td><td></td></tr>
<tr><td>2</td><td>Auxiliary 2 socket signal</td><td></td><td></td></tr>
<tr><td>**If Bits 8-11 = 1**</td><td colspan="2">NA</td><td></td><td></td></tr>
<tr><td rowspan="3">8-11</td><td colspan="3">Reference source</td><td>0 - 1</td><td></td></tr>
<tr><td>0</td><td colspan="2">Socket</td><td></td><td></td></tr>
<tr><td>1</td><td colspan="2">Axis</td><td></td><td></td></tr>
</table>

References

ET[], NF[]

CL – Current Limit Parameters

Set and obtain the current continuous limitations and motor stuck conditions.

CANopen/CoE

Elmo Alias Object: 0x305D

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	See below
Index range	1 to 7
Default	See table below
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

CL[1] defines the maximum continuous motor phase current allowed, in amperes. This parameter is used to protect the motor from overcurrent, and the load from excessive torques. The motor current (torque) command is normally limited to its peak limit, as defined by **PL[1]**. After a short period of torque demands higher than **CL[1]** (as defined by the **PL[2]** parameter and equations in the Platinum EtherCAT and CANopen Implementation Manual), the torque command limit is decreased to **CL[1]**. The torque command remains limited to **CL[1]** until the average torque demand falls below 90% of **CL[1]** for a few seconds. **CL[1]** has no effect if **CL[1]** > **PL[1]**.

CL[2], **CL[3]** and **CL[4]** define how the motor stuck protection is handled. A stuck motor is a motor that does not respond to the applied current command due to failure of the motor, the drive system or the motion sensor.

CL[2] defines the tested torque level as a percentage of the continuous current limit **CL[1]**.

CL[3] specifies the absolute threshold main sensor speed below which the motor is considered not moving.

CL[4] defines the present threshold time for the conditions declared by **CL[2]** and **CL[3]**. If the torque level is above the **CL[2]** limit, and the main sensor speed is below **CL[3]** and, in addition, this occurred continuously for more than **CL[4]** seconds, then the motor is stuck.

If the motor is stuck, the motion fault **MF** = 2,097,152 (0x200000) is set, and the motor is aborted.

If **CL[2]** < 2, the motor stuck protection is not applied.

For other values of **CL[2]**, the motor is disabled, and **MF** is set to 0x200000 if the motor current command level exceeds a selected level for more than 3 seconds, without a change of significant motor speed (result), as defined by **CL[3]**.

CL[5] CL[6] & **CL[7]** commands handle the ability to derate the continuance current limits denoted via **CL[1]**, with respect to the temperature. When derating is activated, the temperature can be maintained below the “over temperature” threshold, preventing a fault that will disable the drive (servo off).

CL[5] defines the threshold temperature from which the current de-rating starts. When **CL[5]** is 0 (default), the continuance current derating is not operational.

CL[6] defines the rate of derating which means what current to reduce from the continuance current limit for every degree beyond the temperature threshold.

CL[7] defines the minimum limit that the continuance current will be reduced.

Indices

Index	Description	Type	Units	Range	Default
1	Continuous current limit	Float	Ampere	**0 to MC[4]**	0
2	Motor stuck current level	Float	Percent (of **CL[1]**)	0 to 100	0
3	Motor stuck speed level - velocity	Float	User defined units per second	From 0 to 10^{11}	0
4	Motor stuck time-out - duration	Float	mSec	0 to 5000	3000
5	Temperature threshold for continuous current	Float	Celsius	0…120	75
6	Current per temperature factor for continuous current	Float	Amp/Celsius	0…**MC[1]**/degree	0
7	Minimum continuous current for the de-rating. Current de-rating procedure cannot reduce current to value lower than **CL[7]**	Float	Ampere	0…**MC[4]**	0

Notes

The motor stuck protection is always applied to the velocity sensor converted to units of position sensor velocity. In dual-loop applications this protection does not pertain to failures in the auxiliary sensor.

The time constant of 3 seconds is used because almost every motion system applies high torques for short acceleration periods while the speed is slow.

Drives with the "R" option (**MC[2]=MC[4]**) allow the user to set the drive`s maximum current (**PL[1]**) without time restriction. In these drives, the **PL[2]** time restriction is not considered. **CL[1]** can also be set to **MC[4]** value.

When current derating is active (**CL[5]** > 0), the actual continuance current limit is reflected via **WF[41]**.

The **LC** command bit 2 is set when the continuance current is under de-rating due to temperature as defined above.

References

PL[*N*], **MC**, **TC**, **MF, LC**

CN[N] – Profile Conditioner

CN specifies the profile conditioner applied to the profiler command.

CANopen/CoE

Elmo Alias Object: 0x305F

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	**CN[1] - The motor must be OFF.**
Range	See below
Index range	NA
Default	See below
Unit modes	UM=2, 5, 6, 7
Non-Volatile	Yes
Axis Related	Yes

Remarks

A new **CN[N]** setting is activated at Motor Enable upon issuing the **MO**=1 command.

Indices

The following table describes the **CN[*N*]** entries.

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Unit</th><th>Default</th><th>Range</th><th>Comments</th></tr>
<tr><td rowspan="4">1</td><td colspan="2">Signal Conditioner Type</td><td rowspan="4"></td><td rowspan="4">0</td><td rowspan="4">0-2</td><td rowspan="4">The motor must be OFF</td></tr>
<tr><td>0</td><td>Disabled</td></tr>
<tr><td>1</td><td>ZV (Zero Vibration)</td></tr>
<tr><td>2</td><td>ZVD (Zero Vibration, Zero Derivative)</td></tr>
<tr><td>2</td><td colspan="2">Profile Conditioner Frequency</td><td>Hz</td><td>20.0</td><td>$\geq \frac{1000000}{2046*HS*\sqrt{(1-CN[3]^2)}}$
in ZV mode
$\geq \frac{1000000}{1022*HS*\sqrt{(1-CN[3]^2)}}$
in ZVD mode</td><td>The command does not verify the value. The actual signal conditioner frequency is limited according to the range.
Range examples:
≥2.02Hz, if HS=250 in ZV mode</td></tr>
</table>

Index	Description	Unit	Default	Range	Comments
				HS – profile sampling time in [µSec]	≥10.1Hz, if HS=100 in ZVD mode
3	Profile Conditioner Damping Ratio		0.1	0.001 to 0.25	

References

AC, **DC**, **PA**, **PR**, **JV**, **JP**

CP[N] – Set Boot Up Mode

CP[*N*] command erases user program.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Command
Source	All, excepts the user program
Restrictions	• The motor must be off, else the command returns error 57 "Motor must be off". • The user program must not be running (**PS !**= 1) else the command returns error 82 "Program is running".
Range	None
Index range	1
Default	none
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The **CP** command completely clears the user program from the non-volatile flash memory of the drive.

After the **CP** command, the program status, **PS**, should be -2.

The **CP** command might take approximately 1 second. During this time, the background is idle.

References

CC, KL, XQ, HP

CR[N] – Capture Resource Selection

CR[N] command defines which hardware capture module is used for the required capture function or capture service.

CANopen/CoE

Elmo parameters objects: 0x3063

Attributes

Attribute	Description	
Type	Unsigned Short, Read/Write	
Source	CANopen, CoE, RS232, USB, TCP, EoE	
Restrictions	None	
Range	0…16 0 no Capture module assigned 1-8 Capture Position module 1 – 8 is assigned 9-16 Capture Time module 1 – 8 is assigned	
Index range	1 to 5 – the function which requires a capture module. See table below.	
Default	**Axis 1:**	AX1.**CR[1]**: 1 AX1.**CR[2]** to AX1.**CR[5]**: 0 (see note)
	Axis 2:	AX2.**CR[1]**: 2 AX2,**CR[2]** to AX2.**CR[5]**: 0 (see note)
	Axis 3	**Platinum drive**: AX3.**CR**[1]: 2; AX3.**CR[2]** to AX3.**CR[5]**: 0 (see note) **Quartet drive:** AX3.**CR**[1]: 3; AX3.**CR[2]** to AX3.**CR[5]**: 0 (see note)
	Axis 4	**Platinum drive:** AX4.**CR**[1]: 2; AX4.**CR[2]** to AX4.**CR[5]**: 0 (see note) **Quartet drive:** AX4.**CR**[1]: 4; AX4.**CR[2]** to AX4.**CR[5]**: 0 (see note)
	Axis y:	AXy.**CR[1]**: 2 AXy.**CR[2]** to AXy.**CR[5]**: 0 (see note) *y>4 and depends on drive type
Unit modes	All	

Attribute	Description
Non-Volatile	Yes
Axis Related	Yes

Remarks

When **CR[x]** is set to 0 by user or by default, there is no Capture module assigned by the command. But on initialization of service the capture module is defined automatically in accordance with the following rule:

- Capture 1 – in the cases of DS-402 Index, HM/HF Index, TP1
- Capture 2 – in the cases of DS-402 Home, HM/HF Home, TP2

Capture services are the functions which require the hardware capture modules. When the **CR[Index]** is used the "Index" denotes the function. The following functions are available:

Index	Capture Service
1	DS-402 Homing mode
2	DS-402 Touch Probe 1 function
3	DS-402 Touch Probe 2 function
4	**Elmo Homing mode via HM[] command**
5	**Elmo capture mode via HF[] command**

The number of available capture modules depends on the drive. Typically, two capture modules are present per axis.

Functional description

Capture Position modules are used to capture the position of an input quadrature signal.

The assignment of the hardware capture to the capture function is done via the CR[N] for example, **Ax2.CR[3]=4*;*** means that Capture Position module 4 is used for Touch Probe 2 function of axis 2 .

The configuration of the capture module should be done prior of using the DS-402 homing, DS-402 touch probe, and in Elmo HM and HF functions when Index or Home switch are required.

Special case 1: Platinum Quartet Servo-drive

The Quartet drive has 4 Capture Position modules. Each Capture Position module is only connected to its own axis, and therefore to run any capture service, like TP1, TP2, DS-402 homing, HM/HF on axis 2, only Capture Position module 2 can be used.

Special case 2: Titanium Drive

The Titanium drive has 4 Capture position modules. The capture module should be selected in accordance with connected port. See following table.

Connected port	Capture module number	CR command	Remarks
Port A1	1	Ax1.CR[Index] =1	Supported by CPLD Quad 1
Port A2	2	Ax2.CR[Index] =2	Supported by CPLD Quad 2
Port B1	3	Ax1.CR[Index] =3	Supported by AM263 Eqep 0
Port B2	4	Ax2.CR[Index] =4	Supported by AM263 Eqep 2

References

HM[], **HF[]**, **GI[N]**, Administrative Guide chapters Capture, DS-402 Homing, Touch Probe

CS – Force Commutation Angle

CS forces the commutation angle to specific angle without feedback checking.

Note:
The **CS** command should be handled with extreme care as it modifies the commutation angle.

CANopen/CoE

Elmo Alias Object: **0x3064**

Attributes

Attribute	Description
Type	Long
Source	All
Restrictions	Commutation may be wrong
Range	0 to **CA[18]**
Index range	NA
Default	0
Unit modes	3
Non-Volatile	No
Axis Related	Yes
Activation	Immediate

Remarks

The CS command forces the commutation angle by bypassing the commutation procedure. Should be used in stepper mode (UM = 3) at a specific angle and sets the commutation angle in units of counts/revolution.

Example of use:

```
MO=0
UM=3
MO=1
TC=1
PA=3072; BG; // 3072 is 270 degrees for stepper mode. The commutation angle is 90 degrees from that point which means 0.
//Wait for few seconds for motor to stabilized.
```

```
CS=0 ;
MO=0
```

Handle the command with care. Incorrect commutation may cause severe damage.

References

UM, MO, PA

CT – Cyclic Thread

CT[N] Get/Set cyclic thread - the number of background loops between thread execution.

CANopen/CoE

Elmo_parameters_objects: 0x3065

Attributes

Attribute	Description
Type	Unsigned Char, Read/Write
Source	All
Restrictions	Thread 0 is non- cyclic able.
Range	0 to 250
Default	0
Index Range	1 to 6
Unit modes	All
Non-volatile	No
Attribute	No

Indices

Index	Description
1	Cyclic call of thread 0 (main).
2	Cyclic call of thread 1.
3	Cyclic call of thread 2.
4	Cyclic call of thread 3.
5	Cyclic call of thread 4.
6	Cyclic call of thread 5.

Notes

- For main (thread 0) the value of cyclic thread can be only 1 (default) .
- For thread 1 – 5 the value of cyclic thread can be any number between 1 (default) – 250.

References

WS[37], MI[], PS[]

CU – CPU Usage

CU calculates the present CPU usage.

CANopen/CoE

Elmo Alias Object: none

Attributes

Attribute	Description
Type	unsigned long, Read-only
Source	All, except CoE
Restrictions	None
Range	0 to 100
Index range	NA
Default	None
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

CPU usage indicates how much of a workload is being handled by the CPU.

A load of 100% means that the processor is fully utilized and that background tasks, such as the user program and communications, will not receive any CPU time.

A load of 50% means that the CPU is available for background tasks half of the time.

The value of **CU** is between 0 and 100%.

This command pauses the background loop for at least 1 msec. This affects the execution time of the user program, connected communication channels and other background tasks.

References

CV – Collect Values (Preliminary)

CV retrieves safety related data.

CANopen/CoE

Elmo Alias Object: 0x3067

Attributes

Attribute	Description
Type	unsigned long, read-only
Source	RS232, USB, TCP, EoE, CoE
Restrictions	None
Range	None
Index Range	1 to 32
Default	NA
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

When no value is set in a sub-index the return value is zero.

Notes:

If the drive has no Safe features built-in, all sub-indexes return 0.

General safety logics refer to logics applied on safe functions, such as “Enabling device” safe logic.

Indices

Table 3, describes the **CV[*N*]** entries. For examples refer to scenarios in Table 6.

Index	Description	Description		Notes
1	Bit 0, Safe Monitoring error indication. Bit 1 – 8: General safety status.	Bits 0:	0: No monitoring error 1: Safe function and, or safe general logic monitoring error.	If value of bit is “1”, read **CV[2]** and **CV[8]** to check which safety function or general safety logic in monitoring error.
		Bits 1-2:	0 or 3: Reset 1: Slave in FailSafe state 2: Slave in Process Data state	Drive internal FSoE state. “Drive Safe” in EAS Safety Monitoring window presents Drive’s FSoE state.

Index	Description	Description		Notes
		Bits 3:	Position sensor: 0: Not ready 1: Ready	If position sensors are not used, their indications set to ready state automatically.
		Bits 4:	Torque Sensor: 0: Not ready 1: Ready	If torque sensors are not used, their indications set to ready state automatically.
		Bits 5:	VDD Enable: 0: not ready 1: Ready	If VDD voltage is activated return value is "1", otherwise "0".
		Bits 6:	Sensor 12V: 0: Not ready 1: Ready	If 12V_ENC is stable value is "1", otherwise "0".
		Bits 7:	Enabling Device: 0: Not ready 1: Ready	If enabling device logic requested by SRA object 0x2E20[0] returned value is 1, otherwise 0.
		Bits 8:	SLP Calibration: 0: Not ready 1: Ready	If SLP absolute position calibration finished successfully value is 1, otherwise 0.
2	Safety ErrorAck Indication	Informs the user which safety functions caused the monitoring error according to the functions mapped to the safety control word, SRA **Object 0x1700**.		Each bit index in return value correlates to mapped function in **Object 0x1700**. If returned bit value is "1" the safe function in monitoring error, if bit value "0", no monitoring error.
3	Safety Control	**Return the value of control word, SRA Object 0x1700.** Bit value "0" safety function is activated. Bit value "1" safety function wasn't activated, in non-monitoring state.		
4	Safety Status	**Return value of status word, SRA Object 0x1B00.** **Each bit correlates to safe function mapped in Object 0x1B00.**		

Index	Description	Description	Notes
		Bit value “1” safe function is in monitoring state. Bit value “0” none monitoring state.	
5	Safe input	Informs the user which safety input is active by the corresponding bit of the command.	
6	Safe output	Informs the user which safety output is active by the corresponding bit of the command.	
7	FSoE state	Informs the user the current state of the FSoE state machine.	**FSoE values and states:** 0x0000 - Reset 0x0001 - Session 0x0002 - Connection 0x0003 - Parameter 0x0004 - ProcessData 0x0005 – FailSafeData “FSoE Control” in EAS Safety Monitoring window presents FSoE state machine.
8	Safety ErrorAck general logic Indication	Informs the user which safety general logic caused the monitoring error. Each bit corresponds to specific general safety logic. Refer to Table 4 below, bits index correlation to general safety logic.	Returns 0 - when no monitoring error. Returns 1 if general logic in correlated bit index in monitoring error state.
9	**Reserved from firmware version 2.0.1.2 and above.**	Bits 0-1: 0 or 3: Reset 1: Slave in FailSafe state 2: Slave in Process Data state	
		Bits 2: Position sensor: 0: Not ready 1: Ready	
		Bits 3: Torque Sensor: 0: Not ready 1: Ready	
		Bits 4: VDD Enable:	

Index	Description	Description		Notes
			0: not ready 1: Ready	
		Bits 5:	Sensor 12V: 0: Not ready 1: Ready	
		Bits 8:	SLP Calibration: 0: Not ready 1: Ready	
10	FSoE Control	Informs the Safe Control requests sent from the FSoE master.		The FSoE control word indicates if safe function activation request by FSoE master. Bit value 0: function activated by FSoE master. Bit value 1: function not activated by FSoE master.
11	Primary Sensor initial Diagnostics errors	Informs the user for initial primary safe diagnostics sensors errors. LSB error for sensor 1, 2nd byte (bits 8-15) for sensor 2 error. To indicate the errors the sensors ID numbers needs to be discovered by SRA **Object 0x3790[0]**, sensors feedback object.		Primary safe initial diagnostics sensors errors are shown in the “Safety Diagnostics Errors Primary” section in Safety Monitoring in EAS.
12	Secondary Sensor initial Diagnostics errors	Informs the user for initial secondary safe diagnostics sensors errors. LSB error for sensor 1, 2nd byte (bits 8-15) for sensor 2 error. **To indicate the errors the sensors ID numbers needs to be discovered by SRA Object 0x3790[0], sensors feedback object.**		Secondary safe initial diagnostics sensors errors are shown in the “Safety Diagnostics Errors Secondary” section in Safety Monitoring in EAS.
13		Informs the user of the torque safety sensor errors in primary and secondary CPU.		Bits 0 to 7 torque sensor error indication to secondary CPU, Bits 8 to 15 torque sensor errors indication to primary CPU.

Index	Description	Description	Notes
			Torque safe diagnostics sensor errors are shown in the "Safety Monitoring General Error" section in Safety Monitoring in EAS.
14-32	Reserved		

Table 3 CV[N] entries description.

CV[8] – bit-index	General logic	Monitoring Error
1	Enabling safety logic	"ENABLING DEVICE ACTIVATION TIMER ELAPSED"
2		
3		
4		

Table 4 mapping of General safe logics with CV[8] bits index.

Examples

This part gives examples to return values of each **CV** command described in Table 3. Table 5 describes safe functions mappings in SRA **Objects 0x1700** RPDO and **0x1B00** TPDO used for next examples.

Sub-index	Safety function	Hex view	Description
0x1B00[0]/0x1700[0]	-	0x20	Number of sub-indexes
0x1B00[1]/0x1700[1]	-	0xE7000108	Safe command word
0x1B00[2]/0x1700[2]	STO	0x66400001	Safe Torque Off
0x1B00[3]/0x1700[3]	SS1	0x66500101	Safe Stop 1, instance 1
0x1B00[4]/0x1700[4]	SOS	0x66680101	Safe Operation Stop, instance 1
0x1B00[5]/0x1700[5]	SLS	0x66900101	Safe Limited Speed, instance 1
0x1B00[6]/0x1700[6]	Safe Input 2	0x2E120201	Safe Input, instance 1
0x1B00[7]	Safe Output 1	0x2E130101	Safe Output, instance 1:

Table 5: 0x1700 RPDO and 0x1B00 TPDO mapping.

Examples

Sub-index	Value	Scenario	Note
CV[1]	0x007D	**Bit 0:** 0x1– safe function in monitoring error.	At least on monitoring error detected

Sub-index	Value	Scenario	Note
		Bit 1-2: 0x2 – Slave in process Data **Bit 3:** 0x1–Feedback sensors in ready state **Bit 4:** 0x1–Torque sensor in ready state **Bit 5:** 0x1–IO VDD voltage in ready state **Bit 6:** 0x1–12V activated and in ready state	
CV[2]	0x0000000C	SLS and SOS in monitoring error state.	The corresponding bits of both the SLS and SOS functions is in logic level high,”1”.
CV[3]	0xFFFFFFFB	Only SLS function is activated and in monitoring state.	Activating the function means the corresponding mapped bit is in level logic low ,“0”.
CV[4]	0x00000008	SLS function in monitoring state.	Safe function status in “1” if the function is activated and in monitoring state.
CV[5]	0x00000002	Safe input 2 is active.	Safe input 2 switch is closed, input in high state.
CV[6]	0x00000001	Safe output 1 is active.	Safe output 1 in high state.”1”.
CV[7]	0x0004	Driver/Slave in “ProcessData” state.	0x0004 - Process data.
CV[8]	0x01	“Enabling Device” General logic in monitoring error.	Refer to Table 4 above, to cross bit index to general safety logic.
CV[9]		For FW version 2.0.1.1 and older/lower versions. **Same as Example for CV[1].**	Reserved from firmware version 2.0.1.2 and above.
CV[10]	0x00000007	FSoE control word activates SLS function.	SLS activated by FSoE master.
CV[11]/CV[12]	**CV[11] / CV[12] =0x0402** **0x3790[0] = 0xC**	Primary/secondary feedback sensor in error state. Feedback sensors are Quad port A, Quad port B.	**Sensor 1 error:** “ Bit 1 - Incorrect setting detected in control register” **Sensor 2 error:** “Bit 2 - Incorrect state transition was detected”

Sub-index	Value	Scenario	Note
CV[13]	**CV[13] = 0x0408**	Primary and secondary torque safety sensor error.	Error reported to primary CPU: *"Bit 2 - Incorrect setting detected in Sigma Delta control register"* Error reported to secondary CPU: *"Bit 3 - Phase Current compare error"*.

Table 6: Scenarios example based on table 1.

References

CW – Controlword

CW[N] provides access to DS-402 Controlword (objects 0x6040, 0x6840, 0x7040, 0x7840, 0x8040, 0x8840).

CANopen/CoE

Elmo parameters objects: 0x3068

Map able objects to RPDO: 0x6040, 0x6840, 0x7040, 0x7840, 0x8040, 0x8840

Attributes

Attribute	Description
Type	Unsigned short, Read/Write
Source	All
Restrictions	None
Range	0 to 65535
Index Range	0,1
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Indices

Write access to **CW[0]** returns error code 3 “Bad index”

Read access to **CW[0]** returns 1 (number of sub-indexes)

CW[1] provides read and write access to DS-402 Controlword

Remarks

None

References

SW

CZ – Non-Volatile Parameters CRC

CZ retrieves the CRC of the last saved non-volatile parameters of the drive.

CANopen/CoE

Elmo Alias Object: 0x306B

Attributes

Attribute	Description
Type	unsigned long, Read-only
Source	All
Restrictions	None
Range	2^{32}-1
Index range	NA
Default	None
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The non-volatile parameters of the drive are saved to flash memory via SV command. A 32 bits CRC is calculated and saved as part of the parameter. The CRC is checked when the parameters are uploaded from the flash memory to the RAM via the **LD** command and after power up cycle. The **CZ** command retrieves the saved CRC. If the last save command failed the CZ results in 0.

References

DC – Set Deceleration

DC specifies the maximum allowed deceleration of the main DS-402 profiler in Profiler Position mode. Decelerations of the superimposed, phasing and socket profilers are equal to the accelerations and are defined by **AC[N]** command (**N**=2,3,4)

CANopen/CoE

DC[1] Profile_acceleration: 0x6084

Elmo Alias Object: 0x3078

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	**Effective on the next call to BG[N]**
Range	$1 \ldots 10^{14}$
Index range	1 to 5
Default	1,000,000,000
Unit modes	**UM = 2, 3, 5, 6, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

For compatibility reasons setting a reserved index will not return an error. The drive ignores such settings.

The **DC** command defines the maximum allowed deceleration during the operation of point-to-point (**PA[1]**, **PR[1]**) and jog (**JV**, **JP[1]**) profilers.

The **DC** command does not affect the present motion. It takes effect only on the next Begin Motion (**BG[1]**).

The **DC** value is fed to object 0x6084 during power-up and when the Begin Motion (**BG[1]**) command is used.

The **DC** command does not affect time-dependent motion, such as Interpolated Position or Cyclic Synchronous Mode (see MAN-P-Administrative Manual).

The acceleration and deceleration of the main profiler are subject to the limits of the **SD** value. If the **DC[1]** value is higher than the **SD** value, the **SD** value is used, and the **DC[1]** value is ignored.

The **DC** value can be given in user-defined units specified by the **FC** command.

References

AC, **SD**, **BG**, **PA**, **PR**, **JV**, **FC**

DD[] – Get CAN Controller Status

DD[] reads values from the drive's CAN controller status.

CANopen

Elmo Alias Object: 0x3079

Attributes

Attribute	Description
Type	Unsigned short, Read-only
Source	All
Restrictions	None
Range	See the table below.
Index range	1
Default	See the table below
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The **DD[]** command reads Error and Status register of HW CANM controller. The values are described in the table below.

Bit	Name of the field	Reset Value	Definition
0 – 2	Last error code (LEC)	7	0 – No error The field indicates the type of the last error on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error.
			1 – Stuff error More than five equal bits in a row have been detected in a part of a received message where this is not allowed.
			2 – Form error A fixed format part of a received frame has the wrong format.
			3 – Acknowledge error The message this CAN core transmitted was not acknowledged by another node.

Bit	Name of the field	Reset Value	Definition
			4 – Bit 1 error During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant.
			5 – Bit 0 error During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (logical value '0'), but the monitored bus level was recessive. During Bus-Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus-Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed)
			6 – CRC error In a received message, the CRC check sum was incorrect. (CRC received for an incoming message does not match the calculated CRC for the received data).
			7 – No CAN bus events No CAN bus event was detected since the last time the CPU read the error and status register. Any read access to the error and status register re-initializes the LEC to value '7.'
3,4			Reserved
5	Error passive state (EPASS)	1	0 - On CAN Bus error, the DCAN could send active error frames. 1 - The CAN core is in the error passive state as defined in the CAN Specification.
6	Warning state (EWARN)	1	0 - Both error counters are below the error warning limit of 96 1 - At least one of the error counters has reached the error warning limit of 96.
7	Bus off state (BOFF)	0	0 – The CAN module is not bus-off state. 1 – The CAN module is in bus-off state.
8	Parity error (PER)	0	0 – No parity error has been detected since last read access. 1 – The parity check mechanism has detected a parity error in the Message RAM. This bit will be reset if error and status register is read.
9 - 15			Reserved

References

AG[], AS[], BV, MC

DH –Initiate Homing Motion

DH command initiates a Homing motion procedure which is based on the DS-402 Homing Mode.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Unsigned short, Read/Write
Source	All
Restrictions	• Motor must be on • Effective on the next call to **BG**
Range	0,1
Index range	1
Default	0
Unit modes	**UM = 2, 5,6,7 –All homing methods are available** **UM = 1,3 – Only methods 35,37 are available**
Non-Volatile	No
Axis Related	Yes

Values

The following table describes the possible values and the modes associated with them.

Value	Description
0	Do not activate homing
1	**Activate homing. The value is reset automatically after BG**

Remarks

DH followed by **BG** command, results-in a motion to a home search based on the following commands:

- For DS-402 homing:

Command	Command description
GI[]	capture input MUX selection
CR[]	Capture resource selection
0x607C[0]	Home offset
0x6098[0]	Homing method
0x6099[1]	Homing high velocity

Command	Command description
0x6099[2]	Homing low velocity
0x609A[0]	Homing acceleration

- For PLC open Home On Block (methods -1,-2,-3,-4): - additional commands will be defined

0x3620[1]	Torque limit
0x3620[2]	Time limit
0x3620[3]	Distance limit
0x3620[4]	Detection Velocity Limit
0x3620[5]	Detection Velocity Time Limit

The following possible homing modes are detailed in the DS-402 standard

Method	Description
Method 1	Homing on RLS and Index Pulse
Method 2	Homing on FLS and Index Pulse
Method 3	Homing on Positive Home Switch and Index Pulse
Method 4	Forward Homing on Positive Home Switch and Index Pulse
Method 5	Forward Homing on Negative Home Switch and Index Pulse
Method 6	Reverse Homing on Negative Home Switch And Index Pulse
Method 7	Reverse Homing on Home Switch/FLS and Index Pulse
Method 8	Forward Homing on Home Switch/FLS and Index Pulse
Method 9	Reverse Homing on Positive Home Switch/FLS and Index Pulse
Method 10	Forward Homing on Negative Home Switch/FLS and Index Pulse
Method 11	Forward Homing on Negative Home Switch/RLS and Index Pulse
Method 12	Reverse Homing on Positive Home Switch/RLS and Index Pulse
Method 13	Forward Homing on Positive Home Switch/RLS and Index Pulse
Method 14	Reverse Homing on Negative Home Switch/RLS and Index Pulse
Methods 15 and 16	Reserved
Method 17	Homing on RLS
Method 18	Homing on FLS
Method 19	Reverse Homing on Negative Home Switch
Method 20	Forward Homing on Positive Home Switch
Method 21	Forward Homing on Negative Home Switch
Method 22	Reverse Homing on Positive Home Switch
Method 23	Reverse Homing on Negative Home Switch/FLS

Method	Description
Method 24	Forward Homing on Positive Home Switch/FLS
Method 25	Reverse Homing on Positive Home Switch/FLS
Method 26	Forward Homing on Negative Home Switch/FLS
Method 27	Forward Homing on Negative Home Switch/RLS
Method 28	Reverse Homing on Positive Home Switch/RLS
Method 29	Forward Homing on Positive Home Switch/RLS
Method 30	Reverse Homing on Negative Home Switch/RLS
Methods 31 and 32	Reserved
Methods 33	Homing in a negative direction on the Index Pulse
Methods 34	Homing in a positive direction on the Index Pulse
Method 35	Homing on the current position
Methods -1	PLC open Home on Block Homing on block (against the wall) in a positive direction.
Methods -2	PLC open Home on Block Homing on block (against the wall) in a negative direction.
Methods -3	PLC open Home on Block Homing on block (against the wall) in a positive direction and Index pulse.
Methods -4	PLC open Home on Block Homing on block (against the wall) in a negative direction and Index Pulse.

On the next **BG** after applying **DH** command:

- **Object 0x6061[0]** is set to 6 (Profile Homing)
- Axis starts motion as described in the table above
- The status of the motion can be retrieved via **0x6041[0]** ((refer to the MAN-P-ADMINGUIDE, Homing Modes chapter))

References

BG, **UM**

DN – Drive User Name

DN[N] specifies asci character array for the user defined drive name.

CANopen/CoE

Objects: 0x3083

Attributes

Attribute	Description
Type	Parameter, char, Read/Write
Source	All
Restrictions	None
Range	0 to 126
Default	"Elmo Drive"
Index Range	1 to 64
Unit modes	All
Non-Volatile	Yes
Axis Related	No
Attribute	None

Remarks

The drive name is a NULL terminated string, that can be saved in the drive.

DN command allow setting up to 64 characters for drive name. Each **DN** sub index corresponds to asci character in the drive name.

The drive name can be read as visible string via **VR** command.

For example:

Setting DN array:

DN[1]=69; **DN[2]**=108; **DN[3]**=109; **DN[4]**=111; **DN[5]**=32;
DN[6]=68; **DN[7]**=114; **DN[8]**=105; **DN[9]**=118; **DN[10]**=101; **DN[11]**=0;

Will be read via the **VR** command:

"Elmo Drive"

References

VR[]

DP – Double Precision

DP determines the precision in floating point presentation to the user. The precision is defined as number of digits appear after the decimal point – to the right to the point.

CANopen/CoE

Elmo Alias Object: 0x3085

Attributes

Attribute	Description
Type	Short (2 bytes) – Read-Write
Source	All
Restrictions	None
Range	-1 to 6
Index Range	Array 2 cells : indexes 0 – 1
Default	-1
Unit modes	All
Non-Volatile	Yes (Save to Flash)
Axis Related	No

Example of usage

Suppose a floating point value is : 123.123456

- **DP** = -1 (default):
  The value remains the same (full precision) : 1.231235e+002
- **DP** = 0 (No digits after decimal point):
  The value : 123
- **DP** = 1 :
  The value : 123.1
- **DP** = 2:
  The value : 123.12
- **DP** = 3:
  The value : 123.123
- **DP** = 4:
  The value : 123.1235
- **DP** = 5:
  The value : 123.12346
- **DP** = 6:
  The value : 123.123456

Remarks

- The **DP** variable is saved to flash and is defined as post process parameter which is set to its default value ('-1' = full precision) when the project is initiated at first.
- The internal values of floating point variables (float and double) remain the same – full precision, the same is for parameters values in parameters xml. (Those values representation is regardless to the value of DP command.

The maximum value of DP command is '6' because this is the maximum number of digits that the function *sprintf* prints after the decimal point.

Index in DP	Description
0	The maximal sub index ('1').
1	Number of digits to present after the decimal point.

DV[] – Desired Value

DV[] returns the desired value to the controller. The desired value is actually the command for the present control cycle derived from the profiler. The value in the case of velocity and position is in user defined units, namely, UU/sec and UU.

CANopen/CoE

Alias Objects: 0x3063

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	None
Range	See table below
Index range	1 to 18
Default	0
Unit modes	All with respect to the relevant control loop
Non-Volatile	No
Axis Related	Yes

Remarks

When the motor is disabled, the controller is not active and the desired value is 0.

Indices

The following table describes the **DV[]** entries.

Index	Description	Units	Notes
1	Total current command	Ampere	Actual current reference to the current controller, including all
2	Total velocity command	UU/sec	Actual speed reference to the speed controller, including the output of the position controller
3	Total position command	UU	Actual position reference to the position controller
4	Velocity command from the socket reference	UU/sec	The socket reference portion of the speed command
5	Main profile velocity command	UU/sec	The software controlled portion of the speed command to the controller

Index	Description	Units	Notes
6	Position command from the socket reference	UU	The socket reference portion of the position command
7	Main profile position command	UU	The software controlled portion of the position command to the controller
8	Field angle	Electrical angle *4096/360	**Field (commutation angle), if UM=1/2/5** **Stepper angle command, if UM=3/6/7**
9	Current command from the socket reference	Ampere	The socket reference portion of the torque command
10	Main profile current command	Ampere	The software controlled portion of the torque command to the controller
11	Yaw current command	Ampere	
12	ECAM table input	UU	The input to the ECAM table (after ratio)
13	Superimposed profiler position reference	UU	
14	Phasing profiler position reference	UU	
15	Socket profiler position reference	UU	
16	Superimposed profiler velocity reference	UU	
17	Phasing profiler velocity reference	UU	
18	Socket profiler velocity reference	UU	

References

E1….E8[N] – ECAM/Error Correction Sub Tables

E1…..E8[*N*] is used to address the **2000 e**ntries of the ECAM/Error Correction position array **ET[K]**.

CANopen/CoE

Elmo Alias Object: 0x3091…0x3098

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restriction	None
Range	-2^{52} to $(2^{52} - 1)$
Index range	1 to 250
Default	None
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

The following table describes the relationship between **ET[K]** and **E1….E8[*N*]** commands:

E# command	ET command	CANopen object
E1[1…250]	**ET[1…250]**	0x3091
E2[1…250]	**ET[251…500]**	0x3092
E3[1…250]	**ET[501…750]**	0x3093
E4[1…250]	**ET[751…1000]**	0x3094
E5[1…250]	**ET[1001…1250]**	0x3095
E6[1…250]	**ET[1251…1500]**	0x3096
E7[1…250]	**ET[1501…1750]**	0x3097
E8[1…250]	**ET[1751…2000]**	0x3098

- The user must make sure that there is no overlapping between ECAM and error correction portions of the **E1….E8[*N*]** arrays.
- When the motor is enabled (**MO**=1) and the ECAM table is controlling the motion ((**EM[1]=1 or EM[1]=2**) and **RM=1**), the ECAM table entries can be changed "on the fly",
with the exception of the **ET[EM[8]-2]…ET[EM[8]+2]** entries. If during these entries, a change is made "on-the-fly", an error is produced.

References

EM[*N*],PC[N]

EA[N] – Emulation Parameters

EA[*N*] enables the configuration and activation of feedback emulation. The command is non-axis related.

CANopen/CoE

Elmo Alias Object: 0x309A

Attributes

Attribute	Description
Type	Long, Unsigned Short
Source	USB, RS232, TCP, EoE
Restrictions	According to the array index
Range	According to the array index
Index range	1 to 17
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The emulation function emulates any feedback/encoder (socket) readings to one of the following waveform formats on Port-C A/B/I outputs:

- AqB quadrature format. Channel A – Port C_A, channel B – Port C_B
- Pulse/direction format. Pulses – Port C_A, direction – Port C_B
- Up/down format. Up Pulses – Port C_A, Down Pulses – Port C_B
- Hall signal format, TBD
- PWM 100% and Direction format. Pulses – Port C_A, Direction – Port C_B
- PWM 50% format. Pulses – Port C_A, Direction – Port C_B (optional)

Emulation is supported by the specific drive revision, starting from TCON-based Rev C (**WS[8]** command returns value 6).

The emulation mode requires hardware routing to PORT C. This is done via the **GC[]** command. Function 5 is used for the emulation (e.g. **GC[1]**=5 for Port C channel A, **GC[2]**=5 for Port C channel B). Refer to the **GC[]** command for more details.

If sockets which are used by the emulation are changed during the emulation operation, the emulation is terminated. The value of **EA[1]** is set to 0 (disabled).

The various emulation modes (excluding PWM) require internal emulation feedback to ensure that all required pulses are transmitted. For compatibility reasons and in situations where **EA[8]** is set to 1, a socket must be used as a feedback. This is done by setting with commands **S1[2]…S8[2]** bits 1-3, value = 5. Where **EA[8]** is 0 and internal feedback is used, there is no need to select a socket for this purpose.

If the following error between the emulated socket and the emulation position is greater than ±1,000,000,000, the emulation will automatically stop, and **EA[1]** will report -1. Afterwards, the user must disable the emulation, i.e., must set **EA[1]** = 0, before re-enabling it.

PWM emulation does not require emulation feedback.

The PWM emulation includes 100% and direction (using 2-pin wires, for pulse and direction) and 50% (single pin wire for pulse only) modes. The 100% mode denotes that the complete range of PWM is used for signal emulation while the direction (positive or negative) requires another signal. The benefit is the higher resolution (relative to 50% mode) while the drawback is the second wire required for direction.

The 50% mode indicates that when the signal is 0, the PWM will be 50%. The higher positive value of the input signal results in 100% PWM while the highest negative value results in 0% PWM. In PWM 50% mode Port C channel A is used for pulses. Optionally, if **GC[2]** is set to value 5, Port C channel B presents source signal direction

The difference between the 100% and 50% modes is mainly related to the resolution. The algorithm and functionality of both modes are handled similarly.

PWM emulation can be linked to Motor Enable (**SO**) via bit 1 in **EA[3]** as follows:

Motor Status	Results
Enabled	The emulation automatically starts. Note that the emulation will start ~100µSec after the servo is enabled (**SO=1**).
Disabled	The emulation is automatically suspended. In this case, the PWM duty cycle is set to value "medium" that is related to medium value of emulated signal.

For example, if the source signal is defined as Velocity (**EA[9]**=1), velocity range is symmetric (**EA[12]** = -**EA[13]**) and PWM range is symmetric (**EA[16]**=(100000-**EA[17]**)) then duty cycle will be set to:

in PWM 100 mode to (**EA[16]**/1000)%

in PWM 50% mode to 50%

Indices

The following table describes the **EA[*N*]** entries. Values are generally Read/Write if not mentioned otherwise.

Index	Description		Type	Default	Notes
1	**Value**	**Emulation Output**	Long	0	
	-1	Read-only. Emulation error absolute value is greater than 1e9			
	0	Emulation disable			
	1	Quadrature wave signals			

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Type</th><th>Default</th><th>Notes</th></tr>
<tr><td></td><td>2</td><td>Pulse/Direction wave signals</td><td></td><td></td><td></td></tr>
<tr><td></td><td>3</td><td>Up/Down wave signals</td><td></td><td></td><td></td></tr>
<tr><td></td><td>4</td><td>Hall signals</td><td></td><td></td><td></td></tr>
<tr><td></td><td>5</td><td>PWM mode 100% & Direction</td><td></td><td></td><td></td></tr>
<tr><td></td><td>6</td><td>PWM mode 50%</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td colspan="2">Emulation pulse width N in up/down or pulse/direction waves
N = 1 to 65534 pulse width is N * 20.0 [nsec], i.e. Pulse width is 0.02– 1310.680 microseconds</td><td>Unsigned Short</td><td>10</td><td></td></tr>
<tr><td>3</td><td>Bit</td><td>Description</td><td>Long</td><td>0</td><td></td></tr>
<tr><td></td><td>0</td><td>Emulation direction
0: Direction similar to the emulated encoder
1: Direction of emulated output is inversed</td><td></td><td></td><td></td></tr>
<tr><td></td><td>1</td><td>For PWM modes only, the emulation can be bound to Servo Enable (SO command)
0: Emulation is not bind to SO
1: Emulation binds to SO. When SO=0 and source signal range and PWM range are symmetric:
In PWM 100% and Direction mode duty cycle is set to EA[16]/1000 %.
In PWM 50% mode, duty cycle is set to 50%.
Note, Servo bind has no sense in the case where PWM source signal is defined as Velocity/Current socket (EA[4]=0 and EA[9]=2 or EA[9]=5)</td><td></td><td></td><td></td></tr>
</table>

Index	Description	Type	Default	Notes
4	Emulated socket number depends of emulation mode: **In AqB, Pulse/Dir, Up/Dwn, Hall modes (EA[1]=1-4)**: • The value is bit field, where: Bits 0…3 – Socket number Bits 4…7 – Signal number: 0 – Main socket signal 1,2 – Aux 1,2 socket signals Bits 9…31 – Reserved, set to 0 **In PWM modes (EA[1]=5 or 6)**: • When source signal is "velocity socket" (**EA[9]=2**) and emulated socket is set to 0 (**EA[4] = 0**), the velocity is derived from socked defined by **SA[2]**. If this socket is not defined (**SA[2]** <= 0 or > Max socket number) Emulation Error 5 is released (**EE[17]=5**, **EC=107**). • **When source is "current socket" (EA[9]=6)** and emulated socket is set to 0 (**EA[4] = 0**), the current is derived from socket defined by **SA[4]**. If it is not defined, the emulation error 6 is released (**EE[17]=6**, **EC=107**).	Long	1	If **EA[8]**!=0, then value must differ from **EA[5]**.
5	These socket ID values indicate emulation quadrature feedback sockets. The value is updated during the initialization of the socket with commands **S1[2]…S8[2]** bits 1-3, value 5. Applicable only to **EA[8]**=1 If **EA[8]=1** and there is no command **S1[2]…S8[2]** that has value 5 in bits 1-3 , then an emulation error 3 "FB socket not initialized" is released **(EE[17]=3; EC=107**) on initialization of emulation types Quadrature wave signals, Pulse/Direction wave signals and Up/Down wave signals (**EA[1]**=1,2,3). For PWM emulation types **EA[5]** is not relevant	Long	0	Read-only Write access has no effect and does not return an error
6	Emulation multiplier N = 1 to 2147483647 defines the emulation multiplier. The number of emulated encoder	Unsigned Long	1	

<table>
<tr><th>Index</th><th>Description</th><th>Type</th><th>Default</th><th>Notes</th></tr>
<tr><td></td><td>pulses and, as a result, the velocity value will be multiplied by N =EA[6]/$2^{EA[7]}$</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>Emulation scale factor
N = 0 to 30 defines the emulation scale factor 2^N. The number of emulated encoder pulses and velocity value will be divided by 2^N.</td><td>Long</td><td>0</td><td></td></tr>
<tr><td>8</td><td>Not relevant for PWM emulation modes.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Use internal emulation feedback</td></tr>
<tr><td>1</td><td>Use AqB socket for emulation feedback</td></tr>
</table></td><td>Long</td><td>0</td><td></td></tr>
<tr><td>9</td><td>Emulation Source selector for PWM mode only
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Velocity Command, The input of the velocity controller</td></tr>
<tr><td>2</td><td>If EA[4]=0, Velocity socket is defined by SA[2]
If EA[4]=1-8, Velocity socket is main FB socket for axis 1-6</td></tr>
<tr><td>3</td><td>Reserved</td></tr>
<tr><td>4</td><td>Current Command. The input of the current controller</td></tr>
<tr><td>5</td><td>Current feedback</td></tr>
<tr><td>6</td><td>If EA[4]=0, Current socket is defined by SA[4]
If EA[4]=1-8, Current socket is main FB socket for axis 1-6</td></tr>
<tr><td>7</td><td>User manual source signal that is set by TW[70] command in the range of
-100000 … 100000</td></tr>
</table></td><td>Long</td><td>1</td><td></td></tr>
<tr><td>10</td><td>PWM frequency in Hz.</td><td>Long</td><td>3000</td><td>Permitted range is 1000…200000</td></tr>
<tr><td>11</td><td>Emulated axis for PWM mode only</td><td>Long</td><td>1</td><td>Permitted range is 1…6</td></tr>
</table>

Index	Description	Type	Default	Notes
12	Speed range minimum value. In the case when source signal is Velocity Command (**EA[9]=1**) the value is saturated to –**VH[2]**. The PWM will be saturated to "Min PWM" (**EA[16]**) when Speed (or Speed command) is lower than this value.	Long	25e6	Units: If **EA[9]**=1 User Units / sec If **EA[9]**=2 Counts/sec
13	Speed range maximum value. In the case when source signal is Velocity Command (**EA[9]=1**) the value is saturated to **VH[2]**. The PWM will be saturated to "Max PWM" (**EA[17]**) when Speed (or Speed Command) is higher from this value.	Long	25e6	Units: If **EA[9]=1** User Units / sec If **EA[9]=2** Counts/sec
14	Current range minimum value. In the case when source signal is Current Command (**EA[9]**=4) or Current Feedback (**EA[9]**=5) the value is saturated to –**PL[1]**. The PWM will be saturated to "Min PWM" (**EA[16]**) when Current (or Current Command) is lower than this value.	Long	0	Units: Milli Ampere
15	Current range maximum value. In the case when source signal is Current Command (**EA[9]**=4) or Current Feedback (**EA[9]**=5) the value is saturated to **PL[1]**. The PWM will be saturated to "Max PWM" (**EA[17]**) when Current (or Current Command) is higher than this value.	Long	0	Units: milli Ampere
16	PWM minimum range. The PWM emulation duty cycle is limited to this range.	Long	0	1/1000 of %
17	PWM maximum range. The PWM emulation duty cycle will not exceed this range.	Long	100000	1/1000 of %

Example 1, Pulse – direction emulation:

To define Socket 1 as Analog Sin/Cos signal (perform by miscellaneous commands) set:

S1[2]=1 Port B, Quad 2, Sensor signal inverse

S1[1]=3 Analog Sin/Cos

To define Socket 1 as emulated source socket set:

GC[1]=5;GC[2]=5; Port C A and B used as emulation signals

EA[4]=1; Socket 1 is emulated source signal socket

EA[3]=0; Emulation direction non-invert

EA[2]=5; Pulse width is 100ns

TW[1]=2000 Emulation error gain

EA[8]=1; Use FB socket

To define Socket 2 as FB AqB socket set:

S2[1] = 0 reset socket

S2[2]=26 Port B, Quad 2

S2[1]=11 Pulse/Direction type

S2[5] must be set to non-zero value 40 … 5140

To start emulation set:

EA[1]=2 Pulse/Direction wave signals

Example 2, AqB Emulation Example:

GC[1]=5;GC[2]=5; Port C A and B used as emulation signals

EA[4]=1; Socket 1 is emulated source signal socket

EA[3]=0; Emulation direction non-invert

EA[2]=5; Pulse width is 100ns

TW[1]=2000; Emulation error gain

EA[8]=1; Use FB socket

S2[1]=0; reset socket 2

S2[2]=26; Port B, Quad 2

S2[5]=1953; glitch filter

S2[1]=1; AqB type

EA[1]=1; Start emulation

Example 3, PWM 100 and Direction Emulation:

GO[14]=5; GO[15]=5; Port C A and B used as emulation signals

EA[4]=1; Socket 1 is emulated source signal socket

EA[11]=1; Axis 1 is emulated source signal axis

EA[3]=0; Emulation direction non-invert

EA[9]=1 Source signal is Velocity Command, The input of the velocity controller

EA[12]=-20000; Speed min value

EA[13]=20000; speed max value

EA[10]=2000; PWM frequency is 2kHz

EA[16]=5000;	Permitted low PWM value is 5%
EA[17]=95000;	Permitted high PWM value is 95%
EA[1]=5;	Start Emulation PWM 100 type

References

CA[], **GO[]**, **OL[]**, **VP**

EC[] – Error Codes

EC[] specifies the interpreter and communication error code.

CANopen/CoE

Elmo Alias Object: 0x309c

Attributes

Attribute	Description
Type	Integer, Read/Write
Source	All, except the user program
Restrictions	None
Range	0 to 255
Index range	NA
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The EC command reports the error code of the last accepted command that returned an error. When the processing of a command fails, the error code is returned immediately with a question mark in the response to that command.

For example: **HX[2]** = 1;

The maximum sub-index of **HX** command is '1' thus, sub-index '2' will generate the error code 3?;

Here 3 is the number of the error code (BAD_INDEX). The number is not a printable ASCII table.

The question mark (?) means that an error occurred in the last command.

The semicolon (;) is a terminator and means that the interpretation was completed.

The **EC** command returns a printable (ASCII) value of the error code.

EC always keeps the last error code, which will be overwritten when the next error occurs. **EC** does not update when the command is completed with no error.

EC=0 clears the **EC**.

Error Codes

Refer to chapter **Drive Error Codes** for the error descriptions related to the error code.

References

EE[N]

EE[] – Extended Error

EE[N] reports detailed error codes according to a specific feature described below.

CANopen/CoE

Elmo Alias Object: 0x309E

Attributes

Attribute	Description
Type	Unsigned Long
Source	All
Restrictions	Read, Write
Range	None
Index Range	1 to 20
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

- A successful application of the **OJ** command will not modify **EE[9]**. Therefore, the last error will remain until reset to 0 by the user.
- Setting 0 (or any other value) to **EE[9]** resets the last error.

Indices

The following table describes the **EE[N]** entries:

Index	Description	Values
1	Detailed information on sensor parameter setting errors: If the **EC** command returns an error code of 79 (sensor settings error), then **EE[1]** will specify the parameter causing the error. Details the sensor setting errors	See the table for **EE[1]** below.
2	Profiler initialization error, relevant in the case of **EC=74**, which indicates an error during profiler initialization.	Returns the number of the object that caused the error. See the table for **EE[2]** below.
3	Download procedure error, relevant in the case of **EC=81**, which indicates an error during download procedure.	Returns a value that indicates the cause of the error. See the table for **EE[3]** below.

Index	Description	Values
4	When SDO (used in CANopen/ECAT) returns a general abort message 0x08000000 which indicates an error while processing the SDO command, **EE[4]** provides further details about the error.	Returns ELMO error value according to the Drive and Sensor General Error Lists. If the value is -1, no further information can be provided
5	Motor enable process failed.	Returns a value that indicates the cause of the motor enable fail. See the Drive and Sensor General Error Lists
6	ECAM initialization error, relevant in the case of EC=27, which indicates an error during ECAM initialization.	Returns a value that indicates the cause of the error. See the table for EE[6] below.
7	Upload procedure error, relevant in case of **EC=17**, which indicates an error during upload procedure.	Returns a value that indicates the cause of the error. See the table for **EE[7]** below.
8	WB (Bulk) command error, relevant in case **EC=247**, which indicates an error in the bulk command processing.	Returns a value that indicates the cause of the error. See the table for **EE[8]** below.
9	OJ (object) command error, relevant in case EC=248, which indicates and error in **OJ** command processing.	Returns a value that indicates the cause of the error. See the table for **EE[9]** below.
10	In case group command handling error, the EC reflects the error and **EE[10]** reflects the first axis which the command failed.	Check **EE[10]** for the axis number.
11	Parameters XML creation (upload) error. The EC command value is 17, indicating an error during the upload procedure and has redirection to **EE[7]** which contains the value 8. The parameters XML failed and has a redirection to **EE[11]** for details of the error of the parameters XML creation process- this error denotes a bad format of the XML. **Note:** If an error occurred with a specific parameter, the detailed error with object index, sub-index, axis number and specific error code will be retrieved via **CD[2]** command.	Returns a value that indicates the cause of the error. See the table for **EE[11]** below.
12	Primary CPLD burn process fails.	Reserved
13	Secondary CPLD burn process fails.	Reserved
14	SIL initialization fails. Bit 20 in **BZ** command is ‘1’.	Returns a value that indicates the cause of the error. See the table for **EE[14]** below.
15	Extended Homing and Touch Probe Error	**See table for EE[15] below**

Index	Description	Values
16	Advanced filter initialization error	**See table for EE[16] below**
17	Feedback emulation error	**See table for EE[17] below**
18	Reserved	
19	Kinematics error	**See table for EE[19] below**
20	Output Compare Extended error	**See table for EE[20] below**

Table 7: EE[] Indices

Errors

The following table lists errors reported by **EE[1]**

EE[1] Value	Description
1	S#[2] is out of range.
2	Velocity FIR filter illegal value.
3	Glitch filter is out of range.
4	Serial sensor clock frequency illegal value.
5	Serial sensor hardware resolution smaller than minimum 8 bits.
6	Serial sensor negative multi-turn resolution error.
7	Serial sensor negative reduced bits value.
8	Serial sensor absolute position offset is limited to 48 bits.
9	Serial sensor communication time out of range
10	Serial sensor illegal port resource.
11	Serial None-binary BISS resolution is maximum 48Bit.
12	Panasonic clock frequency should be 2.5Mhz.
13	Multi-turn bits of Panasonic Inc. sensor must be zero.
14	serial Tamagawa sensor delay init is smaller than 1500mSec
15	Tamagawa clock frequency should be 2.5Mhz.
16	Nikon serial sensor clock frequency should be 2.5Mhz or 4Mhz.
17	Yaskawa serial sensor clock frequency should be 4Mhz.
18	Serial sensor must have TS=62.5 or TS=125
29	Resolver sensor excitation offset must be positive up too senor's half cycle time
30	Resolver pole pair must be positive value
31	Hiperface serial resolution must be between 8 bits to 32 bits
32	Hiperface serial Multi-turn resolution must be between 0 bits to 16 bits

EE[1] Value	Description
33	Hiperface frequency for QUAD only threshold must be zero or positive
34	Analog sensor multiplication factor must be between 2 and 16
35	Analog sensor sine offset must be long value
36	Analog sensor cosine offset must be long value
37	Analog amplitude limits must be between 3276 to 32768
38	Analog amplitude warning limits must be between 3276 to 32768
39	Analog sine gain multiplier must be between 0.3 to 1.7
40	Analog phase shift must be between -90 degrees to +90 degrees
41	Analog amplitude limits error maximum smaller than minimum, check parameters S#[11],S#[12],S#[13],S#[14]
50	QUAD encoder use 10 bit interpolation with rising edge flag set, S#[2] bit 7
51	QUAD 1/T speed threshold must be non-negative
60	Gantry master or slave axis is outside range
61	Gantry Axis 1, position socket must be between 0 to 8
62	Gantry axis2, position socket must be between 0 to 8
63	Gantry Axis 1, velocity socket must be between 0 to 8
64	Gantry Axis 2, velocity socket must be between 0 to 8
65	Gantry slave socket index must be bigger value than master socket
70	Sine sensor command units S#[5], outside of range
71	Sine sensor frequencies outside of range, check F#[2],F#[4] to be in range of zero to 5000
72	Sine sensor DC slope F#[6], must be zero or positive
80	Motor simulation resistance F#[1] must be positive
81	Motor simulation inductance F#[2] must be positive
82	Motor simulation Back EMF constant F#[3] can't be negative
83	Motor simulation torque constant F#[4] can't be negative
84	Motor simulation mechanical inertia F#[5] can't be negative
85	Motor simulation viscous friction coefficient F#[6] can't be negative
86	Motor simulation Maximum 48 bits resolution
87	Motor simulation resolution can't be zero
90	sensor axis outside of range
95	Velocity observer sensor resolution unit F#[1],must be bigger than zero

EE[1] Value	Description
100	Analog input signal type **S#[5]** error , must be value 1 to 3
101	Analog input filter type **F#[5]** error
105	BEMF observer, Motor inductance **F#[2]**, must be positive
106	BEMF observer, Motor inductance **F#[1]**, must be positive
107	BEMF observer, KSW GAIN **F#[5]**, must be positive
108	BEMF observer, velocity factor **F#[6]**, must be positive
109	BEMF observer, LPF frequency **F#[3]**, must be positive
110	BEMF observer, velocity threshold v2 **S#[6]**, must be positive
111	BEMF observer, velocity threshold v3 **S#[7]**, must be positive
112	BEMF observer, velocity deceleration v1 **S#[9]**, must be positive
113	BEMF observer, sigmoidal function tau **F#[10]**, must be positive
114	BEMF observer, KSW LPF frequency **F#[7]**, must be positive

The following table lists errors reported by **EE[2]**

EE[2] Value	Description
0x6091	**Object 0x6091 error - position ratio is out of range**
0x607C	**Object 0x607C error - homing offset is out of position limits**
0x607B	**Object 0x607B error - minimum position range limit is bigger than maximum position range limit**
0x607D	**Object 0x607D error - error in software position limits minimum/maximum**
0x207B	**Object 0x207B error - minimum additional position range limit is bigger than maximum limit**

Table 8: EE[2] Values

The following table lists errors reported by **EE[3]**.

EE[3] Value	Description
1	Header packet missing DL
2	Header packet termination code is not **0x1234**
3	Receive header packet does not match expected packet number
4	Header packet size is incorrect
5	Header packet checksum error
6	Header packet type is not legal

EE[3] Value	Description
7	Body packet termination code is not **0x1234**
8	Received body packet number does not match expected packet number
9	Body packet size is incorrect
10	Body packet checksum error
11	The last packet was identified, but the packet number is incorrect
12	Downloaded file checksum failed
13	After download data base post process failed
14	Time-out waiting for message
15	Downloaded data is larger than user parameters, if DL type is 19
16	Writing to FLASH memory error
17	DL abort, no precise error
18	Reserved
19	Number of Bytes in packet is ODD
20	Body packet type is not legal
21	Indicates that abort packet received
22	The downloaded file size exceeds the allowed size
23	The data after decompression has a bad format
24	Bad xml file format
25	The received packet is too large
26	The running FW image key is not valid
27	Downloaded FW+TOC CRC is not valid
28	Downloaded firmware checksum is not valid
29	Read from FLASH memory error
30	Incorrect boot password
31	Incorrect firmware/Boot signature
32	Download parameters xml process has failed – reason in **EE[11]**
33	Download process has failed due to incorrect length in downloaded header
34	**Download failed due to unsuccessful SIL running trial**

EE[3] Value	Description
35	**Firmware version is incompatible to the hardware**
36	**Download type is not supported by this version**

Table 9: EE[3] Values

The following table lists the errors reported by **EE[6]**

EE[6] Value	Description
1	Ratio denominator (**EM[10]**) is out of range
2	First/last table index (**EM[5]/EM[2]**) is out of range
3	Table gap (**EM[4]/EM[7]**) is out of range
4	Velocity/acceleration FIR filter length (**EM[12]/EM[13]**) is out of range
5	Entire master + slave table length is not even
6	Master table does not start from 0
7	Master table is not monotonic rising
8	Illegal ECAM mode
9	**EM[7]** must be smaller than **EM[4]**
10	**EM[5]** must be smaller than **EM[2]**
11	Active table segment cannot be changed (et[N];N=**EM[8]**±2)
12	Master source defined wrongly

Table 10: EE[6] Values

The following table lists the errors reported by **EE[7]**.

EE[7] Value	Description
1	Read from FLASH memory error
2	Data size exceeds limits
3	Uploaded data CRC is not valid
4	Data after decompression is not valid
5	Format of **UL** command not valid
6	**UL** command offset exceeds limits
7	The signature of data in flash is not valid

EE[7] Value	Description
8	The upload of parameters XML failed

Table 11: EE[7] Values

The following table lists the errors reported by **EE[8]**.

EE[8] Value	Description
1	The index of WB array is out of range
2	The index of WB array is read only
3	The command entered is illegal
4	The command cannot be in the set of bulk commands
5	The array index of bulk command is out of range
6	The string of entered commands is incorrect
7	The referenced array (set of commands) is empty
8	The type of command is not supported
9	The get process failed

Table 12: EE[8] Values

The following table lists the errors reported by **EE[9]**

EE[9] Value	Description
1	The sub-index of object is out of range
2	Access to this object through interpreter is not allowed
3	The entered object number doesn't appear in object table list
4	The sub-index entered by the user is not valid
5	Failed Reading the value entered to set
6	Object type is not supported for download
7	The EtherCAT SDO handler returned invalid abort code
8	The size of the data to set is not valid
9	The EtherCAT SDO handler failed parsing upload request
10	The EtherCAT SDO handler failed parsing download request
11	The string uploaded from the object is not valid

EE[9] Value	Description
12	Object type is not supported for upload
13	The firmware must be configured to either CAN or EtherCAT
14	In segmented upload via CAN, not all bytes of data arrived
15	The number to set into the object is not a number
16	The number to set into the object is out of range
17	The SDO handler is in the middle of another SDO request (invalid state for '**OJ**')
18	Not enough space for CAN uploaded string
19	EtherCAT SDO handler result code : Function is called with one or more invalid arguments
20	EtherCAT SDO handler result code : Mailbox error , invalid size
21	EtherCAT SDO handler result code : Memory limit (usually cannot allocate new memory)
22	EtherCAT SDO handler result code : General error
23	EtherCAT SDO handler abort code : Client/Server command specifier not valid or unknown
24	EtherCAT SDO handler abort code : Out of memory
25	EtherCAT SDO handler abort code : Write access is not permitted in this ECAT state
26	EtherCAT SDO handler abort code : Attempt to read to write only object
27	EtherCAT SDO handler abort code : Attempt to write to read only object
28	EtherCAT SDO handler abort code : The object does not exist in the object directory
29	EtherCAT SDO handler abort code : The number and length of the objects to be mapped would exceed the PDO length
30	EtherCAT SDO handler abort code : Data type does not match, length of service parameter too low
31	EtherCAT SDO handler abort code : sub-index does not exist
32	EtherCAT SDO handler abort code : Value range of parameter exceeded (only for write access)
33	EtherCAT SDO handler abort code : General error
34	Toggle bit is not alternated
35	Command specifier is invalid

EE[9] Value	Description
36	The block size is invalid
37	The sequence number is invalid
38	Attempt to read a write only object
39	Attempt to write a read only object
40	The object does not exist
41	The object cannot be mapped
42	The length of the PDO exceeds limits
43	General parameter incompatibility
44	Hardware failed
45	Parameter too long
46	The sub-index doesn't exist
47	The parameter is out of range
48	General error
49	The data cannot be transferred
50	Wrong device state
51	No data available

Table 13: EE[9] Values

The following table lists the errors reported by **EE[11]**

EE[11] Value	Description
-1	Section 'ParameterList' is missing
-2	Section 'ParameterList' closing is missing
-3	Section 'ParameterList' format error
-4	Section 'ParameterGroup' closing is missing
-5	Section 'Parameter' is missing
-6	Section 'Name' is missing
-7	Section 'Name' closing is missing
-8	Section 'Index' is missing

EE[11] Value	Description
-9	Section 'Index' closing is missing
-10	Section 'Subindex' is missing
-11	Section 'Subindex' closing is missing
-12	Section 'Value' is missing
-13	Section 'Value' closing is missing
-14	Section 'Size' is missing
-15	Section 'Size' closing is missing
-16	XML memory exceeds the allowed size
-17	XML memory exceeds the allowed size
-18	There is an error other than XML create or parse - check **CD[2]** command

Table 14: EE[11] Values

The following table lists the errors reported by **EE[12]** & **EE[13]**

EE[12], EE[13] Value	Description
0x0000 – 0xFFFF	Internal use. Error detail while distributing firmware version

Table 15: EE[12] and EE[13] Values

The following table lists the errors reported by **EE[14]**

EE[14] Value	Description
1	Init of SIL failed at power up due to read from flash failure
2	Init of SIL failed at power up due to bad length of SIL code - exceeds sector in RAM
3	Init of SIL failed at power up due to bad signature - SIL is invalid!
4	Init of SIL failed at power up due to read from flash of SIL code failure
5	Init of SIL failed at power up due to bad CRC
6	Load of burned params failed at power up due to failure of reading header from flash
7	Load of burned params failed at power up due to bad signature
8	Load of burned params failed at power up due to bad length of data in header
9	Load of burned params failed at power up due to failure of reading data from flash

EE[14] Value	Description
10	Load of burned params failed at power up due to bad CRC
11	The save to flash of SIL burn parameters failed
12	New SIL cycle was called when previous SIL cycle wasn't finished
13	SIL not initialize OR initialization failed
14	SIL parameter file too long
15	SIL sample time must be multiple of profile sample time
16	Non-existing axis was selected in one of the SIL function blocks
17	Non-existing analog input was selected
18	SIL to FW synchronized copy buffer size was exceeded
19	SIL communication definition error

Table 16: EE[14] Values

The following table lists errors reported by **EE[15]**

EE[15] Value	Description
0	No homing error
1	CW bit 4 was reset by master
2	Capture settings were changed and DS-402 homing stopped
3	Capture settings were changed and Touch Probe stopped
4	Capture settings were changed and HM/HF homing stopped

Table 17: EE[15] Values

The following table lists the errors reported by **EE[16]**

EE[16] Value	Description
1	Non-legal filter type
2	Wrong advanced filter frequency (**KV[N-4]**)
3	Wrong advanced filter numerator frequency (**KV[N-4]**)
4	Wrong advanced filter denominator frequency (**KV[N-2]**)
5	Wrong advanced filter damping (**KV[N-3]**)
6	Wrong advanced filter numerator damping (**KV[N-3]**)

EE[16] Value	Description
7	Wrong advanced filter denominator damping (**KV[N-1]**)
8	Wrong advanced filter phase (**KV[N-3]**)
9	Wrong advanced filter quality factor (**KV[N-3]**)
10	Wrong advanced filter amplification/attenuation (**KV[N-2]**)

Table 13: EE[16] Values

The following table lists the errors reported by **EE[17]**.

EE[17] Value (Object)	Description	Details
1	Emulated socket is not defined	The socket number related to axis defined by **EA[4]** is zero or greater than 8. Check **EA[4]**, S**1[] - S8[]** commands.
2	Emulated socket changed	Emulation was stopped due to change of emulated socket with S1-S8 command. Check **EA[4]**, **S1[] - S8[]** commands.
3	Emulation FB socket is not initialized	Feedback emulation socket was required (**EA[8]**==1) and FB socket is not initialized (**EA[5]**==0). Check **EA[4]**, **EA[5]**, **S1[2] - S8[2]** bits 1-3
4	Emulation FB socket changed	Emulation was stopped due to change of emulated socket with **S1 - S8** command. Check **EA[4]**, **EA[5]**, **S1[2] - S8[2]** bits 1-3
5	In PWM mode velocity socket is not defined	In PWM mode (**EA[1]**=5 or 6), when the source signal is "velocity" (**EA[9]**=2) and emulated socket is set to 0 (**EA[4]**=0), the velocity derived from the socket is defined by **SA[2]**. If this socket is not defined (**SA[2]** <= 0 or > Max socket number) an Emulation Error is issued
6	In PWM mode current socket is not defined	In PWM mode (**EA[1]**=5 or 6), when the source signal is "current" (**EA[9]**=6) and the emulated socket is set to 0 (**EA[4]**=0), the current derived from the socket is defined by **SA[4]**. If it is not defined, an Emulation Error is issued
7	Incorrect emulated velocity limit	In PWM mode, the minimum range for the emulated velocity (**EA[12]**) is higher or equal to the max velocity range (**EA[13]**)
8	Incorrect emulated current limit	In PWM mode the minimum range for the emulated current (**EA[14]**) is higher or equal to the max current range (**EA[15]**)
9	Incorrect emulated PWM limit	The PWM emulated low value (**EA[16]**) is higher or equal to the PWM high value (**EA[17]**)

Table 14: EE[17] - Emulation Errors

The following table lists the errors reported by **EE[18]**.

EE[18] Value (Object)	Description	Details
1	Read user SIL parameters from FLASH failed	Read SIL saved parameters from FLASH memory failed during drive power-up.
2	SIL parameters length is out of range	SIL saved parameter size exceeds maximum allowed size.
3	SIL parameters bad signature - parameters are invalid	SIL parameters are not saved to FLASH memory.
4	SIL parameters bad CRC - parameters are invalid	SIL saved parameters incorrect CRC, calculated vs. saved.
5	Failed to assign parameter, check **CD[3]** command	Write saved SIL parameter failed. Specific parameter ID, Offset and size is reported in **CD[3].**

Table 18: EE[18] Values

The following table lists the errors reported by **EE[19]**.

EE[19] Value (Object)	Description	Details
1	Kinematics setting error	Some parameters have invalid value
2	Kinematics type cannot be set	Trying to set illegal kinematics type
3	Wrong number of axes in the group	Group for the specific robot must have fixed number of axes: • Scara and Three Link – 4 axes; • Delta – 3 axes; • XYZW Cartesian – 4 axes; • XYZ Cartesian – 3 axes.
5	Wrong initial robot elbow orientation	Actual Scara or Three Link robot elbow orientation does not match elbow sign parameter **K1[7]**
6	Requested motion path crosses illegal area	Applies to Scara and Three Link robots
7	Requested position target is out of robot's motion limits	Applies to Delta, Scara and Three Link robots
8	Wrong Delta robot axes positions	Initial ACS positions of all axes must be between -PI/2 and PI/2 [rad] (or -90 and 90 [deg])
9	Delta robot inverse kinematics fail	Actual robot's motion crosses illegal area

Table 19: EE[19] Values

The following table lists the errors reported by **EE[20]**.

EE[20] Value (Object)	Description	Details
1	Output Compare table mode infinite repetition error	Infinite repetition cannot be used in table mode Position + Duration (**OC[1]**=4)
2	Output Compare absolute position mode infinite repetition error	Infinite repetition can be used in modes Absolute Position + Duration (**OC[1]**=1 or 6) only if number of pulses **OC[5]** is set to 1
3	Output Compare Auto increment mode delta position error	Wrong delta position value in PAL Auto Increment Mode. The range shall be: (-32768)...32767
5	Output Compare Advanced mode parameter's error	Both **OC[18]** and **OC[38]** must be defined as advanced mode, but one of them must be a master, the other a slave
6	Output Compare repetition period error	Repetition Period cannot be less or equal of Pulse width
7	HW OC resource was overridden	Previously enabled Output Compare overridden due to using the same HW OC
8	HW AOC resource was overridden	Previously enabled Output Compare overridden due to using the same HW AOC
9	HW OC and AOC resources were overridden	Previously enabled Output Compare overridden due to using the same HW OC and AOC
10	Restart when not completed	Output Compare was restarted when previously enabled Output Compare with infinite number or a long sequence of repetition is still running
11	Output Compare SW Start standalone mode incompatibility	Output Compare SW start standalone mode cannot start due to paired OC is running on SW start synchronized mode. Disable paired OC
12	Output Compare SW Start synchronized mode incompatibility	Output Compare SW start synchronized mode cannot start due to paired OC is running on SW start standalone mode. Disable paired OC
13	Output Compare non-advanced mode incompatibility	Paired Output Compare is running in Advanced mode. Disable paired OC

Table 20: EE[20] Values

The following table lists the errors reported by **EE[21]**.

EE[21] Value (Object)	Description	Details
1	OEM Communication Error	Communication error occurred while OEM command sent.
2	OEM Memory Error	OEM memory error, the input memory region does not exist.

4	OEM Memory Index Error	OEM memory index error, the input memory index invalid.
8	Endat OEM Memory Protection	There is a memory write protection, command failed. (Only for Endat).
16	OEM Read Resolution Error	Read resolution command failed.
32	OEM Memory is Full	Sensor's memory is full.
64	OEM Memory is Empty	Sensor's memory is empty, please initialize memory blocks.
128	OEM Timeout Error	No response from sensor after max timeout.

Table 21: EE[21] Values

The following table list the errors reported by **EE[22]**.

EE[22] Value (Object)	Description	Details
0	FLOW_ERROR_NONE	No Error
1	FLOW_ERROR_ALIVE_COUNTER	Keep alive counter
2	FLOW_ERROR_CLR_PWR_COMMAND	Clear Power-up command
3	FLOW_ERROR_SET_PWR_COMMAND	Set Power-up command
4	FLOW_ERROR_SET_DRIVE_PRM_POWER_UP	Set drive parameter during power-up
5	FLOW_ERROR_SET_FG_PARAMETRES_POWER_UP	Set **FG** parameters power-up
6	FLOW_ERROR_SET_FG_PARAMETER	Update **FG** parameter
7	FLOW_ERROR_UPDATE_AXIS1	Update Axis-1 parameters
8	FLOW_ERROR_UPDATE_AXIS2	Update Axis-2 parameters
9	FLOW_ERROR_UPDATE_AXIS3	Update Axis-3 parameters
10	FLOW_ERROR_UPDATE_AXIS4	Update Axis-4 parameters
11	FLOW_ERROR_START_MODE	Start mode fail
12	FLOW_ERROR_GENERIC_PWR	PWR module error
13	FLOW_ERROR_GENERIC_MCON	MCON module error

Table 22: EE[22] Values

Error Codes

Refer to chapter **Drive Error Codes** for the error descriptions related to the error code.

References

EC, **MF**, **SR**, **DL**, **UL**, **CD[2]**

EG[N] – Acceleration/Velocity Observer

EG specifies the velocity and acceleration observer.

CANopen/CoE

Elmo Alias Object: 0x30A0

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	See below
Range	NA
Index range	1 to 9
Default	See below
Unit modes	**UM=2, 5, 6, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

Indices

The following table describes the **EG[*N*]** entries.

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Unit</th><th>Default</th><th>Range</th><th>Comments</th></tr>
<tr><td rowspan="6">1</td><td colspan="2">Observer Type</td><td rowspan="6"></td><td rowspan="6">0</td><td rowspan="6">0-4</td><td rowspan="6">The motor must be off</td></tr>
<tr><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>Elmo Single-Gain velocity and acceleration observer</td></tr>
<tr><td>2</td><td>Elmo Single-Gain velocity observer</td></tr>
<tr><td>3</td><td>Model based velocity and acceleration observer</td></tr>
<tr><td>4</td><td>Model based acceleration observer</td></tr>
<tr><td></td><td colspan="2">Observer usage mode</td><td></td><td></td><td></td><td></td></tr>
</table>

Index	Description		Unit	Default	Range	Comments
2	0	Monitoring only		0	0-3	The motor must be off
	1	Estimated acceleration is used as an acceleration feedback				
	2	Estimated velocity is used as a velocity feedback				
	3	Both estimated signals (acceleration and velocity) are used as feedbacks				
3	Elmo Single-Gain observer coefficient K			0	≥ 0	Effective for EG[1]=1 and EG[1]=2
4	Velocity observer proportional gain K_{VP}			0	≥ 0	Effective for EG[1]=3
5	Velocity observer integral gain K_{VI}			0	≥ 0	Effective for EG[1]=3
6	Velocity observer derivative gain			0	≥ 0	Effective for EG[1]=3
7	Acceleration observer proportional gain K_{AP}			0	≥ 0	Effective for EG[1]=3 and EG[1]=4
8	Acceleration observer integral gain K_{AI}			0	≥ 0	Effective for EG[1]=3 and EG[1]=4
9	Acceleration observer derivative gain K_{AD}			0	≥ 0	Effective for EG[1]=3 and EG[1]=4
10	Observer torque input gain K_{TRQ}			0	≥ 0	
11	Acceleration feedback gain K_{AFB}			0	≥ 0	
12	Observer input (P_M, V_M) source			0	0-2	
	0	Main feedback				
	1	Velocity feedback				
	2	Socket (bits 7…4 contains Socket #)				

References

EI – Initialize External Reference Generator

EI configures the external reference generator (ECAM/Follower) according to the latest **EM[N]** settings.

CANopen/CoE

Elmo Alias Object: 0x30A2

Attributes

Attribute	Description
Type	Long, Write only
Source	All
Restrictions	NA
Range	NA
Index range	NA
Default	NA
Unit modes	**UM=2, UM=5**
Non-Volatile	Yes
Axis Related	Yes

Remarks

EM[N] settings, except for EM[9], only operative after applying EI=1 command.

References

EM[N]

EM[N] – ECAM / Follower Parameters

EM[*N*] determines the behavior of the motions defined by the external reference generator (ECAM (Electronic CAM)/Follower, direct reference).

CANopen/CoE

Elmo Alias Object: 0x30A6

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 18
Default	According to array index
Unit modes	**UM=5 (Position - ECAM and Follower)** **UM=2 (Velocity – Follower)**
Non-Volatile	Yes
Axis Related	Yes

Indices

The following table details the **EM[]** entries.

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Type</th><th>Default</th><th>Restrictions</th></tr>
<tr><td rowspan="6">1</td><td colspan="2">ECAM mode</td><td></td><td></td><td></td></tr>
<tr><td>Value</td><td>Mode</td><td rowspan="5">Double</td><td rowspan="5">-1</td><td rowspan="5"></td></tr>
<tr><td>-1</td><td>Disabled</td></tr>
<tr><td>0</td><td>Follower</td></tr>
<tr><td>1</td><td>Non-periodical (linear) ECAM</td></tr>
<tr><td>2</td><td>Periodical (cyclic) ECAM</td></tr>
<tr><td>2</td><td colspan="2">Last valid index of ECAM table.</td><td>Double</td><td>10</td><td>2 to 2000</td></tr>
<tr><td>3</td><td colspan="2">Starting position
value of the input to the ECAM function, where the ECAM function output is ET[EM[5]]</td><td>Double</td><td>0</td><td>-2^{52} to $(2^{52} - 1)$</td></tr>
<tr><td>4</td><td colspan="2">Master reference (ΔPY) distance (gap) between consecutive points in the ECAM table ET[N].</td><td>Double</td><td>1</td><td>1 to $(2^{52} - 1)$</td></tr>
</table>

<table>
<tr><th>Index</th><th>Description</th><th>Type</th><th>Default</th><th>Restrictions</th></tr>
<tr><td></td><td>Effective when the constant master gap table is selected (EM[11]:bit2 = 0).</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>First valid index of the ECAM table.</td><td>Double</td><td>1</td><td>1 to 1999</td></tr>
<tr><td>6</td><td>Reserved</td><td>Double</td><td>0</td><td></td></tr>
<tr><td>7</td><td>Last master segment shortening.
Last master distance (gap)
ΔPY=EM[4]-EM[7].
Effective when constant master gap table is selected (EM[11]:bit2 = 0)</td><td>Double</td><td>0</td><td>0 to (2^{52} – 1)
EM[7]<EM[4]</td></tr>
<tr><td>8</td><td>Reports the present index in the ECAM table.
When the ECAM motion is not active, EM[8] reports 0.</td><td>Double</td><td>0</td><td>1 to 2000
Read-only</td></tr>
<tr><td>9</td><td>ECAM ratio numerator.
Set EM[9] to synchronously activate the new ECAM/Follower ratio = EM[9]/EM[10].</td><td>Double</td><td>1</td><td></td></tr>
<tr><td>10</td><td>ECAM ratio denominator.
The EM[10] setting becomes effect only after setting EM[9].</td><td>Double</td><td>1</td><td>>0</td></tr>
<tr><td>11</td><td>ECAM options
<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td colspan="2">Reserved</td></tr>
<tr><td>1</td><td colspan="2">Reserved</td></tr>
<tr><td rowspan="3">2</td><td colspan="2">ECAM table type</td></tr>
<tr><td>0</td><td>One table, fixed (constant) master gap: ET[n],n=1…2048</td></tr>
<tr><td>1</td><td>Two separate master and slave tables, with variable gap:
Master: ET[n]>=0, n=1…N, ET[EM[5]]=0;
Slave: ET[m], m=N+1…2*N;</td></tr>
<tr><td rowspan="3">3</td><td colspan="2">ECAM master source type</td></tr>
<tr><td>0</td><td>External reference encoder (defined in socket architecture)</td></tr>
<tr><td>1</td><td>Main Profiler of another axis</td></tr>
<tr><td rowspan="2">4</td><td colspan="2">Interpolation type</td></tr>
<tr><td>0</td><td>Quadratic interpolation</td></tr>
</table>
</td><td>(long)
Double</td><td>0</td><td></td></tr>
</table>

Index	Description	Type	Default	Restrictions
	1 – Linear interpolation 5 – Table index search mode 0 – Incremental 1 – Direct			
12	ECAM velocity FIR filter length	Double	1	1 to 8
13	ECAM acceleration FIR filter length	Double	1	1 to 8
14	Direct reference ratio numerator. **Set EM[14] to synchronously activate the new direct reference ratio = EM[14]/EM[15].**	Double	1	
15	Direct reference ratio denominator. **The EM[15] setting becomes effective only after setting EM[14].**	Double	1	>0
16	ECAM master source **0 – Disabled** 1…8 – **Socket number, if EM[11]:3=0** **1…WS[1] – Axis number, if EM[11]:3=1**	Double	0	**0 to 8, if EM[11]:3=0** **0 to WS[1], if EM[11]:3=1**
17	Master modulo low limit. **The EM[17] setting becomes effective only after setting EM[18].**	Double	0	-2^{52} to $(2^{52} - 1)$
18	Master modulo high limit **Set EM[18] to synchronously activate the new master modulo range EM[17]/EM[18]**	Double	0	-2^{52} to $(2^{52} - 1)$

Notes

- Modifying of ECAM input source type (**EM[11]:**bit3) is only possible, if motor is off (**MO**=0)
- Parameters **EM[1]**-**EM[5]**, **EM[7]**, **EM[10]**-**EM[11]** are synchronously activated when **EI=1** command is applied. **EM[9]**, **EM[10]** are also activated when **EM[9]** is set.
- Parameters **EM[12]**, **EM[13]** are activated at **RM**=1 command
- If the variable master gap table is selected (**EM[11]**:bit2=1):
  - Master table must start from **0** and be monotonic rising
  - Shared length of master and slave tables is (**EM[2]**-**EM[5]**+1), which must be an even number
  - **EM[8]** reports the slave index. Master index can be calculated by subtracting constant offset ((**EM[2]**-**EM[5]**+1)/2) from the slave index
- Use **EM[7]** to build a fixed master gap table with length which is not an integer multiple of **EM[4]**.

- When the motor is enabled (**MO**=1) and ECAM table is running (**EM[1]**=1 or **EM[1]**=2), ECAM table entries **ET[N]** can be changed "on the fly", with the exception of the **ET[EM[8]**-**2]**…**ET[EM[8]+2]** entries
- **Incremental calculation of the ECAM table index (EM[11]:bit5=0)**
  In this case, the index is directly calculated only once, at the ECAM engage and then incremented or decremented when the master crosses the border between two adjacent ECAM table segments. When using this mode, the difference between two consecutive master samples must not exceed the master table segment length (table gap).
- **Direct search/calculation of the ECAM table index (EM[11]:bit5=1)**
  In this case, the index is calculated every ECAM execution period; this consumes extra DSP time.
- The input to acceleration FIR filter is the output of velocity FIR filter.
- Master modulo **EM[17]**, **EM[18]**:
  - The master modulo position is counted cyclically, and therefore after the position is counted to its maximum value, the next position count will reset the position counter back to its minimum value. The speed reading is not affected by the position jump. For example: If **EM[17]**=-5 and **EM[18]**=5, the master position is counted in a cycle length of 10. The master position will always be in the
    range [-5…4]. If the master position rotates in the positive direction, the master position count will proceed from 0, 1, 2, 3, 4 to -5, -4, -3, -2, -1, 0, 1….. etc.
  - A new **EM[17]**/ **EM[18]** setting is activated after the setting of **EM[18]**.
  - **EM[18]** must be larger than **EM[17]**, i.e. **EM[18]**> **EM[17]**.
  - If **EM[17]** = **EM[18]** = 0, the master position modulo functionality is disabled.
  - If the **EM[18] - EM[17]** delta value is selected low and the master position speed is too high, more than one full revolution of the counter may elapse within a single sampling time. This will cause the master position to behave unpredictably.

References

ET[], **RM**, **EI**

EO – Echo Off

EO specifies the communication echo mode.

CANopen/CoE

Elmo Alias Object: 0x30A8

Attributes

Attribute	Description
Type	short, Read/Write
Source	All
Restrictions	None
Range	0, 1
Default	0
Index Range	1
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

When serial communication is used, the command is prompted back to the host. **EO** can turn this off. Depending on the communication protocol (RS232 or USB), the echoing is performed on character level or on the command level.

EO = 1 Enable echo.

EO = 0 Disable echo.

EO can be set from other communication lines, but it affects only USB and RS232 communication.

References

ER[] – Maximum Tracking Error

ER[] specifies the maximum tracking error (AKA following error) of the relevant control loop.

CANopen/CoE

Elmo Alias Object: 0x30AB

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	Refer to the note below.
Range	See the table below.
Index range	2, 3, 5
Default	According to array index
Unit modes	**ER[2] for the velocity loop under UM = 2, 5, 7** **ER[3], ER[6] for the position loop under UM = 5, 7** **ER[5], ER[7] for position loop in gantry setting under UM=5, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

The tracking error is the difference between the command (desired value) and its feedback. Tracking errors include velocity and position errors. If the error exceeds this value, the motor is automatically disabled. **MF** indicates the reason for the failure.

Indices

The following table describes the **ER[]** entries.

Index	Description	Type	Values	Default
1	Reserved			
2	The maximum allowed velocity error **Its value corresponds to abs(DV[2]-VX))**		0 to 10^{11} [UU/sec]	100000000 [UU/sec]
3	**The maximum allowed position error in user units (0x6065)** **Its value corresponds to abs(DV[3]-PX)**	double	0 to $(2^{52} – 1)$ [UU]	1000000000 [UU]

Index	Description	Type	Values	Default
5	The maximum allowed Yaw position. Yaw position is the difference between master and slave position.	double	0 to (2^{52}– 1) [UU]	1000000000 [UU]
6	**Position following error time out, in milliseconds (0x6066)**	double	0 to 65535	0
7	Velocity error time out, in milliseconds	double	0 to 65535	0

References

MF, **MO**, **SR**, **DV[]**, **VX**, **PX**

ET[N] – ECAM/Error Correction Table

ET[*N*] specifies the double array which can be used as general-purpose non-volatile memory. It can be used by the following drive algorithms: ECAM, error mapping, and output compare.

CANopen/CoE

Elmo Alias Object: 0x30AD

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restriction	None
Range	-2^{52} to $(2^{52} - 1)$
Index range	1 to 2000
Default	None
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

- The user must make sure that there is no overlapping between ECAM and error correction portions of the ET array.
- When the motor is enabled (**MO=1**) and the ECAM table is controlling the motion ((**EM[1]=1** or **EM[1]=2**) and **RM=1**), the ECAM table entries can be changed "on the fly", with the exception of the **ET[EM[8]-2]…ET[EM[8]+2]** entries. If during these entries, a change is made "on-the-fly", an error is produced.
- Using the alias object is relevant only for the first 254 sub-indices. The alias of the **E#[]** allows addressing the whole range of ET indices via EtherCAT \ CANopen.

References

E#[N], EM[*N*], **PC[N]**

EV – Vector Enable

EV defines type of the group - scalar group or vector group.

CANopen/CoE

Elmo Alias Object: 0x30AF

Attributes

Attribute	Description
Type	Unsigned short, Read/Write
Source	All
Restrictions	NA
Range	0, 1, 2
Index range	NA
Default	0
Unit modes	UM = 5
Non-Volatile	No
Axis Related	No

Remarks

- Prefix **GR#.** is used to address the specific group, for example, **GR2.EV**
- **GR#.EV**=0 disables vector mode of group # and defines it as a scalar group. In this mode all Elmo commands following the group prefix **GR#** are duplicated for all group's axes and are sent to them according to the axes order (see **AY** command). Motions of the different group's axes are not synchronized.
- **GR#.EV**=1/2 enables group # in vector mode. In this modes all Elmo motion commands following the group prefix **GR#** are interpreted as vector commands. Motions of the different group's axes are synchronized, they are started and finished simultaneously.
- **GR#.EV**=1 enables group in Axes Coordinate System (ACS) vector mode: the system of coordinates related to the physical motors and the single movements caused by the single drives.
- **GR#.EV**=2 enables group in Machine Coordinate System (MCS) vector mode: the Cartesian system of coordinates that related to the machine. The coordinate system from the physical multiple axis ACS is linked to the MCS via a kinematic transformation (see **K1, K2** commands)

Indices

The following table describes the available options for **EV**.

Index	Description	Values		Restrictions
1	Group vector mode activation	0	Disable group vector mode	
		1	Enable group vector in ACS mode	

Index	Description	Values		Restrictions
		2	Enable group vector in MCS mode	

References

AY[], **K1[]**, **K2[]**, **GR**

FC[] – Scaling Factors

FC[] defines user-defined units and specifies the values of the sensor resolution and transmission ratio for position, velocity and acceleration.

CANopen/CoE

Position_encoder_resolution: 0x608F	Velocity_encoder_resolution: 0x6090 (reserved)
Gear_Ratio: 0x6091	Feed_Constant: 0x6092
Velocity_factor: 0x6096	Acceleration_factor: 0x6097
Elmo Alias Object: 0x30C0	

Attributes

Attribute	Description
Type	Unsigned Long, Read/Write
Source	All
Restrictions	None
Range	1 to (2^{32} – 1)
Index range	1 to 12
Default	1 for all settings
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

The gear ratio together with the feed constant determines the relation between the position in user units and the actual movement in counts.

$$\frac{UserUnits}{EncoderCounts} = \frac{FeedConstant}{GearRatio * PositionEncoderResolution}$$

Here the *feed constant* is the positional movement for any motor movement and is calculated using the formula

$$FeedConstant = \frac{Feed}{DrivingShaftRevolutions} = \frac{FC[7]}{FC[8]} = \frac{0x6092.1}{0x6092.2}$$

The *gear ratio* is the ratio that defines what a gear adds to the movement and is calculated using the formula

$$GearRatio = \frac{MotorShaftRevolutions}{DrivingShaftRevolutions} = \frac{FC[5]}{FC[6]} = \frac{0x6091.1}{0x6091.2}$$

The *position encoder resolution* is the ratio between the motor shaft and the encoder counts:

$$PositionEncoderResolution = \frac{EncoderCounts}{MotorShaftRevolution} = \frac{FC[1]}{FC[2]} = \frac{0x608F.1}{0x608F.2}$$

The drive uses these variables to convert the position in user units into internal units (counts) for all position references (e.g., 0x607A) and for position feedback (e.g., 0x6064).

The *velocity factor* can be used to match the velocity units to the user-defined velocity units. The *user-defined velocity unit* is the *user-defined position unit* / sec:

$$VelocityValue = \frac{PositionValue}{sec} * VelocityFactor$$

$$VelocityFactor = \frac{FC[9]}{FC[10]} = \frac{0x6096.1}{0x6096.2}$$

The *acceleration factor* can be used to match the acceleration units to the user-defined acceleration units. The *user-defined acceleration unit* is the *user-defined velocity unit* / sec:

$$AccelerationValue = \frac{VelocityValue}{sec} * AccelerationFactor$$

$$AccelerationFactor = \frac{FC[11]}{FC[12]} = \frac{0x6097.1}{0x6097.2}$$

Indices

The following table describes the **FC[]** entries.

Index	Description	Type	Note
0	Reserved		
1	Position encoder resolution numerator	Unsigned Long	0x608F.1
2	Position encoder resolution denominator	Unsigned Long	0x608F.2
3	Velocity encoder resolution numerator	Unsigned Long	0x6090.1 (reserved)
4	Velocity encoder resolution denominator	Unsigned Long	0x6090.2 (reserved)
5	Gear ratio numerator	Unsigned Long	0x6091.1
6	Gear ratio denominator	Unsigned Long	0x6091.2
7	Feed constant numerator	Unsigned Long	0x6092.1
8	Feed constant denominator	Unsigned Long	0x6092.2
9	Velocity factor numerator	Unsigned Long	0x6096.1
10	Velocity factor denominator	Unsigned Long	0x6096.2

Index	Description	Type	Note
11	Acceleration factor numerator	Unsigned Long	0x6097.1
12	Acceleration factor denominator	Unsigned Long	0x6097.2

References

AC, DC, SP, VH[2], FS, SD

FF[] – Feed Forward

FF[] specifies the feed forward configuration and is used to improve control performance.

CANopen/CoE

Elmo Alias Object: 0x30C3

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	See the table below.
Index Range	1 to 5, 7
Default	See the table below.
Unit modes	See the table below.
Non-Volatile	Yes
Axis Related	Yes

Remarks

Feed forward is available for position, velocity, and current in open loop stepper mode.

The EAS application locates and uses **FF[1]** automatically during the Tuning Process. To fine tune the drive, reset this value and record a motion. **FF[1]** is equal to the current command (during acceleration) divided by the profile acceleration. It is better to take half of this value.

The actual value of the velocity feed forward is the derivative of the position command multiplied by **FF[2]**. Elmo uses the PIP cascade control structure and usually this value should be equal 1.

FF[3] is the ratio between resolution of the velocity sensor and the resolution of the position sensor, in dual loop. When only one sensor is used, this value should be equal to 1.

FF[4] is used for phase advance in brushless motors. It extrapolates the phase according to the velocity.

FF[5] is used for additional current proportional to velocity in open loop stepper mode (**UM**=6).

For open and closed loop stepper mode, **FF[1]** and **FF[5]** operate during the profiler until completion.

Indices

The following table details the **FF[]** entries:

Index	Description	Default	Values	Restrictions
1	Specifies how much of the second derivative of the position reference (or the first derivative of the velocity	0	0 to 2000	**UM=2, 5, 6, 7**

Index	Description	Default	Values	Restrictions
	reference) is fed as a reference to the current controller. For open and closed loop stepper mode **(UM=6, 7)** specifies the factor of additional gain proportional to acceleration. In amperes per acceleration (counts per second^2).			
2	Specifies the factor of velocity feed forward added to the position controller output in the velocity command.	1	0 to 1	**UM=2, 5, 7**
3	Specifies the ratio between the velocity sensor resolution and the position sensor resolution in dual loop.	1	>=0	**UM=2, 5, 7**
4	Specifies the phase advance in brushless motors. When this value is set to 1, the phase advance is an extrapolation to half TS (where the voltage command is output) according to the velocity.	0	-30 to 30	When **FF[4]<0,** phase advance occurs only on voltage output. When **FF[4]>0,** phase advanced occurs on current reading and also on voltage output.
5	For open and closed loop stepper mode (**UM**=6, 7) specifies additional current proportional to speed.	0	0 to 200	**UM**=6, 7
6	Reserved			
7	Additional field angle advance for voltage calculation only	0	0 to 20	
11	Specifies the factor of the Gantry bridge upper axis (Y) acceleration feed forward added to the yaw controller output. The feedforward depends on position defined via **YG[]** command. In amperes per acceleration (counts per second^2).	0		**UM**=2, 5
12	Specifies the factor of the cross-axis acceleration feed forward added to the	0		**UM**=2, 5

Index	Description	Default	Values	Restrictions
	velocity controller output in the current command. In amperes per acceleration (counts per second^2).			

Examples

Example 1

Suppose that there is a gear motor with a reduction ratio of 5 drives per load. The motor has an encoder with 1000 lines. The motor speed is used for the inner feedback loop. The load position measured by an encoder with 2000 lines is used as feedback for the outer loop. To prevent a steady-state error at constant speed, set the following: **FF[2]** = 1; $\mathrm{FF}[3] = \frac{1000 * 5}{2000} = 2.5$.

Example 2

Suppose that you want to add feed forward of acceleration into the current controller. You know that one ampere will cause an acceleration of 1,000,000 counts/sec^2 . Then you need to set the following:

$$\mathrm{FF}[1] = \frac{1}{1{,}000{,}000} = 1.0e-6$$

Example 3

To obtain current command proportional to velocity reference in stepper open and closed loop mode (**UM**=6, 7), reset **HT[2]** and **HT[3]** and only set **FF[5]**.

References

UM, **CA[], XA[]**, YG[]

FP[] – Feedback Position

FP*[N]* specifies the position of the feedback which is associated with socket *N*.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	The motor must be off.
Range	None
Index Range	1 to 8
Default	NA
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

Each feedback can be mapped to a socket which can then be referenced using the socket index. FP[*N*] is the position of the feedback which is mapped to socket *N*.

The socket always returns a value in physical units (not user units).

On power-up the socket is reset to its feedback value. In case of absolute feedback the feedback is read, and the socket gets the relevant value. In all other cases the position is set to 0.

PX returns the value (in user units) of the position socket as defined in SA[1].

When working with a gantry, homing should set all three sockets: the slave and master by the FP[*N*] command and the main feedback for position (average of the two) by the PX command.

References

SA[], PX, FV[]

FS – PTP Final Speed

FS specifies the configured velocity, which the drive will have on reaching the target position.

CANopen/CoE

Target_velocity: 0x6082Elmo Alias Object: 0x30D0

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	**Effective on the next call to BG**
Range	$0 \ldots 10^{11}$
Index range	1
Default	0
Unit modes	**UM = 2, 5, 6, 7**
Non-Volatile	No
Axis Related	Yes

Remarks

The **FS** command defines the final speed (or end speed), at which the drive will continue to jog after reaching the position target.

FS is part of the profiler command and is calculated on the **BG** command or, when DS 402 is used, on the rising edge of bit 4 in the control word when the motion mode is the Profile Position mode.

Note:
In cases in which **FS** is not 0, after reaching the position, the drive will jog at the **FS** value. Setting **FS** to 0 and performing another **BG** will cause the profiler to reevaluate the profile and cause a movement to the last **PA** command.

FS is important for blend movement as defined by the PLCopen document.

Setting **FS** to 0 and then calling **BG** does not guarantee that the motion will stop. On the contrary, if the target position was not set correctly, the motor will spin backwards.

FS overrides object 0x6082 on BG.

The **FS** value can be given in user-defined units specified by the **FC** command.

The **FS** command must be resent after using **JP** command before the next PTP motion, if non-zero final speed is required.

References

PA, **BG**

FT – Floating Point Trigger Level

FT[] specifies the floating-point trigger for the recorder.

CANopen/CoE

Elmo Alias Object: 0x30D1

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	**Recorder inactive (RR = 0 or RR = -1)**
Range	As in the table below
Index range	1 to 4
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The **FT[]** command allows capturing of floating point triggers for the drive recorder.

The value is typically used by the EAS recorder.

Indices

The following table describes the **FT[]** entries.

Index	Description	Type	Values	Restrictions
0	Reserved			
1	Set for positive and negative slope – First Trigger level 1	Double	Double range	
2	Set for window – First Trigger level 2.	Double	Double Range	
3	Set for positive and negative slope – Second Trigger level 1	Double	Double Range	
4	Set for window – Second Trigger level 2.	Double	Double Range	

References

FV[] – Feedback Velocity

FV[*N*] reads the velocity of the feedback which is associated with socket *N*.

CANopen/CoE

Elmo Alias Object: **0x30D3**

Attributes

Attribute	Description
Type	Float, Read-only
Source	All
Restrictions	For non-active sockets return zero
Range	None
Index Range	1 to 8
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

Each feedback can be mapped to a socket which can then be referenced using the socket index. FV[*N*] is the velocity of the feedback which is mapped to socket *N*.

The socket always returns a value in physical units [Counts/sec] (not user units).

References

CA[], **VX**, **FP[]**

G1[]….G5[] – Capture Array Value from HM/HF

G1[] - G5[] retrieves captured values from the capture array. The command is none-axis related.

CANopen/CoE

Alias Objects:

G1[] - 0x30D9,

G2[] - 0x30DA,

G3[] - 0x30DB,

G4[] - 0x30DC,

G5[] - 0x30DD

Attributes

Attribute	Description
Type	Double, Read/Write
Source	RS232, USB, TCP, EoE
Restrictions	**User is responsible to avoid using the same array entries for several axes or several homing (for example, HM and HF homing runs in parallel and use full array size 1-1024) .**
Range	None
Index range	**G1[] - 1…254 ; Capture array 1…254** **G2[] - 1…254 ; Capture array 255…508** **G3[] - 1…254 ; Capture array 509…762** **G4[] - 1…254 ; Capture array 763…1016** **G5[] – 1…8 ; Capture array 1017…1024**
Default	None
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

During position capture, the set of captured positions can be defined by **HM[12], HM[13]**. In that case the **G1[]…G5[]** commands can be used to retrieve the position inputs.

This command reads the captured array, saving the user the need to know how the software stores the captured values in the array.

Note that the available range of array indexes to be set by HM[12] (Start index), HM[13] (End index) is 1...1024. The available range of HM[1] is 1...32000 (32000 is called "Infinite").

- If **HM[1]**="Infinite" - then captured positions are written into array continuously, with rollover from End index to Start index
- If HM[1] is set as: (End index - Start index + 1) < HM[1] < Infinite
- - then HM[1] captured positions are written into array with rollover from End index to Start index
- If **HM[1]** is set as: ***HM[1] <= (End index - Start index + 1)*** - then HM[1] captured positions are written into array from indexes ***Start index*** to ***End index*** . In that case rollover is not needed
- Reading of array index greater than 1024 returns error 3 "Bad index"
- Reading of array less than 1024 but out of range ***Start index … End index*** returns junk value. Error code is not released in this case
- The next index to be filled is indicated in **HM[9]**
- When homing was activated by settings **HM[1]** ="Value", **HM[1]** decrements every homing event from "Value" to 0, regardless of settings **HM[12]**, **HM[13]**

References

HM[]

GA[N] – Port A, B, C Titanium Mux Selection (Titanium Drive Only)

Routes a dedicated function to the desired Port A, B or C in the Titanium drive.

CANopen/CoE

Alias objects: 0x30E2

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	All
Restrictions	None
Range	See value definitions
Index range	1 to 6
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

Note:
Refer to the Administrative Guide for the actual abilities and restrictions of the outputs.

GA[] Indices and Values

The following table describes the values of the **GA1]…GA[6]** commands in the Titanium drive.

Sub-index 1 …6	Port A, B, C	Bits 15 … 12 value	Bits 11 … 8 value Channel Index	Bits 7 … 4 value Channel B	Bits 3…0 value Channel A
1	X1 Port A	Reserved, see remarks	See following table Column “Channel Index” (Dout11)	See following table Column “Channel B” (Dout10)	See following table Column “Channel A” (Dout9)
2	X1 Port B	Reserved, see remarks	See following table Column “Channel Index” (Dout12)	See Note.	See Note.
3	X1 Port C	Reserved, see Remarks			

Sub-index 1 …6	Port A, B, C	Bits 15 … 12 value	Bits 11 … 8 value Channel Index	Bits 7 … 4 value Channel B	Bits 3…0 value Channel A
4	X2 Port A	Reserved, see remarks	See following table Column “Channel Index” (Dout15)	See following table Column “Channel B” (Dout14)	See following table Column “Channel A” (Dout13)
5	X2 Port B	Reserved, see remarks	See following table Column “Channel Index” (Dout16)	See Note	See Note
6	Reserved for X2 Port C	Reserved, see Remarks			

Note: X1, X2 Port B Channels A and B are input only. Value 0 only is permitted. Rest values have no effect

Value Definition

	Channel Index	Channel B	Channel A
0	Input	Input	Input
1	Output	Output	Output
2	Signal output Index (simulation)	Signal output B (simulation)	Signal output A (simulation)
3	ABS2_Clock /ABS2_Data	ABS1_Data	ABS1_Clock
4	Acuro scsm Ack (simulation)	Port B_B (loop back)	PortB_A (loop back)
5	AUX1_UART_TX	AUX1_UART_TX	AUX1_UART_TX
6	AUX2_UART_TX	AUX2_UART_TX	AUX2_UART_TX
7	EMU_A	EMU_A	EMU_A
8	EMU_B	EMU_B	EMU_B
9	AOC	AOC	AOC
10	Serial TX1 or PWM_SYNC	Serial TX1 or PWM_SYNC	Serial TX1 or PWM_SYNC

Remarks

GA[0] – read access returns the number of sub-indexes, write access returns error

GA[3], GA[6] – are reserved. Read access returns 0, write access has no effect

Reserved bits 12-15 on GA[1], GA[2], GA[4], GA[5]: write access has no effect, read access returns 0

Not reserved bits on GA[1], GA[2], GA[4], GA[5]: write access with incorrect values (see “Value definition” table) returns error “Out of range”

References

GV[], GW[], OC[], GC[], OL[], OB[], OP

GB[N] –Port A Port B Quartet MUX Selection

GB[N] routes Port A1 Index - Port A4 Index, Port B1 Index - Port B4 Index into the IP command read value of the **Platinum Quartet drive**. The command is relevant for the Quartet family according to the **firmware version** described in the table below.

CANopen/CoE

Elmo parameters objects: 0x30E3

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	CANopen, CoE, RS232, USB, TCP, EoE
Restrictions	No
Range	0…1
Index range	1
Default	0
Unit modes	All
Non-Volatile	Yes
Attribute	None
Axis Related	No

Remarks

The following tables detail the **GB[]** command options.

Bits	Name	Description
0	Mux selection	0- Default routing all inputs signals are based on the Quartet I/O port as described in the table below 1- Enable the routing of the index signals of ports A & B as described in the table below (from **version 2.0.3.6 B01**)
1 - 7	Reserved	Write access has no effect, read access returns 0

Note

The digital inputs are reflected in the extended inputs via **XI[1]** command (map-able object **0x2202.0**)

The following table describes the Inputs routing with respect to the mux between the embedded\extended inputs and the sensor PORTS A\B

Note that where no value is shown for **XI[1]**, the values are reserved for future use:

Axis	GB[1]=0 Digital Input number	GB[1]=1 PortA\B +Digital input	Related IL[] command	Extended bits XI[1]	IP Command bit (0x60FD)
Quartet Axis 1	IN1	PortA1 Index	Ax1.IL[1]	0	16
	IN2	PortB1 Index	Ax1.IL[2]	1	17
	IN3	PortA2 Index	Ax1.IL[3]	2	18
	IN4	IN4	Ax1.IL[4]	-	19
Quartet Axis 2	IN1	PortB2 Index	Ax2.IL[5]	4	20
	IN2	PortA3 Index	Ax2.IL[6]	5	21
	IN3	PortB3 Index	Ax2.IL[7]	6	22
	IN4	IN4	Ax2.IL[8]	-	23
Quartet Axis 3	IN1	PortA4 Index	Ax3.IL[9]	8	24
	IN2	PortB4 Index	Ax3.IL[10]	9	25
	IN3	IN3	Ax3.IL[11]	10	26
	IN4	IN4	Ax3.IL[12]	-	27
Quartet Axis 4	IN1	IN1	Ax4.IL[13]	12	28
	IN2	IN2	Ax4.IL[14]	13	29
	IN3	IN3	Ax4.IL[15]	14	30
	IN4	IN4	Ax4.IL[16]	-	31

References

HM[], **HF[]**, **IP[]**, **XI[]**, DS-402 Homing Mode, Touch-Probe

GC[N] – Configure Port C Mux

GC[N] specifies functionality of the Port C pins A, B, and Index (I).

CANopen/CoE

Elmo Alias Object: 0x30E4

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	All
Restrictions	-
Range	0 to 15
Index Range	1 to 3
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Indices

The following table describes the possible **GC[N]** index values and the functionalities associated with them.

Index	Pin	Value	DIR	Signal	Description
1	Port C_A	0	IN	IN	
		1	OUT	F.F	The output is synchronized to the real time interrupt
		2	OUT	Reserved	
		3	OUT	Reserved	
		4	OUT	RS_232_TX	
		5	OUT	EMU_A	QUAD_A/ Pulse/UP/Hall_A according to the Emulation mode
		6	OUT	AOC1	
		7	OUT	AOC2	
		8	OUT	Serial TX1	Serial communication channel 1
		9	OUT	Port A_A	
		10	OUT	Port B_A	
		11	OUT	Monitor ABS1_Clock	
		12	OUT	Reserved	

Index	Pin	Value	DIR	Signal	Description
2	Port C_B	0	IN	IN	
		1	OUT	F.F	The output is synchronized to the real time interrupt
		2	OUT	Serial TX1	Serial communication channel 1
		3	OUT	Reserved	
		4	OUT	RS_232_TX	
		5	OUT	EMU_B	QUAD_B/ DIR/DOWN/Hall_B according to the Emulation mode
		6	OUT	AOC1	
		7	OUT	AOC2	
		8	OUT	Serial TX2	Serial communication channel 2
		9	OUT	Port A_B	
		10	OUT	Port B_B	
		11	OUT	Monitor ABS1_Data	
		12	OUT	Reserved	
3	Port C_I	0	IN	IN	
		1	OUT	F.F	The output is synchronized to the real time interrupt.
		2	OUT	Serial TX1	Serial communication channel 1
		3	OUT	Reserved	
		4	OUT	RS_232_TX	
		5	OUT	EMU_C	Hall_C according to the Emulation mode
		6	OUT	AOC1	
		7	OUT	AOC2	
		8	OUT	Serial TX3	Serial communication channel 3
		9	OUT	Port A_I	
		10	OUT	Port B_I	
		11	OUT	Monitor ABS1_Frame	
		12	OUT	QUAD_OC1	

References

GD[N] – Quartet Port C Mux Selection

GD[N] specifies functionality of the Port C pins A, B in the **Platinum Quartet drive**.

CANopen/CoE

Elmo Alias Object: 0x30E5

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	All
Restrictions	-
Range	0 to 5
Index Range	1 to 8
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Indices

The following table describes the possible **GD[N]** indexes signal/pin associated with them.

Index	Signal/Pin
1	Port 1C Channel A Tx
2	Port 1C Channel B Tx
3	Port 2C Channel A Tx
4	Port 2C Channel B Tx
5	Port 1C Channel B Rx (reserved)
6	Port 1C Channel B direction (reserved)
7	Port 2C Channel B Rx (reserved)
8	Port 2C Channel B Direction (reserved)

Indices 5 – 8 are reserved, write access has no effect, read access returns 0

The Quartet drive supports four user programmable Advanced Output Compare modules AOC1 – AOC4 and one Emulation module.

The following table describes the values possible to write to and read from indices 1 - 4 and the drive HW associated with it.

GD[1]-GD[4] Values	HW selected to signal/pin defined by Index 1 - 4
0	When write has no effect
1	AOC1 output
2	AOC2 output
3	AOC3 output
4	AOC4 output
5	Emulation output

References

OC[], **EA[]**

GG – Configure Gantry Communication

GG configures the serial gantry communication.

CANopen/CoE

Elmo Alias Object: 0x30E8

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	All
Restrictions	-
Range	0 to 0x1FF
Index Range	1
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The following table details the **GG** options.

<table>
<tr><th>Bits</th><th>Name</th><th colspan="2">Description</th></tr>
<tr><td>0</td><td>Serial Communication Enable</td><td colspan="2">0-Disable
1-Enable</td></tr>
<tr><td>1</td><td>Master/Slave</td><td colspan="2">0-Slave
1-Master</td></tr>
<tr><td>3-2</td><td>Transfer length</td><td colspan="2">0- 8 words (16bits x 8)
1-16 words
2-32 words
3-4 words</td></tr>
<tr><td>4</td><td>Not used</td><td colspan="2"></td></tr>
<tr><td rowspan="3">7-5</td><td rowspan="3">Mux Serial RX</td><td>Channel</td><td>Connected to</td></tr>
<tr><td colspan="2">0 – three channels of serial (master)</td></tr>
<tr><td>Serial1</td><td>LVDS_Clock</td></tr>
</table>

Bits	Name	Description	
		Serial2	LVDS_TX
		Serial3	LVDS_RX
		1 - three channels of serial (master)	
		Serial1	Port_A_Index
		Serial2	Port_B_Index
		Serial3	IN5
		2 - one channel of serial	
		serial_rx_in1	PORTC_INDEX_RX
		3 - one channel of serial	
		serial_rx_in1	PORTC_B_RX
		4 - one channel of serial	
		serial_rx_in1	PORTC_A_RX
		5 - three channels of serial (master)	
		Serial1	Port_B_A
		Serial2	Port_B_B
		Serial3	Port_B_I
		6 - one channel of serial	
		serial_rx_in1	PORTA_Index
		7 - one channel of serial	
		serial_rx_in1	PORTB_Index
8	Communication clock frequency	0 – 5Mhz 1 – 10Mhz	

References

GI[N] – Capture Input MUX Selection

GI[N] routes Port A Index, Port B Index , Port C Index or digital inputs into Capture Module 1 – Capture Module 8 input signal multiplexor. The command is device related.

CANopen/CoE

Elmo parameters objects: 0x30EA

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	CANopen, CoE, RS232, USB, TCP, EoE
Restrictions	None
Range	0…11
Index range	1 to 8
Default	Refer to the **GI[*N*]** command format table below.
Unit modes	All
Non-Volatile	Yes
Attribute	None
Axis Related	No

Remarks

The Platinum drive includes 8 quadrature encoder modules Quad 1 - Quad 8 and 8 Capture modules. These modules are used, among other things, to capture the position of an input signal, and are mainly used to count the position and calculate the speed of AQB sensors.

The **GI[*N*]** command enables routing of Port A, B, C Indexes or general purpose digital inputs into Capture Module 1 to Capture Module 8 inputs of the drive's quad modules. This allows any position capturing from various types of inputs.

Note:
Different drives may have different number of available digital inputs and quadrature encoder modules. Consult the specific drive's Installation Guide for details. The drive should not prevent any setting of a non-existing input.

The command includes the input to be routed and the Capture module to which it is routed.

GI[] values are saved to non-volatile memory. At power-up the routing of the signals is evaluated according to **GI[]** value.

Indices

The following table details the **GI[]** options for the **Platinum** drive :

Index	Description	Default	Value/Connected Input
1 - 8	Capture Modules 1 – 8 routing (see Notes after the table)	0	0 – Port A Index 1 – Port B Index 2 – Port C Index 3 – DIN1 4 – DIN2 5 – DIN3 6 – DIN4 7 – DIN5 8 – DIN6 9 … 11 – Reserved

Note the following for the Platinum drive:

- Reserved values have no effect and do not return an error.
- The actual number of Capture Modules depends on the drive hardware. Typically, the Platinum drive has two Capture Modules. The drive firmware allows settings to all indexes even if the hardware is not available.

The **GI[]** values for the Platinum Quartet drive are presented on following table:

Index	Description	Default	Value/Connected Input
1 - 8	Capture Modules 1 – 8 routing (see Notes after the table)	0	0 – Axis 1 Port A Index 1 – Axis 1 Port B Index 2 – Axis 2 Port A Index 3 – Axis 2 Port B Index 4 - Axis 3 Port A Index 5 - Axis 3 Port B Index 6 - Axis 4 Port A Index 7 - Axis 4 Port B Index 8 – Axis 1 DIN 4 9 – Axis 2 DIN4 10 – Axis 3 DIN4 11 – Axis 4 DIN4

Note the following for the Platinum Quartet:

- Reserved values have no effect and do not return an error.
- The actual number of Capture Modules depends on the drive hardware. Typically, the Platinum Quartet drive has 4 (Quartet 4 axes) or 6 (Quartet 4 axes) Capture Modules. The drive firmware allows the settings to all indexes even if the hardware is not available.

The following table details the **GI[]** options for the **Titanium** drive only:

Index	Description	Default	Value/Connected Input
1 - 8	Capture Modules 1 – 8 routing (see Notes after the table)	0: for Axis 1 2: for Axis 2	0 – Axis 1 Port A Index 1 – Axis 1 Port B Index 2 – Axis 2 Port A Index 3 – Axis 2 Port B Index 4 – DIN1 5 – DIN2 6 – DIN3 7 – DIN4 8 – DIN5 9 – DIN6 10 … 11 – Reserved

Note the following for the Titanium drive:

- Reserved values have no effect and do not return an error.
- The actual number of Capture Modules depends on the drive hardware. Typically, the Titanium drive has two Capture Modules, i.e. every axis has one dedicated Capture Module. The drive firmware allows settings to all indexes even if the hardware is not available.

References

HM[], **HF[]**, DS-402 Homing Mode, Touch-Probe

GO[N] – Digital Output Source

Routes a dedicated function to the desired digital output.

CANopen/CoE

Alias objects: 0x30F0

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	All
Restrictions	None
Range	See value definitions
Index range	1 to 16
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

Notes:
Consult the Administrative Guide for the actual abilities and restrictions of the outputs

GO[] Indices and Values

The following table describes the values of the **GO[1]…GO[8]** commands. **GO[5]**, **GO[6]** are not used. When writing they have no effect.

For Platinum Drive:

Sub-index 1 …8	Description
1, 3	Values: **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number **1 -** Advanced Output compare module 1 defined with **OC[1]…OC[20]** is routed to output N **2 -** Advanced Output compare module 2 defined with **OC[21]…OC[40]** is routed to output N 7 – STO status

Sub-index 1 …8	Description
7	Values: **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number
2, 4, 8	Values: **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number **7 –** STO status

For Platinum Quartet

Sub-index 1 …8	Description
2	Values: **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number **1 -** Output Compare 1 for Axis 1; Output Compare 2 for Axis 2 **2 -** Output Compare 2 for Axis 1; Output Compare 1 for Axis 2 **3 –** Axis 1 STO status **4 –** Axis 2 STO status **5 –** Axis 3 STO status **6 –** Axis 4 STO status **7 –** Reserved **8 –** Emulation1 channel A for Axis 1 and Axis 2 **9 –** Emulation2 (reserved)
4	Values: **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number **1 -** Output Compare 1 for Axis 1; Output Compare 2 for Axis 2 **2 -** Output Compare 2 for Axis 1; Output Compare 1 for Axis 2 **3 –** Axis 1 STO status **4 –** Axis 2 STO status **5 –** Axis 3 STO status **6 –** Axis 4 STO status **7 –** Reserved **8 –** Emulation1 channel B for Axis 1 and Axis 2 **9 –** Emulation2 channel B (reserved)
6	Values: **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number **1 -** Output Compare 1 for Axis 3; Output Compare 2 for Axis 4 **2 -** Output Compare 2 for Axis 3; Output Compare 1 for Axis 4

Sub-index 1 …8	Description
	3 – Axis 1 STO status **4 –** Axis 2 STO status **5 –** Axis 3 STO status **6 –** Axis 4 STO status **7 –** Reserved **8 –** Emulation1 Channel A for Axis 3; and Axis 4 **9 –** Emulation2 Channel A (Reserved)
8	Values: **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number **1 -** Output Compare 1 for Axis 3; Output Compare 2 for Axis 4 **2 -** Output Compare 2 for Axis 3; Output Compare 1 for Axis 4 **3 –** Axis 1 STO status **4 –** Axis 2 STO status **5 –** Axis 3 STO status **6 –** Axis 4 STO status **7 –** Reserved **8 –** Emulation1 channel B for Axis 3 and Axis 4 **9 –** Emulation2 channel B (Reserved)
1, 3, 5, 7	Reserved

For Titanium Drive:

Sub-index	Description
1, 2	Values are relevant for Safety and for STO drivers **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number **1 -** Advanced Output Compare module 1 defined with **OC[1]…OC[20]** is routed to output N **2 -** Advanced Output Compare module 2 defined with **OC[21]…OC[40]** is routed to output N **7 –** STO status
3, 4	Values are relevant for Safety drivers. Reserved for Regular IO drivers () **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number **1 -** Advanced Output Compare module 1 defined with **OC[1]…OC[20]** is routed to output N **2 -** Advanced Output Compare module 2 defined with **OC[21]…OC[40]** is routed to output N

Sub-index	Description
	7 – STO status
5, 6	Reserved
7, 8	Values are relevant for Safety and for STO drivers **0 -** Digital Output: The function of the output N is defined according to **OL[*N*]** functionality, where N is the Digital output number

GO[0] read access returns the number of sub-indexes, write access returns error**.**

GO[9] - **GO[16]** are reserved.

For all reserved indices read access returns 0, write access has no effect.

Write access with reserved values returns error 21 "Out Of Range".

References

GV[], **GW[]**, **OC[], GC[]**, **OL[]**, **OB[]**, **OP**

GS[] – Gain Scheduling

GS[] defines the gain scheduling process of the position and velocity controller.

CANopen/CoE

Elmo Alias Object: 0x30F4

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	NA
Range	According to the table
Index range	1 to 20
Default	According to the array index, elsewhere is zero
Unit modes	NA
Non-Volatile	Yes
Axis Related	Yes

Remarks

The Platinum drive's controllers and filters can be automatically adjusted according to the speed, position or profile status. The process of assessing the situation and varying the controller parameters on the fly is called "gain scheduling".

- Gain scheduling by speed may be necessary due to either the difference between the low-speed behavior and the high-speed behavior of the plant or a lack of feedback information in low speed.
- Gain scheduling by position may be necessary due to mechanical changes with position dependence.
- Gain scheduling by profile status ("Best Settling" scheduling) may be necessary when high performance of settling time is needed.

There are up to 63 controllers in the position/speed loop. You can use a specific controller or let the gain scheduling mechanism choose controller automatically according to the selected method.

When PIP controller gain scheduling is disabled (by **GS[2]**), the controller is configured by the parameters **KP[3]**, **KP[2]**, and **KI[2]**. When gain scheduling is enabled, the PIP controller is chosen from the 63 controllers in the **KG[*N*]** array. In Best Scheduling mode only 3 controllers are used.

There are also schedules for three controller filters, two for velocity loop filters (**GS[16]** and **GS[17]**) and one for position loop filter (**GS[18]**). Filter type is set by a relevant **KV[N]** and filter parameters are set by a relevant **KA[N]/KB[N]/KC[N]**.

In general, the following applies:

- The parameters **GS[2, 16, 17, 18]** define the gain scheduling mode of the controllers and filters.
- The parameters **GS[1, 7, 13, 10]** are used when gain scheduling by speed is active.

- The parameters **GS[1,11]** are used for Best Settling scheduling.
- The parameters **GS[19,20]** are used when gain scheduling by position is active.

The **GS[*N*]** array is normally programmed by the EAS. It should be noted that modifying the GS table manually can result in instability of the control system.

Indices

The following table lists the gain scheduling parameters. Unused indices are reserved for compatibility with older drives.

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Default</th><th>Values</th><th>Restrictions</th></tr>
<tr><td>0</td><td colspan="2">Reserved</td><td></td><td></td><td></td></tr>
<tr><td>1</td><td colspan="2">When GS[2] defines the minimum speed used in gain scheduling by speed. Below this speed the gain schedule will choose the first (low-bandwidth) controller. The maximum speed for gain scheduling is defined by GS[13]. Units are counts/sec.
When GS[2]=66, defines the threshold profile speed below which the algorithm assumes that the profiler stopped and that a different controller is set.</td><td>100</td><td>>=1
Max 10^{11}</td><td><GS[13]</td></tr>
<tr><td rowspan="6">2</td><td colspan="2">Use scheduled controller according to the following:</td><td rowspan="6">0</td><td rowspan="6">0 to 68</td><td rowspan="6"></td></tr>
<tr><td>0</td><td>No gain scheduling (by KP[2], KI[2], KP[3])</td></tr>
<tr><td>1 to 63</td><td>Specific controller from table KG[N]</td></tr>
<tr><td>64</td><td>Gain scheduling by speed (use GS[7] to set the speed source and position error, which are multiplied by GS[10])</td></tr>
<tr><td>65</td><td>Gain scheduling by position (see Notes)</td></tr>
<tr><td>66</td><td>Best Settling scheduling.
1. During profiler – index 63
KI=KG[63]
KPvel=KG[126]
KPpos=KG[189]
2. After the profiler stops and the time is less than GS[11] (ms) – index 62.
KI=KG[62]
KPvel=KG[125]
KPpos=KG[188]</td></tr>
</table>

Index	Description	Default	Values	Restrictions
	3. The time is **GS[11]** (ms) or more after the profiler stops – index 61 KI=**KG[61]** KPvel=**KG[124]** KPpos=**KG[187]** 67 – Controller index is manually scheduled: **Index is by Object 0x36E0 low 8 bits** 68 – Controller index is manually scheduled: **Index is by Object 0x36E0 high 8 bits**			
3	Reserved			
4	Upward gain of gain scheduling filter. Units are Hz.	1500	100 to 3000	
5	Downward gain of gain scheduling filter. Units are Hz.	1500	100 to 3000	
6	Reserved			
7	Speed source for gain scheduling by speed 0 – By command 1 – By feedback		0, 1	
8	**Position FIR filter (XA[7]) enable mode** 0 – Always enabled 1 – **Enabled after delay of GS[11] ms after the profiler stops** 2 – Enabled after the profiler stops	0	0, 1, 2	**For 1 or 2 Best Settling scheduling must be selected (one of GS[2, 16, 17, 18] is equal to 66).**
9	**Non-linear factor for position controller. This value limits the position controller output to a specific acceleration to close the position error.** **Abs(KP[3]*Error) <= sqrt(2*GS[9]*abs(Error)).** Units are counts/sec^2.	10^{14}	0 to 10^{14}	

Index	Description	Default	Values	Restrictions
10	**With GS[2]=64. Position error coefficient for position gain scheduling to raise gains. Actual speed for gain scheduling is GS[10]*abs(Error)+abs(speed).** Units are rad/sec.	54	0 to 1200	
11	**With one of GS[2, 16, 17, 18] =66. Time limit for Best Settling scheduling.** Sets the time after the profiler has stopped (in milliseconds) when the second controller will be used, and afterwards the third controller is used.	100	0 to 8000	
12	Defines functionality of object 0x3605 0 – **Can be used as position when connected to socket (sensor ID=29)** 1 – Use as velocity integral drain, for current output 2 – Use as Auto focus (position error in count) sensor 3 – Use as asymmetric current limits. The Low 16 bits for minimum limit, and the high 16 bits for the maximum limit. Measured in milli-percentage of the Rated current denoted via 0x6075.0.	0	0 to 3	
13	**Defines the maximum speed for gain schedule by speed (one of GS[2,16,17,18] is equal to 64)**	6200	>=1 Max 10^{11}	**Should be higher than GS[1]**
14	**Defines the position error threshold for the gain scheduling by position error (one of GS[2,16,17,18] is equal to 69)**			
15	**Defines the delay in [millisecond] for scheduling by position error (one of GS[2,16,17,18] is equal to 69)** **Sets the time after the position error drops under GS[14] when the second controller is used, and afterwards the third controller will be used.**	100	0 to 8000	

Index	Description	Default	Values	Restrictions
16	**Use scheduled gains in velocity advanced filter #1, to disable zero KV[25]** 1 to 63 – **Specific controller from table KA[*N*]** 64 – **Gain scheduling by speed (use GS[7] to set the speed source)** 65 – **Gain scheduling by position (see Notes)** 66 – Best Settling scheduling. 1. During profiler operation – index 63. 2. After the profiler stops and the time is less than **GS[11]** (ms) – index 62. 3. The time is **GS[11]** (ms) or more after profiler stops – index 61. **Use KV[25] to set the type of this filter (and also to cancel it).** **The 63*4 parameters are set in KG[190] to KG[441].** 67 – **Filter schedule index is manually selected by Object 0x36E0 low 8 bits** 68 – **Filter schedule index is manually selected by Object 0x36E0 high 8 bits** 69 – Gain scheduling by position error. 1. When position error is larger than threshold **GS[14]** – index 60 2. After the position error falls under **GS[14]** and the time is less than **GS[15]** (ms) – index 59. 3. The time is **GS[15]** (ms) or more after the position error falls under **GS[14]** – index 58	1	1 to 68	
17	**Use scheduled gains in velocity advanced filter #2, to disable zero KV[30]:** 1 to 63 – **Specific controller from table KB[*N*]** 64 – **Gain scheduling by speed (use GS[7] to set the speed source)**	1	1 to 68	

Index	Description	Default	Values	Restrictions
	65 — **Gain scheduling by position (see Notes)**			
	66 — Best Settling scheduling. 1. During profiler operation – index 63. 2. After the profiler stops and the time is less than **GS[11]** (ms) – index 62. 3. The time is **GS[11]** (ms) or more after the profiler stops – index 61.			
	Use KV[30] to set or cancel the type of this filter. The 63*4 parameters are set in KG[442] to KG[693].			
	67 — **Filter schedule index is manually selected by Object 0x36E0 low 8 bits**			
	68 — **Filter schedule index is manually selected by Object 0x36E0 high 8 bits**			
	69 — Gain scheduling by position error. 1. When position error is larger than threshold **GS[14]** – index 60 2. After the position error falls under **GS[14]** and the time is less than **GS[15]** (ms) – index 59. 3. The time is **GS[15]** (ms) or more after the position error falls under **GS[14]** – index 58			
18	**Use scheduled position advanced filter, to disable zero KV[45]:**	1	1 to 68	
	1 to 63 — **Specific controller from table KC[*N*]**			
	64 — **Gain scheduling by speed (use GS[7] to set the speed source)**			
	65 — **Gain scheduling by position (see Notes)**			
	66 — Best Settling scheduling. 1. During profiler operation – index 63.			

Index	Description	Default	Values	Restrictions
	2. After the profiler stops and the time is less than **GS[11]** (ms) – index 62. 3. The time is **GS[11]** (ms) or more after the profiler stops – index 61.			
	Use KV[50] to set the type of this filter (and also to cancel it). **The 63*4 parameters are set in KG[694] to KG[945].**			
	67 – **Filter schedule index is manually selected by Object 0x36E0 low 8 bits**			
	68 – **Filter schedule index is manually selected by Object0x36E0 high 8 bits**			
	69 – Gain scheduling by position error. 1. When position error is larger than threshold **GS[14]** – index 60 2. After the position error falls under **GS[14]** and the time is less than **GS[15]** (ms) – index 59. 3. The time is **GS[15]** (ms) or more after the position error falls under **GS[14]** – index 58			
19	**First position boundary for gain scheduling by position. This value together with GS[20] defines the position that is divided by 63 for gain scheduling. Units are counts.**	0		
20	**Second position boundary for gain scheduling by position. This value together with GS[19] defines the position that is divided by 63 for gain scheduling. Units are counts.**	0		

Notes

When setting parameters of the scheduled filters (**KV[25]**, **KV[30]** and **KV[45]**), only the relevant filters in the table are checked and set, according to **GS[16,17,18]**. It is therefore advisable to first set **GS[16,17,18]** and then **KV[25,30,45].**

When using gain scheduling by position (**GS[2,16,17,18]=65**):

- If position modulo is defined, the scheduling is by modulated feedback position;
- If position modulo is not defined, the scheduling position source is defined by **SA[7]** and position limits are defined by **GS[19]/GS[20]**

The PIP controllers table **KG[]** can be changed when servo is enabled (**MO** = 1).

References

CA[], **KG[]**, **KV[]**

GT[N] – Capture Time MUX Selection

GT[N] routes Port A Index, Port B Index, Port C Index or digital inputs into Capture Time 1 – Capture Time 4 input signal multiplexor. The command is device related.

CANopen/CoE

Elmo parameters objects: 0x30F5

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	CANopen, CoE, RS232, USB, TCP, EoE
Restrictions	None
Range	Not applicable
Index range	1 to 4
Default	Refer to the **GT[*N*]** command format table below.
Unit modes	All
Non-Volatile	Yes
Attribute	None
Axis Related	No

Remarks

The Platinum drive includes 4 Capture Time modules. These modules are used, among other things, to capture the CPLD time in resolution of 20ns, including "Event; bit 15 of an input signal.

The **GT[N]** command enables routing of Port A, B, C Index or general purpose digital inputs into Capture Time 1-Capture Time 4 modules. This allows any time capturing from various types of inputs.

Note:
Different drives may have different number of available digital inputs and time modules. Consult the specific drive Installation Guide for details. The drive should not prevent any setting of a non-existing input.

The command includes the input to be routed and the Capture Time module to which it is routed.

This command must be configured before using DS-402 homing, DS-402 touch probe, and in some cases of Elmo's legacy homing (HM,HF) sequences.

Indices

The following table details the **GT[]** values for the **Platinum drive**:

Index	Description	Default	Bits	Value	Function
1 - 4	Capture Time module 1 – Capture Time module 4 routing	0	0	0/1	Mode 1 (Time tag) Capture Time on rising/falling edge
			1	0/1	Mode 2 (reserved) Capture Time on rising/falling edge
			2	0/1	Mode 3 (reserved) Capture Time on rising/falling edge
			3	0/1	Mode 4 (reserved) Capture Time on rising/falling edge
			4	0/1	Mode 1 (Time tag) Disable/Enable
			5	0/1	Mode 2 (reserved) Disable/Enable
			6	0/1	Mode 3 (reserved) Disable/Enable
			7	0/1	Mode 4 (reserved) Disable/Enable
			Bits 8 - 15	0	Port A Index
				1	Port B Index
				2	Port C Index
				3	DIN1
				4	DIN2
				5	DIN3
				6	DIN4
				7	DIN5
				8	DIN6
				9 – 65535	Reserved

Reserved values of bits 0 to 7 have no effect and do not return error. Reserved values of bits 8 to 15 return error 21 "Out of Range"

The following table details the **GT[]** values for the **Platinum Quartet**.

Index	Description	Default	Value and function		
			Bits	**Value**	**Function**
1 - 4	Capture Time module 1 – Capture Time module 4 routing	0	0	0/1	Mode 1 (Time tag) Capture Time on rising/falling edge
			1	0/1	Mode 2 (reserved) Capture Time on rising/falling edge
			2	0/1	Mode 3 (reserved) Capture Time on rising/falling edge
			3	0/1	Mode 4 (reserved) Capture Time on rising/falling edge
			4	0/1	Mode 1 (Time tag) Disable/Enable
			5	0/1	Mode 2 (reserved) Disable/Enable
			6	0/1	Mode 3 (reserved) Disable/Enable
			7	0/1	Mode 4 (reserved) Disable/Enable
			Bits 8 - 15	0	Axis 1 Port A Index
				1	Axis 1 Port B Index
				2	Axis 2 Port A Index
				3	Axis 2 Port B Index
				4	Axis 3 Port A Index
				5	Axis 3 Port B Index
				6	Axis 4 Port A Index
				7	Axis 4 Port B Index
				8	Axis 1 DIN 4
				9	Axis 2 DIN 4
				10	Axis 3 DIN 4
				11	Axis4 DIN 4
				12 - 65535	Reserved

Reserved values of bits 0 to 7 have no effect and do not return error. Reserved values of bits 8 to 15 return error 21 "Out of Range"

The following table details the **GT[]** options for the **Titanium** drive :

Index	Description	Default	Value and function		
1 - 4	Capture Time module 1 – Capture Time module 4 routing	0	**Bits**	**Value**	**Function**
			0	0/1	Mode 1 (Time tag) Capture Time on rising/falling edge
			1	0/1	Mode 2 (reserved) Capture Time on rising/falling edge
			2	0/1	Mode 3 (reserved) Capture Time on rising/falling edge
			3	0/1	Mode 4 (reserved) Capture Time on rising/falling edge
			4	0/1	Mode 1 (Time tag) Disable/Enable
			5	0/1	Mode 2 (reserved) Disable/Enable
			6	0/1	Mode 3 (reserved) Disable/Enable
			7	0/1	Mode 4 (reserved) Disable/Enable
			Bits 8 - 15	0	Axis 1 Port A Index
				1	Axis 1 Port B Index
				2	Axis 2 Port A Index
				3	Axis 2 Port B Index
				4	DIN1
				5	DIN2
				6	DIN3
				7	DIN4
				8	DIN5
				9	DIN6
				10 - 65535	Reserved

Reserved values of bits 0 to 7 have no effect and do not return error. Reserved values of bits 8 to 15 return error 21 "Out of Range"

References

HM[], **HF[]**, DS-402 Homing Mode, Touch-Probe

GV[*N*] – Output Compare Editing Table

GV[N] edits entries in the position table for Output Compare modules 1 and 2, that is defined by commands **OC[n]**

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	double, Read/Write
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	-1.0e307 to 1.0e307
Index range	1…1024
Default	None
Unit modes	All
Non-volatile	No
Axis related	No

Remarks

It should be noted that **GV[]**, **GX[]**, and **GY[]**, commands use the same capture array, than, when running simultaneously Capture **HM**, **HF**, Output Compare 1 and 2 the different indices of that array should be used. For example: **GV[1]-GV[256]** for **OC1**, **GV[257]-GV[512]** for **OC2**, **GX[513]-GX[768]** for **HM** and **GY[769]-GY[1024]** for **HF** can be used in parallel, but not **GV[1]-GV[256]** and **GX[1]-GX[256]**.

The **GV[]** command does NOT protect against editing the table while Output Compare is operational, i.e., **OC[1]/OC[21]** can report 1 or 2.

The **GV[]** values should be set to position table in user units. Note, that the table is converted to counts at the time of Output Compare activation (**OC[1]/OC[21]** = 1,3,4), and read access of **GV[]** command returns converted values

References

OC[], **GO[]**, **GC[]**

GX[] – Capture Array Value from HM

GX[] retrieves captured values from the capture array.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Double, Read-only
Source	RS232, USB, TCP, EoE
Restriction	
Range	None
Index range	1 to 1024
Default	None
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

Avoid using the same array entries for different axes or different services, e.g. same array indices are used by **HM[]** and **HF[]** at the same time, which produces ambiguous results.

During position capture, the set of captured positions must be defined by **HM[12], HM[13]** in an array. In that case the **GX[]** command can be used to retrieve the position inputs.

This command reads the captured array, saving the user the need to know how the software stores the captured values in the array.

GX[] always returns the value that is currently in the capture array, according to the specified index.

Note that the available range of array indexes to be set by **HM[12]** (Start index), **HM[13]** (End index) is 1...1024. The available range of **HM[1]** is 1...32000 (32000 is called "Infinite").

- If **HM[1]**="Infinite" - then captured positions are written into an array continuously, with rollover from End index to Start index.
- If **HM[1]** is set as: **(End index - Start index + 1) < HM[1] < Infinite**
  then **HM[1]** captured positions are written into an array with rollover from End index to Start index
- If **HM[1]** is set as: **HM[1] <= (End index - Start index + 1)**
  then **HM[1]** captured positions are written into array from indexes **Start index** to **End index**. In that case rollover is not necessary.
- Reading of an array index greater than 1024 returns error 3 "Bad index"
- Reading of an array less than 1024 but outside the range **End index** to **Start index** returns a junk value. An Error code is not released in this case.

- The next index to be filled is indicated in **HM[9]**.

References

HM[]

H1….H8[] – Halls Socket Parameters

H1….H8[] sets the halls configuration attached to a specific socket No.

CANopen/CoE

Elmo Alias Object: 0x30FD - 0x3104

Attributes

Attribute	Description
Type	Unsigned Short
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 6
Default	According to array index
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

It should be noted that modification of the command should be performed using the EAS Quick or Expert Tuning interface. Manual modification may result in ambiguous results from the sensor or function that uses it.

The **H1…H8[]** commands set the halls parameters attached to a socket.

The number of socket is denoted by a number, e.g. **H1** means halls attached to socket 1. There are up to 8 sockets.

To activate the change in parameter the socket sensor ID must be set (**S1…..S8[1]**).

Parameter change can be activated only if socket is not used for commutation, or axis that uses the sensor are disabled.

Changing the setting resets commutation if used.

Indices

The following table details the **H1…H8[]** entries.

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Def.</th><th>Values</th><th>Restrictions /Notes</th></tr>
<tr><td rowspan="11">1</td><td colspan="2">Halls Setting Parameter</td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Bits</td><td>Description</td></tr>
<tr><td>0</td><td rowspan="3">Polarity of Hall sensors A, B and C, respectively</td><td rowspan="3">1</td><td rowspan="3">0 – Reverse polarity
1 – Do not reverse polarity</td><td rowspan="3"></td></tr>
<tr><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3-4</td><td rowspan="3">Correlate between Hall sensors and motor phases A, B and C,</td><td>1</td><td rowspan="3">1 – Hall A
2 – Hall B
3 – Hall C</td><td rowspan="3">Each Hall sensor must be assigned to a different motor phase.</td></tr>
<tr><td>5-6</td><td>2</td></tr>
<tr><td>7-8</td><td>3</td></tr>
<tr><td>9-12</td><td>Halls resource used</td><td>0</td><td>0 – No Halls
1 – Digital Halls Ax1
2 – Digital Halls Ax2
3– Digital Halls Ax3
4 – Digital Halls Ax4
5 – Digital Halls Ax5
6 – Digital Halls Ax6</td><td></td></tr>
<tr><td>13-15</td><td>Reserved</td><td></td><td></td><td></td></tr>
<tr style="display:none"></tr>
<tr><td>2</td><td colspan="2">Hall State 2 Width in stepper units</td><td>682</td><td>0 to 1364</td><td rowspan="5">Sum of all hall states width should not exceed 4096</td></tr>
<tr><td>3</td><td colspan="2">Hall State 3 Width in stepper units</td><td>683</td><td>0 to 1364</td></tr>
<tr><td>4</td><td colspan="2">Hall State 4 Width in stepper units</td><td>683</td><td>0 to 1364</td></tr>
<tr><td>5</td><td colspan="2">Hall State 5 Width in stepper units</td><td>682</td><td>0 to 1364</td></tr>
<tr><td>6</td><td colspan="2">Hall State 6 Width in stepper units</td><td>683</td><td>0 to 1364</td></tr>
</table>

References

S1….S8[], F1…..8[]

HL[] /LL[] – High/Low Feedback Limit

HL[] and **LL[]** specify the minimum and maximum allowed motor speed and position.

CANopen/CoE

HL - Elmo Alias Object: 0x3111

LL - Elmo Alias Object: 0x31A1

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	No
Range	**LL[3]: -2^{52} to $(2^{52} - 1)$** **HL[2]: 0 to 10^{11}** **HL[3]: -2^{52} to $(2^{52} - 1)$**
Index range	**LL[*N*]: N=3** **HL[*N*]: N=2, 3**
Default	**LL[3] = 0** **HL[2] = 0** **HL[3] = 0**
Unit modes	**VH[2]: UM = 1, 2, 5, 7** **VL[3],VH[3]: UM = 5, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

Speed limit:

- The parameter **HL[2]** defines the limits of the allowed motor speed. If the motor speed exceeds **HL[2]** or is lower than (**–HL[2]**), the drive is automatically disabled and the "Over Speed Violation" fault is activated (**MF**=0x20000).
- If **HL[2]** is 0, the "Over speed protection" is deactivated. Disable the drive (Motor Off) in order to activate/deactivate this protection.

Position limits:

- The parameters **HL[3]** and **LL[3]** define the allowed motor position range. If the motor position exceeds **HL[3]** or is smaller than (**LL[3]**), the drive is automatically disabled and the "Out of position limits" fault is activated (**MF**=0x400000).

- If both **HL[3]** and **LL[3]** are 0, the "Out of position limits" protection is deactivated. Disable the drive (Motor Off) in order to activate/deactivate this protection.

Indices

The following table describes the **HL[]/LL[]** entries.

Index	Description	Units	Range
0	Reserved		
1	Reserved		
2	The motor speed is limited to the range **[-HL[2]…HL[2]].**	User-defined	0 to 10^{11}
3	**The motor position is limited to the range [LL[3]...HL[3].**	User-defined	-2^{52} to $(2^{52} - 1)$

Note:
The **HL[]** and **LL[]** values should be given in user-defined position units specified by the **FC** command.

References

VH[N], VL[*N*], FC[N]

HM[]/HF[] – Main/Socket Homing Parameters

HM[] /HF[] enables modification of the axis position (home) and to capture the actual position using a variety of events.

CANopen/CoE

HM[] – Elmo Alias Object: 0x3112

HF[] – Elmo Alias Object: 0x310B

Attributes

Attribute	Description
Type	Double
Source	USB, RS232, TCP, EoE
Restrictions	See table of relevant **HM/HF** array index
Range	See table of relevant **HM/HF** array index
Index range	1 to 13
Default	None
Unit modes	**UM** = 2, 5 –All event definitions are available **(HM[3]/HF[3]=0…20**) **UM** = 1,3,6,7 – Immediate event only is available (**HM[3]/HF[3]=0**)
Non-Volatile	Yes, exclusive **HM[1], HF[1]**
Axis Related	Yes

Remarks

HM[] defines the main homing parameters, which always operate on the main feedback sensor. The sensor socket is selected by the user with **HM[10]** (see later details).

The **HF[]** command defines the additional homing parameters. The sensor socket is selected by the user (see **HF[10])**.

Main position capture via **HM[]** and additional position capture via **HF[]** can be performed simultaneously.

The following description refers to **HM[]**, but also applies to **HF[]** unless stated otherwise.
The command sets and gets parameters of the **HM/HF** homing and capture process, which the drive uses to set a trap for a user-defined event. When the event occurs, the drive can perform one of the following tasks:

- Modify the feedback position as pointed by the sensor selection (**HM[10]** or **HF[10]**) using the value defined by the user in **HM[2]** or **HF[2]**.
- Log the event position prior to the position modification. (See **HM[7]/HF[7]**)
- Flag a digital output (see **HM[6]/HF[6]**)
- Stop the axis motion (see **HM[4]**)

Platinum drive support two types of capture accuracy:

- Index, Home. Using these inputs must be configured by the **GI[]** command

- Digital Inputs 1-6. Software capture with a jitter of (**HS*TS** + **TS**) microseconds and a delay that is influenced by the **IF[]** command (Input filter)

Polarity and functionality of the digital input are defined by the **IL[]** command.

Home events are requested via **HM[3]**. Note that for the digital input relevant functionality to correspond to **HM[3]** (e.g. RLS), it should be defined via **IL[]** prior to the homing activation.

The **HM[]** parameters are used by homing procedure when **HM[1]** is activated i.e. set from 0 to !=0. Setting of the homing parameters is allowed during an on-going homing procedure, the values however will take effect only when the **HM[1]** is set from 0 to != 0. This allows preparing of the next homing routine while in motion.

During Homing there is no restrictions of **XM[1], XM[2]**, **VL[3], VH[3]**.

The Position absolute value, denoted by **HM[2]**, is set to the axis actual position (**PX**, **0x6064**, **PU**) when the event occurs and according to **HM[5]**.

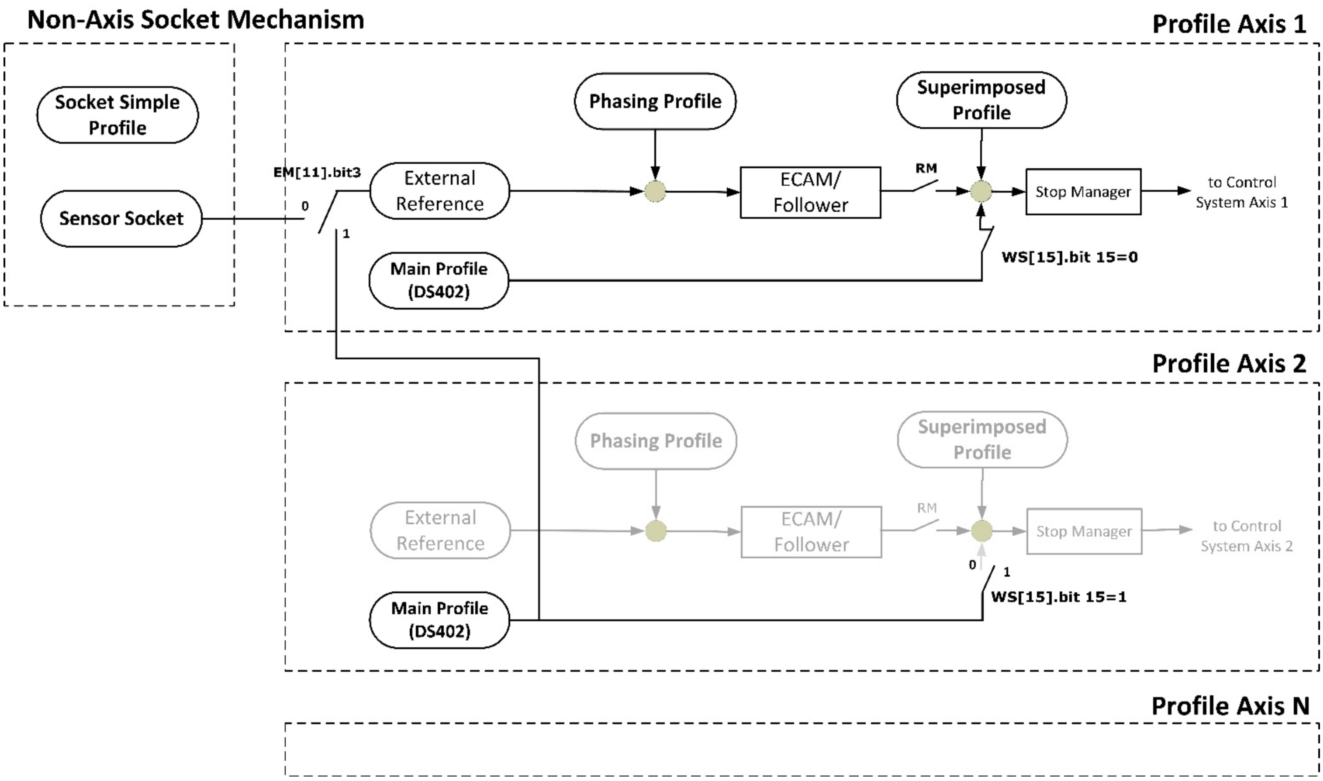


Figure 1: External reference

External reference possible configurations are presented in Figure 1.

If the main profiler socket is selected by **HF[10]**, and **HF[5]** and is set to 0 (absolute settings of the HW position of the socket) the command **HF[1]**=1 (or any not zero value) returns with error.

The following two cases are recommended for **HF** homing only, **HM** is not recommended.

- When **EM[11]:3** =0, sensor socket is connected, the number of the socket is defined by **HF[10]**. Note that **EM[16]** must be set to the same socket number.
- When **EM[11]:3** =1, main profiler of another axis is connected . The number of the connected axis is defined by **EM[16]**. Note that the number of socket defined by **HF[10]** should be the same as defined for the axis selected with **EM[16]**.

The main profiler of an axis can be disconnected from input of Stop Manager by manufactures' specific setting. The status of this setting can be retrieved with command **WS[15]** bit 15, where 1 means "disconnected". In this case only **HF** homing can be performed, as described before. The main profiler of this axis can be used by another axis as external reference.

When **WS[15]** bit 15 == 0, Stop Manager is connected to output of main profiler and **HM** homing can be performed. **HF** homing is not recommended.

When capturing more than one event is to be configured, i.e., when **HM[1]** > 1, the delta between each event must be **HS** microseconds for digital inputs, and **TS** microseconds for home input or Index.

If **HM[1]**=32000, called "Infinite capture", the drive performs endless capture.

The samples of captured position are stored in home capture table within index range of **HM[12]** to **HM[13]**. The next capture table index can be retrieved by the **HM[9]** command. When the next capture table index exceeds the value of **HM[13]** it is cyclically set to value of **HM[12]**.

If **HM[1]** is set to a value that exceeds the number of possible (but not 32000) entries in the home capture table, the total number of capture events will be **HM[1]** regardless of the **HM[13], HM[12]** settings.

When homing was activated by settings **HM[1]**="Value", **HM[1]** decreases every homing event from "Value" to 0, regardless of the **HM[12]**, **HM[13]**, settings.

The following table refers to the logic level of the relevant input. The relation between the physical level and the logic level is determined via the **IL[]** command.

Indices

The following table describes the **HM[]/HF[]** entries.

Index	Purpose	Value	Description
0	Reserved	None	Reserved
1	Activation mode	0 (default)	Stop the homing process. **HM[1]** is automatically reset to 0 when homing is complete.
		1…31999	Number of events. HM[4] is performed and position offset HM[2] is applied at last event.
		32000	Infinite number of event captures. **HM[4]** is ignored, **HM[2]** is never applied. The capture array will be filled between HM[12] and HM[13] in an infinite loop.
2	Homing position offset	-4503599627370496.0 to 4503599627370495.0Default=0	Value set to the absolute position according to the method specified in **HM[5]**. Absolute value is limited to the position counter range.
3	Event definition	0 default	Immediate: The trigger is the receipt of **HM[1] = 1**.
		1 and 2	Home Switch High transition event. High transition is level change from low to high, i.e.

Index	Purpose	Value	Description
			rising edge. The home switch is selected with **GI[2]** command.
		3	Index Pulse High transition event (rising edge). The index pulse source is selected with **GI[1]** command.
		4	Index Pulse low transition event (falling edge). The index pulse source is selected with **GI[1]** command. Logic is set via **IL[]**. **Note**: This option is not supported for serial encoder **FB** when **GI[]** is set to encoder index. No error is indicated
		5 and 6	FLS switch logical high transition event (regardless of motion direction).
		7 and 8	RLS switch logical high transition event (regardless of motion direction) .Logic is set via **IL[]**.
		9 and 10	High level event according to the DIN1 switch.
		11 and 12	High level event according to the DIN2 switch.
		13 and 14	High level event according to the DIN3 switch.
		15 and 16	High level event according to the DIN4 switch.
		17 and 18	High level event according to the DIN5 switch.
		19 and 20	High level event according to the DIN6 switch.
		21 and 22	Reserved
		23 and 24	Reserved
		25 and 26	Reserved
		27 and 28	Reserved
4	After event behavior. Executed when **HM[1]** reaches 0.	0 default	In **UM** = 1,2,3,5,6 and 7: stop immediately using the **SD** deceleration value.
		1	Set digital output. Equivalent to **OP = HM[6]**.
		2	After event, perform no operation
		>2	Reserved
5	Position setting option. Defines the position parameter	0	In the case of **HM** homing: absolute settings of the position in user units: **PX=HM[2],** adapt main feedback position and position command including modulo.

Index	Purpose	Value	Description
	(**PX/FP**) to be set on event.		In the case of HF homing: absolute settings of the hardware position of the socket: **FP=HF[2]**. **Note 1:** Using the **HF[]** in this mode while the **HF[10]** socket selection is not the main position socket a jump in the motion may occur. The drive does not prevent this.. **Note 2:** Error mapping must be disabled
		1	Reserved
		2	Do not update the position. This mode is used for capture mode only.
		3	In the case of **HM** homing: do not update the position. In the case of **HF** homing: Absolute setting of the socket position **FP=HF[2]**, adapt main feedback position and position command including modulo. In case the sensor is defined as Gantry, this will adjust all relevant Gantry sockets (TBD).
		>=4	Reserved
6	Output value	0-0xFFFFFFFF	Digital output value to set if **HM[4] = 1**. Only outputs defined as general outputs are affected.
7	PX Captured value	None	The capture value of **PX/FP**. The position value is captured before **PX/FP** is changed according to **HM[5]**. In the case of **HM** homing the captured value is converted to user units
8	Event counter	None	Read: Counting up from 0 to 65535 and overflowed event counter value. Write: only value 0 is permitted to reset counter
9	Next capture array index.	1 to 1024	The value, indicating the index in the capture array for inserting the next captured value. The captured value can be read via the **GX[]**, **G1[]…G5[]** commands.

Index	Purpose	Value	Description
10	Socket selection	0x00 to 0x08 0x10 to 0x18 0x20 to 0x28; Default=0 (socket for **HM/HF** is not specified, use main profiler socket)	The number of socket and socket signal used in **HM/HF** homing The **HM[10]** command set which socket and socket signal are used for different functions of the axis: Lowest nibble – socket number; Second nibble – socket signal used for axis function: 0: main socket signal; 1: auxiliary 1 socket signal; 2: auxiliary 2 socket signal. For example, **HM[10]=0x12** means that auxiliary signal 1 of socket 2 is used as position feedback. Values: 0x00 to 0x08 0x10 to 0x18 0x20 to 0x28; **Note:** Rest of the bits are reserved, and must be set to 0
11	Capture array selection	0 (default)	Home capture table **HM/HF** array of 1024 entries is selected. The value cannot be changed
12	Home capture table low index	1…1024, default=1	Low and High home capture table index: Position captured value will be stored in this index range. Index will roll if the number of captures exceed the range.
13	Home capture table high index	1…1024, default=1	

Examples

Example 1

The following example uses capture on the main home switch when DIN1 is routed into Capture 2 Home input by the **GI[]** command.

HM[1] will be set to 0 in case that the digital input 1 will be logically set to '1'.

Command	Description
HM[1]=0	Disable the ongoing homing sequence
HM[2]=1000	Home position has been offset by 1000 counts from input
IL[1]=17	DIN1 is configured as Home switch, active high
GI[2]=3	Route DIN1 into Capture 2 Home input (for Gold: **GI[4]=2**)
HM[3]=2	Wait for the event on Home signal (the first rising edge)

Command	Description
HM[1]=1	Start searching for a single event

Example 2

The following example uses capture on the Index signal when DIN2 is routed into Capture 1 Index input by the **GI[]** command.

HM[1] will be set to 0 in case the digital input 2 will be logically set to '1' .

Command	Description
HM[1]=0	Disable the ongoing homing sequence
HM[2]=1000	Home position has been offset by 1000 counts from input
IL[2]=7	DIN2 is configured as general purpose digital input, active high
GI[1]=4	**Route DIN2 into Capture 1 Index input (for Gold: GI[3]=3)**
HM[3]=3	Wait for the event on Index signal (the first rising edge)
HM[1]=1	Start searching for a single event

Example 3

The following example uses capture on the motor Index signal that is routed into Capture 1 Index input by the **GI[]** command. The number of events is set to 20. **HM[4]** is performed and position offset **HM[2]** is applied at the last event.

HM[1] will be set to 0 in case the motor index signal will be logically set to '1' at the last event.

Command	Description
HM[1]=0	Disable the ongoing homing sequence
HM[2]=1000	Home position has been offset by 1000 counts from input
GI[1]=0	Route Port A motor index into Capture 1 Index input (for Gold: **GI[3]=1**)
HM[3]=3	Wait for the event on Index signal (the 20th rising edge)
HM[1]=20	Start searching for events

Example 4

The following example uses capture on the main home switch when DIN5 is routed into Capture 2 Home input by the **GI[]** command, and position captured values are stored in capture array infinitely between indexes 1 to 200.

Command	Description
HM[1]=0	Disable the ongoing homing sequence.
HM[2]=0	Home position is offset by 0 from the input.
IL[5]=17	DIN5 is configured as Home switch, active high.
GI[2]=7	**Route DIN5 into Capture 2 Home input (for Gold: GI[4]=6).**
HM[3]=1	Wait for the event on home signal (rising edge)
HM[12]=1	The low capture array index is 1.

Command	Description
HM[13]=200	The high capture array index is 200.
HM[1]=32000	**Start capturing events infinitely and add them into the capture array between indexes 1 and 200. HM[9] will indicate the next array index to be filled. The values are read using the GX[] command.**

Reference

GI[], **GX[]**, **GY[]**, **IL[]**, **IF[]**

HP – Halt Program

UP[N] Halt User Program.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Unsigned Char, Read/Write
Source	All
Restrictions	None
Range	0 to 255
Default	0
Index Range	1 to 7
Unit modes	All
Non-volatile	No
Attribute	No

Remarks

The command halts the execution of the user program.

A later XC command shall resume the program from the instruction at which the program was halted. Pending Auto Routine will remain pending.

An HP command issued when no program is running does nothing and sets no error.

HP command is per thread:

Index	Description
1	Halt thread 0 (main).
2	Halt thread 1.
3	Halt thread 2.
4	Halt thread 3.
5	Halt thread 4.
6	Halt thread 5.
7	Halt **all** threads.

Notes

- **XC** command resumes the execution after a halt.
- Thread Program status (**PS[2]**) indicates 0 when program is halted.

References

XC[], **PS[]**, **WS[37]**, **CT[]**, **KL[]**

HS – Profile Sampling Factor

HS specifies the sampling time of the profile, **HS*TS** microseconds.

CANopen/CoE

Elmo Alias Object: 0x3118

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	All
Restrictions	**MO = 0, new value activated after drive reboot**
Range	1 to 10
Default	According to the table below
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

HS*TS is the sampling time of the profile, fault handling and control word handler algorithms.

Motor enable is not allowed after a **HS** change until the drive reboots.

When **TS** is modified, **HS** automatically receives the default value:

TS, [μsec]	Default HS	Minimal HS
50.0	5	2
62.5	4	2
100.0	2	1
125.0	2	1

The default value can be overwritten by the **HS** command.

The drive does not allow an excessively low value for **HS**, to prevent an overflow of the required CPU computing power (see minimal **HS** in above table).

References

TS

HT[] Stepper Torque Command

HT[] specifies the stepper torque commands for stepper 2-phase (**UM**=6) and closed loop stepper (**UM**=7) modes

CANopen/CoE

Elmo Alias Object: 0x3119

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	NA
Range	**0 to PL[1]**
Index range	1 to 3
Default	0
Unit modes	**UM = 6, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

The **HT[]** command sets the torque command for 3 different movement types: no movement, movement at certain speed without acceleration and movement with some acceleration.

In addition to the specified torque, there is an option to add gain proportional to the velocity or the acceleration using **FF[]** command.

Indices

The following table describes the available options for **HT[]**.

Index	Description	Units	Values	Restrictions
1	Holding torque if there is no speed and no acceleration	Amperes	**0 to PL[1]**	
2	**Torque when there is speed and no acceleration. Receives the maximum value when between HT[1] and HT[2]. If HT[1] > HT[2] the maximum value obtained is HT[1].**	Amperes	0 to PL[1]	
3	Torque when there is acceleration. **Receives the maximum value when between HT[1] and HT[3].**	Amperes	0 to PL[1]	

Index	Description	Units	Values	Restrictions
	If HT[1] > HT[3] the maximum value obtained is HT[1].			

References

FF[], PL[], UM

HX[] – Hex Format (Interpreter)

HX[] command defines that the answer from the drive will be in hex-decimal representation.

CANopen/CoE

Elmo Alias Object: 0x311D

Attributes

Attribute	Description
Type	Read/Write
Source	All
Restrictions	
Range	0/1
Index range	1
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

Setting ‘1’ into **HX** command (**HX=1**) defines that the answer from the interpreter (drive) returns in hex-decimal format.

Setting ‘0’ into **HX** command (**HX=0**) defines that the answer from the interpreter (drive) returns in decimal format.

IB[] – Digital Input Bits

IB[] reads a digital input bit. The command is axis related.q

CANopen/CoE

Digital_inputs: 0x60FDElmo Alias Object:
0x312B, 0x382B, 0x412B, 0x482B, 0x512B,
0x582B

Attributes

Attribute	Description
Type	Long Indices 0 to 16 are RO, indices 17 to 32 are R/W
Source	All
Restrictions	None
Range	0, 1
Index range	1 to 32
Default	None
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

IB[1] to **IB[32]** reflect the **IP** register bits 0 to 31, respectively.

Refer to the **IP** command for more details about the digital input function and state.

The 'Write" function clears a fetched input (sticky bit). Refer to **IL[]** command "sticky bit" section.

In the Quartet drive Ports, A1 – A4, B1, B3 can be presented in IB[] command instead of digital inputs. Refer to **GB[]** command for detailed information.

Write access to sub-index 0 returns error 3 "Bad index".

Write access to sub-indices 1-16 has no effect and does not return an error.

Example

IB[1] reflects **IP** bit 0, indicating a general-purpose function state is active (1)or not active (0).

IB[17] reflects **IP** bit 16, indicating that digital input 1, regardless to its function, is active (1)or not active (0).

Indices

Index	Description	Type	Values	Restrictions
1	General purpose input is active	Long	0..1	

Index	Description	Type	Values	Restrictions
2	Safety (O.K.)	Long	0..1	
3	Main home switch	Long	0..1	
4	Auxiliary home switch	Long	0..1	
5	Soft stop	Long	0..1	
6	Hard stop	Long	0..1	
7	Forward limit (FLS)	Long	0..1	
8	Reverse limit (RLS)	Long	0..1	
9	INH (enable) switch	Long	0..1	
10	Hardware BG (begin)	Long	0..1	
11	Abort function	Long	0..1	
12..16	Not used. Always zero.	0		
17..22	Digital input 1..6 logical pins state	Long	0..1	Write 1 to clear sticky bit, Write 0 has no effect
23..30	Reserved. Always 0	0		
31..32	Digital input 15..16 logical pins state	Long	0..1	Write 1 to clear sticky bit, Write 0 has no effect

References

IP, **IL[*N*], GB[]**

ID, IQ – Active/Reactive Current

ID and **IQ** obtain the active (**IQ**) and the reactive (**ID**) components of the motor current, in amperes.

CANopen/CoE

ID - Elmo Alias Object: 0x312D

IQ - Elmo Alias Object: 0x313A

Attributes

Attribute	Description
Type	Float, Read-only
Source	All
Restrictions	None
Range	NA
Index range	NA
Default	NA
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

A brushless motor carries alternating currents in its phases. The alternating currents in the motor phases create a rotating magnetic field, which can be projected in two directions. The first magnetic field component is aligned with the magnetic direction of the rotor; it produces no mechanical torque. The other magnetic field component is perpendicular to the magnetic direction of the rotor and produces all the mechanical torque.

IQ [ampere] is the component of the motor phase current that creates the effective torque. The current controller attempts to make **IQ** equal to the current command. **ID** is the component of the motor phase current that does not create torque. Usually the current controller tries to null **ID**.

When the motor is off (**MO** = 0), **IQ** and **ID** are not calculated and return 0.

References

- Language and User Program Manual: Chapter 10, "The Current Controller"

AN[], **MC**, **PL[*N*]**, **CL[*N*]**

IF[] – Digital Input Filter

IF[] defines the time period of the digital input filter. The command is non-axis related.

CANopen/CoE

Elmo Alias Object: 0x312F

Attributes

Attribute	Description
Type	Parameter, Float
Source	All
Restrictions	None
Range	See table below
Index range	1 to 20
Default	0 (no filter)
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The following table describes the conditions for the specific **IF[]** groups:

IF Index	Description
IF[1] – IF[16]	Digital input filter in milliseconds. Each input points to the referenced index from 1 to 16. Input level is considered as valid when the level remains steady for the period of the filter time. Pulses which are shorter than the filter time will be discarded, and the filter starts again. The actual filter resolution is based on the High Speed cycle which is the multiplication of **TS** * **HS** µSecs (e.g. if **TS**=50 and **HS**=2 then the filter resolution is 100uSec). The value from the user in mSecs is rounded down (floor function) to the nearest value per the resolution as described above. For example, if **TS**=50, **HS**=5 which means 250uSec resolution and **IF[1]** is set to 10.48, the executable filter time will be set to 10.25 mSecs. For inputs that are not supported by the product, the relevant index will be accepted but ignored.
IF[17] – IF[20]	The **IF[17] – IF[20]** are high resolution filters per input as described in the Indices table below. Filter resolution is in 100uSec e.g. if **IF[17]=15** the filter is set to 1.5 mSecs. The high resolution filter will be added to the filters as described in **IF[1]-IF[16]** above.

Indices

The following table describes the **IF[]** entries.

Index	Description	Range	Comments
1 to 16	Digital input filter for DIN 1 – 16 in milliseconds	0.00 to 2000.00	
17	HW Digital Input Filter in 100 microseconds resolution	0 to 15	**For Platinum and Titanium drives**: HW implemented Digital Input filter in the range of 0 – 1.5 ms. The same value for **D_{IN}** 1, 2, 3, 4, 7, 8. Remaining inputs do not have HW filter **For Platinum Quartet**: HW implemented Digital Input filter in the range of 0 – 1.5 ms. Relevant for **D_{IN}** 4 only (Axis 1 Input 4)
18	HW Digital Input Filter in 100 microseconds resolution	0 to 15	**For Platinum Quartet only**: HW implemented Digital Input filter in the range of 0 – 1.5 ms. Relevant for **D_{IN}** 8 only (Axis 2 Input 4)
19	HW Digital Input Filter in 100 microseconds resolution	0 to 15	**For Platinum Quartet only**: HW implemented Digital Input filter in the range of 0 – 1.5 ms. Relevant for **D_{IN}** 12 only (Axis 3 Input 4)
20	HW Digital Input Filter in 100 microseconds resolution	0 to 15	**For Platinum Quartet only**: HW implemented Digital Input filter in the range of 0 – 1.5 ms. Relevant for **D_{IN}** 16 only (Axis 4 Input 4)

References

IB[], **IL[]**, **IP**

IL[] – Digital Input Logic

IL[] specifies the function and logic of the digital inputs. The command is axis related

CANopen/CoE

Elmo Alias Object: 0x3135

Attributes

Attribute	Description
Type	Unsigned long, Read/Write, Bit Filed
Source	All
Restrictions	The routing of the desired input for capture purposes is done via **GI** command.
Range	NA
Index range	1 to 16
Default	Input 1 to 16: 5 – Ignore
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

IL[*N*] is a bit field command which allows the user to map digital input *N* to a desired function and to determine the logic level at which the input will be active.

Each input in the drive can be mapped to a "built-in" function. This means that when the input is logically sensed, the function will be activated. The available built-in functions are described in the section Function Description.

The logic level defines the relation between the hardware connectivity of the digital input and the activation of this input.

Positive logic (*active high*) means that the input will be sensed if current flows through the input pin.

Negative logic (*active low*) means that the input will be sensed if no current flows though the input pin. This is normally used, for example, in abort functions, where the user would typically like to prevent motion when no current flows to the drive input.

The number of inputs and physical input logic depends on the drive hardware. Refer to the drive Installation guide (IG) for more details about the number of inputs and physical levels.

Note:
The actual number of digital inputs depends on the drive hardware. Typically, 6 inputs are available. The drive firmware will allow the digital input settings even if the drive's hardware is not available.

Ports A, B, and C can be used as DINs in the case when they are not used for sensor signals.

The following DINs are connected to ports:

Platinum Drive and Titanium Single Axis Drive

Port A_I	Port A_B	Port A_A
DIN11	DIN10	DIN9

Port B_I	Port B_A
DIN13	DIN12

Port C_I	Port C_B	Port C_A
DIN16	DIN15	DIN14

Platinum Quartet Drive

The Platinum Quartet drive includes 16 digital inputs distributed between 4 axes as presented in the following table. All inputs can be mapped to any of the built-in functions.

Axis	Digital Input	Related IL[] command
1	IN1	**Ax1.IL[1]**
	IN2	**Ax1.IL[2]**
	IN3	**Ax1.IL[3]**
	IN4	**Ax1.IL[4]**
2	IN1	**Ax2.IL[5]**
	IN2	**Ax2.IL[6]**
	IN3	**Ax2.IL[7]**
	IN4	**Ax2.IL[8]**
3	IN1	**Ax3.IL[9]**
	IN2	**Ax3.IL[10]**
	IN3	**Ax3.IL[11]**
	IN4	**Ax3.IL[12]**
4	IN1	**Ax4.IL[13]**
	IN2	**Ax4.IL[14]**
	IN3	**Ax4.IL[15]**
	IN4	**Ax4.IL[16]**

Titanium 2 Axes Drive

Some pins of ports A1, A2, B1, B2 can be used as DINs in the case when they are not used for sensor signals. The following DINs are connected to ports:

Axis 1 Port B_I	Axis 1 Port B_A	Axis 1 Port A_I	Axis 1 Port A_A
DIN12 IP bit 27	DIN11 IP bit 26	DIN10 IP bit 25	DIN9 IP bit 24

Axis 2 Port B_I	Axis 2 Port B_A	Axis 2 Port A_I	Axis 2 Port A_A
DIN116 IP bit 31	DIN15 IP bit 30	DIN14 IP bit 29	DIN13 IP bit 28

Sticky Bit

If the input is defined as sticky bit (bit 8 of the **relevant IL[]** is set to '1'), an active input will be latched. Writing '1' to a latched input (via **IP** or **IB** command) clears the indication. This is useful for cases when the host cycle time is slower than the input active time and the input might be reset back to ‘0’ before the host samples the level. Using the ‘sticky bit’ allows the host to reset the indication only after the indication was sampled.

Note for Extended Inputs:
In some drive hardware configurations there might be up to 32 extended inputs. These inputs are not influenced by the **IL[]** command.

Values for IL[N] (N: the number of input)

Command Value	Logic Level	When Active
IL[*N*] = 0	Low	Inhibit: Shut off the servo drive, freewheel. Note that when this function is required when the switch is not active the servo is automatically enabled.
IL[*N*] = 1	High	
IL[*N*] = 2	Low	Stop immediately under control: Hardware and auxiliary stops
IL[*N*] = 3	High	Stop immediately under control: Hardware and auxiliary stops
IL[*N*] = 4	Low	No function is attached. Ignore the switch
IL[*N*] =5	High	No function is attached. Ignore the switch
IL[*N*] = 6	Low	General-purpose
IL[*N*] = 7	High	General-purpose
IL[*N*] = 8	Low	Reverse limit switch (RLS). Only forward motion is allowed
IL[*N*] = 9	High	Reverse limit switch (RLS). Only forward motion is allowed
IL[*N*] = 10	Low	Forward limit switch (FLS). Reverse motion is allowed

Command Value	Logic Level	When Active
IL[*N*] = 11	High	Forward limit switch (FLS). Reverse motion is allowed
IL[*N*] = 12	Low	**Begin:** activates the **BG** command.
IL[*N*] = 13	High	**Begin:** activates the **BG** command.
IL[*N*] = 14	Low	**Stop immediately under control: Soft stop only.** **Activates the ST command.**
IL[*N*] = 15	High	**Stop immediately under control: Soft stop only.** **Activates the ST command.**
IL[*N*] = 16	Low	Main Home switch
IL[*N*] = 17	High	Main Home switch
IL[*N*] = 18	Low	Additional (auxiliary) Home switch
IL[*N*] = 19	High	Additional (auxiliary) Home switch
IL[*N*] = 20	Low	Stop immediately under control, HW and SW stops: stop both the software profiler and the auxiliary reference.
IL[*N*] = 21	High	Stop immediately under control, HW and SW stops: stop both the software profiler and the auxiliary reference
IL[*N*] = 22	Low	Abort motion. Shut off the servo drive, freewheel
IL[*N*] = 23	High	Abort motion. Shut off the servo drive, freewheel
IL[N] = 26	Low	Additional Abort motion. Shut off the servo drive, freewheel
IL[N] = 27	High	Additional Abort motion. Shut off the servo drive, freewheel
IL[*N*] = 28	Low	Set RM =1; Engage Follower and ECAM
IL[*N*] = 29	High	Set RM =1; Engage Follower and ECAM
IL[*N*] = 30	Low	Set RM =0; Disengage Follower and ECAM
IL[*N*] = 31	High	Set RM =0; Disengage Follower and ECAM
IL[*N*] = 262	Low	General-purpose sticky input
IL[*N*] = 263	High	General-purpose sticky input

In the Quartet drive Ports A1 – A4, B1, B3 function can be defined by **IL[]** command bits instead of digital inputs. Refer to **GB[]** command for detailed information.

Bit-Field Entries

The following table describes the bit-field entries for logic and function in **IL[]**.

Bit	Description	Type	Values		Restrictions
0	Logic level	Boolean	0	Active low	
			1	Active high	

Bit	Description	Type	Values		Restrictions
1 to 4	Function number and behaviors * See detailed description in the section Function Description below	Long	0	Shut off the servo drive, freewheel.	
			1	Stop immediately under control: Hardware and auxiliary stops.	
			2	No function is attached. Ignore the switch.	
			3	General-purpose.	
			4	Hard-enable forward direction only (RLS).	
			5	Hard-enable reverse direction only (FLS).	
			6	**Begin:** activates the **BG** command.	
			7	Stop immediately under control: soft stop.	
			8	Main Home switch	
			9	Additional (Auxiliary) Home switch	
			10	Stop immediately under control, HW and SW stops: Stop both the software trajectory and auxiliary reference.	
			11	Abort motion. Shut off the servo drive, freewheel.	
			12	Reserved for safety function compatibility.	
			13	**Additional Abort motion. Function can be used to shut off the motion, produces a different fault code.** (See **MF** command)	
			14	**Engage ECAM/Follower: set RM=1**	
			15	**Disengage ECAM/Follower: set RM=0**	
5 -7	Reserved				

Bit	Description	Type	Values		Restrictions
8	Sticky function	Boolean	0	Non sticky	
			1	Sticky	
9 to 15	Reserved				

Function Description

Function 0: Inhibit (freewheel)

When the inputs are active, the servo is off (**MO** = 0) and the motor is not under control. No current is applied through the motor phases. If the motor was previously running, it will continue to coast on its own inertia.

The motor fault code (see the **MF** command) is 0x10. When the input is “not active”, a motor restart will be attempted. This attempt is made within a few milliseconds. In addition, when the motor is restarted, the #@AUTO_ENA automatic routine, if declared in a User Program, will be activated.

Function 1: Hard stop immediately under control

The drive will stop all auxiliary motion (e.g., ±10 V analog reference, follower, ECAM etc.) in the fastest possible way. If **UM** = 1 (current control), the torque command is set to 0 immediately. In any other unit modes (velocity and position) the drive will stop using the **SD** command.

The #@AUTO_STOP automatic routine, if declared in a User Program, will be activated.

When this digital input is changed to its non-active state, the Hard Stop situation is terminated.

Note that in Time dependent modes (CSP and IP modes) when the function is active, the motion stops regardless of the set-point from the host (**0x607A** or **0x60C1**). The host should return the set point to the actual position of the drive prior to releasing the Hard Stop switch. If the host does not return the set point to the actual position of the drive, and the switch is released, the drive motion will jump to set point.

Typically, the host should regulate the set point to stop immediately and return to the actual position of the drive. A release of the switch in this mode will immediately continue the motion, where the first position after the release is the present set-point. If when releasing the switch, the actual position of the drive and the set point from the host do not match, a jump occurs. By default, the Hard Stop event is ignored by the drive in these motion modes. This can be modified by clearing of bit 2 in **XA[4]** command.

Function 2: Input is ignored

The input is ignored. No function is applied in the drive and **IP**/**IB[*N*]** indications always read zero.

Function 3: General purpose

The purpose of this function is to allow the user general use of the input. The relevant input entry will be signaled in the **IP** or **IB[*N*]** command. With the use of the User Program, the user can perform any desired action, for example, signaling an output.

If an #@AUTO_IN routine is declared in the User Program, the routine will be automatically called. For example, if digital input 3 is declared as general-purpose with active low logic, #@AUTO_IN3 will be called if no current flows though input 3 pin.

Function 4: Reverse limit switch (RLS): forward only

When this function is active, reverse motion is not available, and any reverse command will be discarded by the Stop Manager.

The word "reverse" refers to when the current and the velocity commands have negative values.

If the motion is in the reverse direction during activation of the function, the motion will be stopped according to the following:

- If **UM** = 1 or when torque motion is applied (via **Object 0x6060**) (current control), the torque command is set to 0 immediately.
- If **UM** = 2 or **UM** = 5 (velocity and position modes) or when the profiled Velocity or Profiled Position, the drive will stop using the **SD** command.

The limit is applied to all reference including auxiliary reference (e.g. follower) Note that in auxiliary reference when the switch is released, the command is recovered and the motion resumes. Same happens in Profile Velocity mode (0x6061 is 3).

Notes:
By default, in Time dependent modes (Interpolated position, CSP) where the motion is controlled by the host, the event of RLS is ignored by the drive. The drive will respond to RLS event if the Bit 2 of the **XA[4]** command is set to 0.
When the RLS is active (bit 2 of **XA[4]** is set to 0), the position of the RLS event is captured. In these modes, releasing the switch will not resume motion. The motion is resumed only after the trajectory returns to the captured position. In these modes, if both FLS & RLS are set at the same time, the motion will be set to off automatically with appropriate fault indication (refer to **MF=16**).

If an #@AUTO_RLS routine is declared in the User Program, the routine will be called automatically.

Function 5: Forward limit switch (FLS)
When this function is active, forward motion is not available and any forward command will be discarded by the Stop Manager.

The word "forward" refers to when the current and the velocity commands have positive values.

If the motion is in the forward direction during activation of the function, the motion will be stopped according to the following:

- If **UM** = 1 (current control), the torque command is set to 0 immediately.
- If **UM** = 2 or **UM** = 5 (velocity and position modes), the drive will stop using the **SD** command.

If an #@AUTO_FLS routine is declared in the User Program, the routine will be called automatically.

This function does not change the drive's reference command. When the switch is released, the reference command (speed or position) is recovered. Note that in Time dependant modes (Interpolated position, CSP) the event of FLS is captured. In these modes releasing the switch will not resume motion. The motion is resumed only after the trajectory returns to the captured position.

Function 6: Begin
This function behaves like a software **BG** command, i.e., it starts the programmed motion. See the **BG** command for more details.

If an #@AUTO_BG routine is declared in the User Program, the routine will be called automatically.

Function 7: Software Stop
This function stops the motor under control, stopping the software reference similar to **ST** command.

Function 8: Main Home switch

When this function is active, the inputs are used as a Homing Switch in the Homing/Capture process. For this purpose, in addition to the configuration of a digital input as Main Home Switch, the **GI[*N*]** command should be set (simpler in EASII) in order to connect the Home Switch digital input to the Drive Homing Module (refer to the **GI[*N*]** command).

This function activates the #@AUTO_HM routine in the user program, depending on its declaration.

Function 9: Additional Home switch

When this function is active, the inputs are used as an Additional Homing Switch in the Homing/Capture process. For this purpose, in addition to the configuration of a digital input as Additional Home Switch, the **GI[*N*]** command should be set (simpler in EASII) in order to connect the Additional Home Switch digital input to the Drive Homing Module (refer to the **GI[*N*]** command).

This function activates the #@AUTO_HM routine in the user program, depending on its declaration.

Function 10: Hard and Soft stops

This function stops the motor under control, stopping the auxiliary reference and software reference.

This function activates the #@AUTO_STOP routine in the user program.

See Function 1: Hard stop immediately under control for further details.

Function 11: Abort motion

The behavior is similar to the Inhibit function with the exception that the “Abort” input release does not start the motor automatically. After the Abort is activated, **MO** = 1 must be set either by communication or by the internal User Program.

The function activates the #@AUTO_ER routine, if it exists, in the user program.

Function 12: Additional Abort motion

The function behaves similar to the Abort Motion. It allows the user to have a second freewheeling function and to distinguish between the source of the Abort. This can be used for e.g. PTC function. The Additional Abort reports a different **MF** (Motor Fault) value from the Abort motion.

Function 14: Engage ECAM/Follower

This function enables the ECAM/Follower functionality of the external reference generator (**RM=1**). The function does not execute the **EI** command.

Function 15: Disengage ECAM/Follower

This function disables the ECAM/Follower functionality of the external reference generator (**RM=0**).

References

IP, **IB[]**, **HM[]**, **MF, XI[], GB[]**

IP – Input Port

IP reports the status of the digital inputs. The command is axis related

CANopen/CoE

Digital_inputs:0x60FD Elmo Alias Object: 0x3139, 0x3839, 0x4139, 0x4839, 0x5139, 0x5839

Attributes

Attribute	Description	
Type	Bit Field, Read-Write	
Source	All	
Restrictions	NA	
Range	Bit 0	General-purpose input is active
	Bit 1	Safety (OK)
	Bit 2	Main home switch
	Bit 3	Auxiliary home switch
	Bit 4	Soft stop
	Bit 5	Hard stop
	Bit 6	Forward limit switch (FLS)
	Bit 7	Reverse limit switch (RLS)
	Bit 8	Inhibit (enable) switch
	Bit 9	Hardware motion begin (**BG**)
	Bit 10	Abort function
	Bit 11	Additional abort
	Bit 12	Set **RM =1**; Engage Follower and ECAM
	Bit 13	Set **RM =0**; Disengage Follower and ECAM
	Bits 14 to 15	Not used. Always zero
	Bit 16	Digital input 1 logical pin state **For Quartet drive**: Axis 1 Digital input 1 logical pin state
	Bit 17	Digital input 2 logical pin state **For Quartet drive**: Axis 1 Digital input 2 logical pin state
	Bit 18	Digital input 3 logical pin state **For Quartet drive**: Axis 1 Digital input 3 logical pin state

Attribute	Description	
	Bit 19	Digital input 4 logical pin state **For Quartet drive**: Axis 1 Digital input 4 logical pin state
	Bit 20	Digital input 5 logical pin state **For Quartet drive**: Axis 2 Digital input 1 logical pin state
	Bit 21	Digital input 6 logical pin state **For Quartet drive**: Axis 2 Digital input 2 logical pin state
	Bits 22	Digital input 7 logical pin state **For Quartet drive**: Axis 2 Digital input 3 logical pin state
	Bits 23	Digital input 8 logical pin state **For Quartet drive**: Axis 2 Digital input 3 logical pin state
	Bits 24	Digital input 9 logical pin state. Port A, A encoder entry **For Quartet drive**: Axis 3 Digital input 1 logical pin state **For Titanium drive**: Digital input 9 logical pin state. Axis 1 Port A, A encoder entry
	Bits 25	Digital input 10 logical pin state. Port A, B encoder entry **For Quartet drive**: Axis 3 Digital input 2 logical pin state **For Titanium drive**: Digital input 10 logical pin state. Axis 1 Port A, Index encoder entry
	Bit 26	Digital input 11 logical pin state. Port A, INDEX encoder entry **For Quartet drive**: Axis 3 Digital input 3 logical pin state **For Titanium drive**: Digital input 11 logical pin state. Axis 1 Port B, A encoder entry
	Bit 27	Digital input 12 logical pin state. Port B, A encoder entry **For Quartet drive**: Axis 3 Digital input 4 logical pin state

Attribute	Description	
		For Titanium drive: Digital input 12 logical pin state. Axis 1 Port B, Index encoder entry
	Bit 28	Digital input 13 logical pin state. Port B, INDEX encoder entry **For Quartet drive**: Axis 4 Digital input 1 logical pin state **For Titanium drive**: Digital input 13 logical pin state. Axis 2 Port A, A encoder entry
	Bit 29	Digital input 14 logical pin state. Port C, A encoder entry **For Quartet drive**: Axis 4 Digital input 2 logical pin state **For Titanium drive**: Digital input 14 logical pin state. Axis 2 Port A, Index encoder entry
	Bit 30	Digital input 15 logical pin state. Port C, B encoder entry **For Quartet drive**: Axis 4 Digital input 3 logical pin state **For Titanium drive**: Digital input 15 logical pin state. Axis 2 Port B, A encoder entry
	Bit 31	Digital input 16 logical pin state. Port C, INDEX encoder entry **For Quartet drive**: Axis 4 Digital input 4 logical pin state **For Titanium drive**: Digital input 16 logical pin state. Axis 2 Port B, Index encoder entry
Index range	NA	
Default	NA	
Unit modes	All	
Non- Volatile	No	
Axis Related	Yes	

Remarks

The **IP** command reports the logic state and the activated function of the whole digital input port.

The command is divided in to two sections of 16 bits each:

- Bits 0 to 15 report the actual function which is active e.g. Reverse Limit Switch, Homing etc.
- Bits 16 to 31 report the logic level of the input where 1 means that the input is logically active regardless of the physical state. Writing "1" to bits 16 to 31 clears sticky bit (latched input), so command **IP**=IP will clear only the sticky bits, and other bits which are not sticky do not change. Writing "0" is not allowed. For more information about "sticky bits" refer to the **IL[]** command.

For example:
If digital input 2 is configured as Forward Limit (FLS), input 4 is configured as Main Home Switch and both inputs become logically active, bits 2, 6, 17 and 19 will be set to 1. In this case the **IP** command returns 655428 or 0x000A0044.

In the Quartet drive Ports A1 – A4, B1, B3 can be presented in IP command bits instead of digital inputs. Refer to GB[] command for detailed information.

	Note: Digital inputs 9 to 16 are read from the differential encoder entries of Port A , Port B, and Port C. In order to prevent the triggering of these signals, set between **IL[9]** to **IL[16]**, to "ignore" (**IL[N]** = 4 or 5).

References

IB[*N*], **IL[*N*], GB[]**

JP[N] – Jog Velocity in Position mode

JP[N] specifies the motor speed reference for jogging in position control mode.

CANopen/CoE

JP[1]: Profile_velocity 0x6081 End_velocity: 0x6082**Emo Alias Object**: Not Available

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	• Motor must be on (except for **JP[4]**) • Effective on the next call to **BG[N]**
Range	$-10^{11}.....10^{11}$
Index range	1 to 4
Default	0
Unit modes	**UM = 2, 5, 6, 7**
Non-Volatile	No
Axis Related	Yes

Indices

Refer to PA[N] command for more details.

Remarks

On the next **BG[1]** after applying the **JP[1]** command:

- The motion control will be switched to the position control loop.
- The motor will jog at **JP[1]** speed according to **AC[1], DC** and **SF**.

On the next **BG[N]** after applying **JP[N]** command (**N=2,3,4**) the corresponding profile will jog according to **AC[N]** and **JP[N]**.

When **JP[1]** is set and **BG[1]** is commanded, the motion mode (object 0x6061) is modified to 1 (Profile Position).

Objects 0x6081, 0x6082 will be overridden by the **JP[1]** value. Refer to the **BG** command for more details.

JP[N] jog in Position mode is not an endless motion like **JV** jog, therefore it will stop on the software position limits (**VH[3]**, **VL[3]**).

JP[1] can be set higher than **VH[2]** (the velocity limit). In this case, the actual Speed command will be saturated by **VH[2]**.

The motor will abort if the feedback speed is higher than **HL[2]**.

The **JP[N]** value can be given in user-defined units specified by the **FC** command.

References

PA, **SP**, **AC**, **DC**, **UM**, **JV**, **FC**

JV – Jog Velocity

JV specifies the motor speed reference for jogging in velocity control mode.

CANopen/CoE

Target_velocity: 0x60FF**Emo Alias Object**: Not Available

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	• Motor must be on • Effective on the next call to **BG**
Range	$-10^{11}.....10^{11}$
Index range	1
Default	0
Unit modes	**UM = 2, 5**
Non-Volatile	No
Axis Related	Yes

Remarks

On the next **BG** after applying **JV** command:

- the motion control will be switched to the velocity control loop
- the motor will jog at the speed specified by **JV** according to **AC**, **DC** and **SF**

When **JV** is set and **BG** is commanded, the motion mode (object 0x6061) is modified to 3 (Profile Velocity).

Object 0x60FF will be overridden by the **JV** value. Refer to the **BG** command for more details.

Jog is an endless motion, which does not halt at any of the software limits (**VH[3]**, **VL[3]**) or the software range modulo (**XM[1]**, **XM[2]**).

JV can be higher than **VH[2]** (the velocity limit). In this case, the actual Speed command will be saturated by **VH[2]**.

The motor will abort if the feedback speed is higher than **HL[2]**.

The **JV** value can be given in user-defined units specified by the **FC** command.

References

PA, **SP**, **AC**, **DC**, **UM**

KA[] –Scheduled Velocity Advanced Filter #1

KA[] specifies the parameters of the gain scheduled velocity advanced filter #1.

CANopen/CoE

Elmo Alias Object: 0x3172

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	**Refer to the KV command for the specific dependent ranges**
Index range	1 to 252
Default	0
Unit modes	**UM=2, 5, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

Velocity advanced filter #1 is configured by **KA[1…252]** and activated by **KV[25]**, and gain schedule mode is activated for it by **GS[16]**.

Indices

The following table details the use of the **KA[]** parameters array:

Index	KG[N] Value	Units	Length
1 to 63	Parameter 1 for scheduled velocity advanced filter #1	**By filter type KV[25] and gain schedule method GS[16]**	63
64 to 126	Parameter 2 for scheduled velocity advanced filter #1		63
127 to 189	Parameter 3 for scheduled velocity advanced filter #1		63
190 to 252	Parameter 4 for scheduled velocity advanced filter #1		63

References

GS[], **KV[]**

KB[] –Scheduled Velocity Advanced Filter #2

KB[] specifies the parameters of the gain scheduled velocity advanced filter #2.

CANopen/CoE

Elmo Alias Object: 0x3173

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	Refer to the **KV** command for the specific dependent ranges
Index range	1 to 252
Default	0
Unit modes	**UM=2, 5, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

Velocity advanced filter #1 is configured by **KB[1…252]** and activated by **KV[30]**, and gain schedule mode is activated for it by **GS[17]**.

Indices

The following table details the use of the **KB[]** parameters array:

Index	KG[N] Value	Units	Length
1 to 63	Parameter 1 for scheduled velocity advanced filter #2	**By filter type KV[30] and gain schedule method GS[17]**	63
64 to 126	Parameter 2 for scheduled velocity advanced filter #2		63
127 to 189	Parameter 3 for scheduled velocity advanced filter #2		63
190 to 252	Parameter 4 for scheduled velocity advanced filter #2		63

References

GS[], **KV[]**

KC[] –Scheduled Position Advanced Filter

KC[] specifies the parameters of the gain scheduled position advanced filter.

CANopen/CoE

Elmo Alias Object: 0x3174

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	Refer to the **KV** command for the specific dependent ranges
Index range	1 to 252
Default	0
Unit modes	**UM= 5, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

Position advanced filter is configured by **KC[1…252]** and activated by **KV[50]**, and gain schedule mode is activated for it by **GS[18]**.

Indices

The following table details the use of the **KC[]** parameters array:

Index	KG[N] Value	Units	Length
1 to 63	Parameter 1 for scheduled position advanced filter	**By filter type KV[45] and gain schedule method GS[18]**	63
64 to 126	Parameter 2 for scheduled position advanced filter		63
127 to 189	Parameter 3 for scheduled position advanced filter		63
190 to 252	Parameter 4 for scheduled position advanced filter		63

References

GS[], **KV[]**

KG[] – Gain Scheduled Controller

KG[] specifies the parameters of the gain scheduled speed and position controllers

CANopen/CoE

Elmo Alias Object: 0x3178

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	Refer to the commands **KI** and **KP** for the specific dependent ranges
Index range	1 to 189
Default	0
Unit modes	**UM=2, 5, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

The **KG[]** parameters apply only if the controller gains scheduling is enabled (see the **GS[2]** command).

Indices

The following table details the use of the **KG[]** parameters array:

Index	Description	Unit Modes	Units
1 to 63	**KI for velocity loop**	**UM=2, 5**	Hz
		UM=7	TBD
64 to 126	**KP for velocity loop**	**UM=2, 5**	ampere/(counts/sec)
		UM=7	TBD
127 to 189	**KP for position loop**	**UM=5, 7**	rad/sec

References

GS[], **KP[]**,**KI[]**

KI[], KP[] – PI Controllers

KI[] and **KP[]** define the parameters of the PI controllers without the second-order filters.

CANopen/CoE

KI - Elmo Alias Object: 0x317A

KP - Elmo Alias Object: 0x3181

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	KI[*N*] > 0 **KP[7] > 0**
Index range	KP[1 to 5, 7], KI[1, 2, 4]
Default	See below
Unit modes	See below
Non-Volatile	Yes
Axis Related	Yes

Remarks

KI[1] and **KP[1]** define the PI current control filter. The units of **KP[1]** are volt/ampere.

KI[2] and **KP[2]** define the PI velocity control filter. The units of **KP**[2] are ampere/(counts/sec)

KP[3] defines the gain of the position controller. The units of **KP[3]** are rad/sec.

KI[4] and **KP[4]** for PI gantry velocity control filter. The units of **KP[4]** are ampere/(counts/sec), and for stepper closed loop PI, the units are counts/(counts/sec).

KP[5] defines the gain of the gantry position controller. The units of **KP[5]** are rad/sec.

The parameters **KP[2]**, **KI[2]** and **KP[3]** apply only if gain scheduling is not used: **GS[2]** = 0.

Indices

The following table describes the **KI[]** and **KP[]** entries.

Index	Description	Default	Unit Modes	Units
1	Defines the PI current controller	0	All	**KP: volt/ampere** **KI: Hz**
2	Defines the PI velocity controller (when **GS[2]** = 0)	0	**UM=2, 5**	**KP: ampere/(counts/sec)** **KI: Hz**

Index	Description	Default	Unit Modes	Units
			UM=7	**KP: TBD** **KI: TBD**
3	Defines the gain of the position controller (when **GS[2]** = 0)		**UM=5, 7**	**KP: rad/sec**
4	Defines the PI gantry velocity controller		Position and current with gantry	**KP: ampere/(counts/sec)** **KI: Hz**
5	Defines the gain of the gantry position controller		Position and current with gantry	**KP: rad/sec**
6	Non-linear factor for Yaw controller. This value limits the Yaw controller output to a specific acceleration to close the position error. abs(KP[5]*Error) <= sqrt(2*KP[6]*abs(Error)).	10^{14}	Position and current with gantry	**KP: counts/sec^2**
7	Stepper closed loop speed controller conversion factor	1		**KP: NA**

References

KV[], **GS[*N*]**, **KG[]**

KL[N] – Kill user program

KL[*N*] stops execution of the user program and turns the servo off.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Command
Source	All, except the user program
Restrictions	None
Range	None
Index range	2
Default	none
Unit modes	All
Non-Volatile	No
Axis Related	No

ndices

The following table details the **KL[]** entries.

Index	Description	Notes
1	Disable the servo via internal **MO**=0 Disable the user program	The **AF[1]** is set to 0 (Local control). If User program exists the status via **PS** command is -1.
2	Disable the user program	Only User Program execution is terminated. Servo state w/o change.

Remarks

The program can run again from the start using the XQ command.

The **KL** command issued when no program is running does nothing and sets no error code.

KL differs from **HP**, which halts the program and allows it to resume from the same point.

After **KL[1]** (or **KL**) the **AF[1]** is set to 0 and the Control word is handled only locally by the drive. This will be indicated via the DS402 Status word (Object 0x6041). To resume to Remote control set **AF[1]=**1.

References

HP, XQ, XC, PS, AF

KR – Kill Motion Repetitive

KR command stops the ongoing special motion.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	None
Source	All
Restrictions	Profiled position mode (PTP) only
Range	None
Index range (used in vectored commands)	None
Default	0
Unit modes	**UM = 5**
Non-Volatile	No
Axis Related	Yes
Attribute	None

Remarks

The **KR** command stops the special motion mode (i.e. repetitive motion), after the current motion is completed.

This command does not stop the ongoing motion.

Where the buffer mode (**MR[1]**=4) or blended mode (**MR[1]**=5) are used, the motion stops when the last set point is completed, regardless of the command **KR**.

Special motion mode includes the repetitive modes. Refer to the **MR[]** command for more details.

Note:
If the special motion mode is not enabled, this command is ignored.

At the next **BG** command, the special mode is re-evaluated.

The **ST** command stops the ongoing motion and the repetitive mode (set **MR[1]**=0) immediately. The next **BG** does not initiate the repetitive motion i.e. **MR[1]** should be set again.

References

UM, **SR**, **MR[*N*]**, **ST**

KV[] – High-Order Controller Filter Parameters

KV[] specifies the parameters of the second-order 2x2 (advanced) filters. This filter has DC gain (gain is equal to one in zero frequency).

Indexes equal to 5n specify the filter type according to Table 23 below. When changing filter type drive check the validity of filter parameters. This parameter must be changed at motor disabled.

CANopen/CoE

Elmo Alias Object: 0x3187

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	No
Range	See the tables below
Index range	1 to 100
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

Each filter has five parameters. The first four parameters are the specific parameters of the filter, and the fifth parameter is the filter type. The filter is enabled after the fifth parameter **KV**[5**n*] is set.

The following table describes the parameters.

Filter Location	Filters	Parameters	Gain Schedule Parameters
High-order speed controller filters (which filter the PI output)	Filter #1	KV[1] to KV[5]	
	Filter #2	KV[6] to KV[10]	
	Filter #3	KV[11] to KV[15]	
	Filter #4	KV[16] to KV[20]	
	Gain schedule filter #1	**KV**[25]	**KA[1…252], GS[16]**
	Gain schedule filter #2	**KV**[30]	**KB[1…252], GS[17]**
	Filter #1	**KV**[31] to **KV**[35]	

Filter Location	Filters	Parameters	Gain Schedule Parameters
High-order position controller filters (which filter the proportional output)	Filter #2	**KV**[36] to **KV**[40]	
	Gain schedule filter	**KV**[45]	**KC[1…252], GS[18]**
Reserved		**KV**[46] to **KV**[50]	
Velocity feedback	Filter #1	**KV**[51] to **KV**[55]	
	Filter #2	**KV**[56] to **KV**[60]	
Velocity External Reference	Filter #1	**KV**[61] to **KV**[65]	
Acceleration External Reference	Filter #1	**KV**[66] to **KV**[70]	
Reserved		**KV**[71] to **KV**[75]	
Reserved		**KV**[76] to **KV**[80]	
Gantry	Filter #1	**KV**[81] to **KV**[85]	
	Filter #2	**KV**[86] to **KV**[90]	
Velocity presentation	Filter #1	**KV**[91] to **KV**[95]	
Reserved		**KV**[21…24] **KV**[26…29] **KV**[41…44]	

Table 23 All advanced filters in drive

The following table describes the parameter options for each filter and the indices. There are five parameters for each filter. For a specific filter *n* (*n*=1 to 19), there are four filter parameters (5n-4, 5n-3, 5n-2, 5n-1), except for the filters that run in gain schedule, where P1 to P4 parameters are in **KA[*N*]/KB[N]/KC[N]** parameters.

Filter Type Value P5*n	Filter Type	P1 = 5*n – 4	P2 = 5*n – 3	P3 = 5*n – 2	P4 =5*n – 1
0	Filter is canceled				
1	Second-order low pass	Frequency [Hz] Range: 1 to 0.8 Nyquist freq.	Damping Range: 0.3 to 0.95		
2	First-order lead/lag	Frequency [Hz] Range: 0.2 to 0.2 Nyquist freq.	Phase [deg] Range: -60 to 60		
3	Second-order lead/lag	Frequency [Hz] Range: 1 to 0.3 Nyquist freq.	Phase [deg] Range: -90 to 90		

Filter Type Value P5*n	Filter Type	P1 = 5*n – 4	P2 = 5*n – 3	P3 = 5*n – 2	P4 =5*n – 1
4	Notch filter	Frequency [Hz] Range: 1 to 0.8 Nyquist freq.	Quality factor Range: 0.1 to 10	Attenuation [dB] Range: 0 to 40	
5	Anti-Notch	Frequency [Hz] Range: 1 to 0.8 Nyquist freq.	Quality factor Range: 0.1 to 10	Amplification [dB] Range: 0 to 40	
6	General Bi-Quad	Numerator frequency [Hz] Range: 1 to 0.8 Nyquist freq.	Numerator damping Range: 0.005 to 5	Denominator frequency [Hz] Range: 1 to 0.8 Nyquist freq.	Denominator damping Range: 0.005 to 5

Notes

- To obtain the lead filter, in the lead/lag filter, the phase should be positive.
- When setting gain schedule filter using the commands **KV[25]**, **KV[30]**, and **KV[45],** all the filters in table are set and checked, but output an error only according to the relevant indexes in **GS[16]**, **GS[17]**, and **GS[18]**, so it is better to first set **GS** and then **KV**

References

KG[],GS[]

LC – Current Limit Flag

LC reports the status of the current limiting process.

CANopen/CoE

Elmo Alias Object: **0x3198**

Attributes

Attribute	Description
Type	Unsigned char, Read-only
Source	All
Restrictions	None
Range	see "Remarks" table
Index range	1
Default	**NA**
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

Two different current limits are in use. The peak limit PL[1] specifies how much current can be applied to the motor during short time periods (PL[2]) and the continuous limit CL[1] specifies how much current can be applied to the motor continuously. The value of the command also reflected in Object 0x3619.

To protect the drive the following condition should be met:

$$(MC^2[2] - MC^2[4]) \cdot MC[3] \geq (PL^2[1] - CL^2[1]) \cdot PL[2]$$

Remarks

Bit	Description
0	*Current High* threshold exceeded, see notes
1	*Temperature threshold peak current* limit PL[5] exceeded
2	*Temperature threshold continuous* current CL[5] exceeded
3	*Voltage threshold* for current derating PL[7] exceeded
4-7	0, reserved

Notes

- Current protection method (I2T or RMS current) is defined by **AB[8]** command.
- In the case of RMS current protection method, *Current High* threshold is **CL[1]**, but not greater than Max Continuous current **MC[4].**
- In the case of I2T protection method, *Current High* threshold is calculated in accordance with this method.

References

MC, **PL[N]**, **CL[N], AB[N]**

LD[] – Load Parameters

LD command Loads parameter's data from FLASH memory to the RAM (from previous saved data).

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Not relevant
Source	All
Restrictions	All servo must be off
Range	According to array index
Index range	1
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

This procedure copies the parameter's image in FLASH and inserts their values into the parameters in RAM. (the values from previous save).

LD is automatically called during boot up procedure.

If the parameters image in FLASH is different to the parameters image in the firmware (for example, after adding new parameters) then perform the **"load parameters" process:**

1. First perform **RS** in order to set the default values into all parameters.

2. Then perform **LD** in order to set the FLASH values of the old parameters (from previous save).
If an error occurs during the Reset process, the returned error is : 'BAD_DATABASE', and the detailed error will contain the following: object index, sub-index, Error Code of the particular error, is retrieved via **CD[1]** command.

References

RS, **SV**, **CD**

MC[] – Maximum Current

MC[N] reports the drive DI settings related to the maximum phase current (peak current). This command informs the software about the rate of the servo drive used with the controller.

The **MC[N]** values are burned in during the production of the drive and cannot be modified by the user.

CANopen/CoE

Elmo Alias Object: 0x31BC

Attributes

Attribute	Description
Type	Float, Read-only
Source	All
Restrictions	No
Range	NA
Index range	NA
Default	According to the servo drive. It cannot be changed
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Indices

The following table describes the **MC[]** entries.

Index	Description	Units	Notes
1	Maximal current	Ampere	The maximum phase current (peak current) allowed for the drive. The drive can run at this current during 3 seconds
2	**Maximal PL[1]**	Ampere	**Maximum possible value for the motor peak current PL[1]**
3	**Maximal PL[2]**	Sec	**Maximum possible value for the motor peak current duration PL[2]**
4	**Maximal CL[1]**	Ampere	**Maximum possible value for the motor continuous current CL[1]**
5	Current sensor factor	Bits/Ampere	**Number of A2D bits corresponding to the MC[1]**

Remarks

The current can be limited with the **PL[1]** and **CL[1]** commands.

References

CL[], PL[]

MF – Drive Fault

MF latches and report the reason that caused the motor to be disabled (**MO** = 0).

CANopen/CoE

Elmo Alias Object: 0x31BF

MF reports the fault reason only if the motor was enabled while the fault occurred.

Attributes

Attribute	Description
Type	Bit field, Read-only
Source	All
Restrictions	None
Range	None
Index range	NA
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

An EMCY message that contains the fault is transmitted.

MF is automatically set to 0 on the next motor enable command from any source: an **MO** = 1 interpreter command, the DS-402 state machine or Inhibit \ Enable input.

The DS-402 state machine "Fault Reset" command does not clear the **MF** value, it switches the state machine to “Switch-On-Disable” state.

After the motor is shut down, the drive will delay the next enabling of the motor by 7.5 miliseconds.

Amplifier Faults are faults that are related to the power stage. If the fault is caused by an amplifier fault, the drive status LED will turn red, and the **AOK** function indicating the fault via digital output will be activated if required (see the **OL[]** command).

The AUTO_ER routine of the user program will be activated upon an **MF** event.

For the visual effect of the **MF** Value Status below refer to the section 3.4.14 Status LED Blinks in the MAN-P-ADMINGUIDE manual.

The following table details the bit-field structure with respect to the fault type (reason).

MF Value (Hex)	Fault type	EMCY code (Hex)	Description and resolution
1 (0x1)	Main feedback error	7300	Motor disabled immediately due to sensor error. Check **WS[43]** to get the sockets with errors (bit 1 for socket 1, etc.) To get the sensor latched error, check **W#[1]**, and **W#[3]** for current error (# stands for the socket number).
2 (0x2)	Commutation process fail	7382	Motor disabled immediately due to commutation process fail. Try running expert tuner commutation process again
4 (0x4)	Hall main feedback mismatch	7380	Motor disabled immediately due to illegal Hall combination detected during servo enable procedure for example: feedback loss, no match between encoder and Hall location. Try running EAS Expert Tuning "Commutation" procedure again or check both encoder and Halls connection
8 (0x8)	The peak current limit has been exceeded.	8311	Motor disabled immediately. The current has exceeded the value of **MC** but has not yet reached the level of a short. This is typically caused by instability of the current loop. Try running EAS Expert Tuning "Current" procedure again
16 (0x10)	External Inhibit was triggered (INH/ENB)	5441	Deceleration and servo off in accordance with fault reaction option code object 0x605E. The motor was disabled due to Inhibit or Abort input. The digital input function is defined by **IL[]**. The Inhibit function is also activated if both limit switches (FLS and RLS) are active simultaneously in Cyclic Synchronous Position motion mode. See **IL[]** for more details about the Inhibit/Abort functions. Note the Additional Abort function. Inhibit and/or Abort input should be deactivated

MF Value (Hex)	Fault type	EMCY code (Hex)	Description and resolution
32 (0x20)	AC fail: Loss of phase	3130	Motor disabled immediately due to loss of phase. Requires specific hardware configuration dependent on drive. Check that the drive's main AC phases are properly connected.
64 (0x40)	Hall sensor speed is too high or disconnected.	7381	Deceleration and servo off in accordance with fault reaction option code object 0x605E. Digital Hall run too fast or disconnected. Check Hall sensor connection or check motor speed. It cannot be more than 60 degree in servo cycle. Try to reduce TS in this case. Try to run commutation series of miscellaneous commands again
128 (0x80)	Speed tracking error	8480	Deceleration and servo off in accordance with fault reaction option code object 0x605E. Speed error exceeded the value defined in **ER[2]**. Speed error is defined as difference between the total commanded speed (including proportional gain) and the feedback in position mode and it is defined as the difference between commanded speed and the feedback in speed loop. This indication is not related to the Max Slippage Error as defined in the DS-402 Profile Velocity mode. Check that the value of **ER[2]** is fit to your sensor and profile. Try to run tuning again, consider using feed forward. Check that you are using optimal commutation Lower the acceleration or speed of your motion profile
256 (0x100)	Position tracking error	8611	Deceleration and servo off in accordance with fault reaction option code object 0x605E.

MF Value (Hex)	Fault type	EMCY code (Hex)	Description and resolution
			Position error exceeded the value defined in **ER[3]**. Position error is defined as difference between the total commanded position and the feedback position in position mode. Check that the value of **ER[3]** corresponds to the sensor resolution and the profile motion. Try to run tuning again. Check that you use optimal commutation. Lower the acceleration or speed of your motion profile.
1024 (0x400)	Gantry Yaw error limit exceeded	5280	Deceleration and servo off in accordance with fault reaction option code Object 0x605E . Gantry Yaw error exceeded the value of **ER[5]**. Check that the value of **ER[5]** corresponds to the gantry slave sensor resolution. Try running EAS Expert Tuning "Gantry Yaw". Lower the Acceleration or Speed of your motion profile.
2048 (0x800)	Communication failed, loss of synchronization or frame loss was detected.	8130	**For ECAT drives:** Motor is disabled due to Sync Manager watchdog event, loss of frame, loss of synchronization to the EtherCAT master or unexpected change state from OP to any state while the servo was enabled. Deceleration and servo off in accordance with fault reaction option code Object 0x605E. **For CANopen drives:** Motor is disabled due to a heartbeat violation, according to CANopen DS-301 **Object 0x1016**, due to bus off state or CANopen Master NMT STOP command. Drive reaction according to the Abort Connection Option code **Object 0x6007**
4096 (0x1000)	Under voltage (Amplifier Fault)	3120	Motor disabled immediately due to under voltage protection.

MF Value (Hex)	Fault type	EMCY code (Hex)	Description and resolution
			The measured bus voltage was lower than the minimum allowed value. The minimum allowed voltage is reported by **WS[25]** (nominal burnt-in) and **WS[26]** (actual threshold via **XP[5]**). The actual bus voltage is reported by **AN[6]**. Check bus voltage connections and bus voltage reading **AN[6]**. Check that voltage does not drop due to high acceleration.
8192 (0x2000)	Over voltage (Amplifier Fault)	3310	Motor disabled immediately due to overvoltage protection. The measured bus voltage was higher than the maximum allowed value. The maximum allowed voltage is reported by **WS[23]** (nominal burnt-in) and **WS[24]** (actual threshold via **XP[1]**). The actual bus voltage is reported by **AN[6]**. Check bus voltage connections and bus voltage reading **AN[6]**. Check that the voltage does not increase due to high deceleration.
16384 (0x4000)	Safe Torque Off (STO) switch are disabled (Safety state). (Amplifier Fault)	FF20	Motor disabled immediately due to Safe Torque Off. The drive is in safe state. STO state is reported via SR bits 25 and 26. Check that STO inputs are activated. Also check that there is no STO error reported via the EAS. **SR[3]** reports the error in case of Safety error.
32768 (0x8000)	Short protection (Amplifier Fault)	2340	Motor disabled immediately due to short protection: The current has exceeded a range which is considered as a phase-to-phase or phase-to-ground short. This instantaneous fault is measured by the hardware and typically cannot be recorded or indicated outside of the **MF** command. Check that the motor phases are properly connected and try running

MF Value (Hex)	Fault type	EMCY code (Hex)	Description and resolution
			EAS Expert Tuning "Current" and "Commutation".
65536 (0x10000)	Motor over temperature protection	4210	Motor disabled immediately due to motor over temperature protection: The measured motor temperature exceeded motor over temperature threshold. Relevant only if motor temperature sensor is used. The measured motor temperature is reported by **TI[3]** (Celsius) and **TI[4]** (Fahrenheit). The motor temperature threshold is reported by **WS[28]**.
131072 (0x20000)	Over speed protection	8481	Deceleration and servo off in accordance with fault reaction option code **Object 0x605E**. Motor is disabled due to over speed protection: the motor speed has exceeded the value of maximal allowed motor speed which is defined by **HL[2]** in user units. The motor main speed is reported by **VU**. To cancel this protection, set **HL[2]=0**.
262144 (0x40000)	SLT fault reaction	6180	Motor disabled immediately after Safe Limited Torque (SLT or SAT) fault reaction sequence finishes.
524288 (0x80000)	Vector axis fault	6181	Axis was immediately disabled due to a fault in one of the axes sharing the same vector. Check the **MF** of the other axes sharing the vector to know the reason for the fault.
2097152 (0x200000)	Motor is stuck	7121	Deceleration and servo off in accordance with fault reaction option code object 0x605E. Motor is disabled due to motor stack state. A stuck motor indication can be requested by using **CL[2], CL[3]** and **CL[4]** according to the following format: If the motor speed is lower than **CL[2]** (in counts/sec) and the measured

MF Value (Hex)	Fault type	EMCY code (Hex)	Description and resolution
			current is higher than **CL[3]** (in amperes), and if this is observed for more than **CL[4]** msec, the motor is considered to be in the "Motor Stuck" state. Check for physical obstructions. Check that the setting of the Motor Stuck protection fits the application friction.
4194304 (0x400000)	Feedback is out of position limits	8680	Deceleration and servo off in accordance with fault reaction option code object 0x605E. Motor is disabled due to position limit protection. The main position feedback exceeded the **HL[3]** or **LL[3]** limit. The main feedback is reported in **PU**. To cancel this protection, set **HL[3]=LL[3]=0**. These parameters are in user units. Adjust the position limits to the application requirements. Check that the axis was "homed" to the desired absolute position values.
8388608 (0x800000)	Kinematics error	FF34	Motor disabled immediately due to kinematics error
16777216 (0x1000000)	Gantry slave disabled	FF40	Deceleration and servo off in accordance with fault reaction option code object 0x605E. Gantry master disabled because Gantry slave is not enabled at current mode. Re-enable Gantry slave in current mode
33554432 (0x2000000)	Additional Abort input is active	5442	Deceleration and servo off in accordance with fault reaction option code object 0x605E. The drive sensed an input switch that is defined as Additional abort (refer to **IL[]**). The fault is similar to the "Abort" function with a different value report. This allows user to distinguish between two different fault states such as Inhibit and PTC.

MF Value (Hex)	Fault type	EMCY code (Hex)	Description and resolution
			External Abort input should be reset.
67108864 (0x4000000)	Drive over-temperature (Amplifier Fault)	4310	Motor disabled immediately due to drive over-temperature protection. The actual temperature is reported by **TI[1]** (Celsius) and **TI[2]** (Fahrenheit). The drive temperature threshold is reported by **WS[27]**. Try running EAS Expert Tuning "Current" and "Commutation". Improve the drive mounting for a thermal conductance, consider modifying the switching frequency via **XP[2]**.
134217728 (0x8000000)	Gantry master drive fault	FF35	Gantry slave motor disabled immediately due to failure in gantry master drive. Check **MF** command in gantry master drive
268435456 (0x10000000)	Attached slave drive fault (Gantry)	FF50	Motor disabled immediately due to failure in attached slave drive. Check **MF** command in attached slave drive
536870912 (0x20000000)	Failed to enable the motor	FF10	Cannot start motor because of internal problem. Check the reason in **EE[5]** command. In the case that **EE[5]** is 74, check the fault reason in **EE[2]**. In the Gantry drive check also status of **MF bit 28 (MF=0x10000000)**, "Attached slave drive fault"
1073741824 (0x40000000)	Motor disabled local	FF30	MO=0 command received from interpreter or user program. Deceleration and servo off in accordance with **AB[1]** command "Motor OFF option code"

References

SR

MI – Mask Interrupts

MI[N] Mask the execution of User Program

CANopen/CoE

Elmo_parameters_objects: 0x31C2

Attributes

Attribute	Description
Type	Unsigned Long, Read/Write
Source	All
Restrictions	Auto-routine AUTO_PERR is non-mask able. Thread 0 (main) is non-mask able.
Range	0…(2^{32}-1)
Default	0
Index Range	1, 2
Unit modes	All
Non-volatile	No
Attribute	No

Remarks

MI[1] - Mask the execution of User Program specified auto routine.

MI[2] - Mask the execution of User Program specified thread.

MI[1]: Mask auto routine

Indices

A user program may include a main code and some automatic routines.

When the program runs, the conditions for calling these routines are checked continuously. If the conditions for running an automatic routine are met, it is called. At certain times, you may want to block some of the automatic routines.

For example:

- - An AUTO_RLS automatic routine may be deactivated in a homing process.
- - Require that a certain code sequence to be un-interruptible.

Note:
MI[1] masks the execution but does not prevent it. The routine shall be executed after the MI[1] allows it.

The **MI[1]** bits are detailed in the following table, the priority of each routine is in the descending order:

MI[1] Value	Masked Interrupt	Relevant Routine
1 (0x1)	Not used	0
2 (0x2)	Abort	AUTO_ER
4 (0x4)	Soft stop	AUTO_STOP
8 (0x8)	Soft begin	AUTO_BG
16 (0x10)	RLS	AUTO_RLS
32 (0x20)	FLS	AUTO_FLS
64 (0x40)	Switch enable	AUTO_ENA
128 (0x80)	Digital input 1	AUTO_I1
256 (0x100)	Digital input 2	AUTO_I2

Notes

- The bit field characteristic of the **MI[1]** allows blocking of several auto routine in a single command.
- **MI[1]** prevents the call of the routine while the specific bit is set, blocked routine will be called when the specific bit in **MI[1]** was reset to 0.
- If AUTO_PERR is activated, all other interrupts are automatically masked (**MI[1]**=0x7fff).
- When auto routine is called the first executable line is performed under "critical section" allowing the user to set **MI[1]** in the same instance of the routine called.

MI[2]: Mask thread

MI[2] bits	Description	
0-3	Info about thread 0	
	0 (0x0)	Thread 0 is Unlock
	1 (0x1)	Thread 0 is Lock
4-7	Info about thread 1	
	0 (0x0)	Thread 1 is Unlock
	1 (0x1)	Thread 1 is Lock
8-11	Info about thread 2	
	0 (0x0)	Thread 2 is Unlock
	1 (0x1)	Thread 2 is Lock
12-15	Info about thread 3	
	0 (0x0)	Thread 3 is Unlock
	1 (0x1)	Thread 3 is Lock
16-19	Info about thread 4	

MI[2] bits	Description	
	0 (0x0)	Thread 4 is Unlock
	1 (0x1)	Thread 4 is Lock
20-23	Info about thread 5	
	0 (0x0)	Thread 5 is Unlock
	1 (0x1)	Thread 5 is Lock
34-31	Reserve	

Notes

- Main() – Thread 0 is always active – so ignore the info about thread 0.
- If thread is locked (set to 1) it will not be called and can't move to active status until set back to Unlock (set to 0).

References

XQ, KL[], CT[], HP[],XC[], PS[]

MO/SO – Motor On, Servo On

MO enables and disables the motor, and **SO** checks the servo state.

CANopen/CoE

MO - Elmo Alias Object: 0x31C8

SO - Elmo Alias Object: 0x32A0

DS-402 state machine using the Control Word (0x6040) and Status Word (0x6041) objects. Enabling the servo via DS-402 will be indicated by the **SO** command as well. Differences between DS-402 enabling and **MO**=1 enabling are described below.

Attributes

Attribute	Description
Type	Short
Source	All
Restrictions	None
Range	0,1
Index range	1
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Enabling the motor

MO = 1 is the operative state of the servo drive, driving the motor, activating and executing the required motion. The software procedure runs a set of tests to ensure that all conditions for running the motor are met. When the motor enabling procedure completes successfully, the **SO** is set to 1 indicating that the servo is enabled – next chapter elaborates on the **SO**. The enabling procedure may include a mechanical brake release and a commutation search that may increase the procedure duration. When the **SO** is set than bit 2 (Operation Enabled) of DS-402 status word is set as well. If **MO** is set to 1 and the motor is already On the procedure will not be executed again.

The status of the motor (**MO**) and the servo (**SO**) can be retrieve via **SR** command bits 22 and 4 respectively.

The status of the drive can be retrieved prior to enabling the motor via the **SR** command bits 0-3. If one of these bits are set than the **MO**=1 fails and the reason for the failure is reported via **MF** command.

The **MO** command can only be activated if the drive is in Local mode of the DS-402 state machine. Local\Remote mode can be alternated via **AF[1]** command.

When the motor is enabled, the drive reinitializes the internal parameters and motion drivers.

The drive may fail to start if during the servo enable procedure a parameter or functionality is in a state that does not permit the enabling of the servo. If such situation occurs than an error will be reported via **EC.** In some situations of failure during the servo enabling the **EC** will indicate to look for more details in **EE[5]** command.

Motor Fault indication (**MF**) is cleared during the motor enabling procedure. If the fault still persists than the **MF** will be reset. MF can be recorded via the "Exception" signal of the EAS recorder.

"Stay in place" procedure is activated during the servo enabling assures that the motor is always started so that it does not jump.

Note that between two consecutive motor enable and motor disable calls, the motor enabling will be delayed for few milliseconds (3-8 milliseconds).

The SO command On when the Motor is On

The Motor On request returns to the interpreter almost immediately. This, however, does not mean that the motor can be controlled by the application/profiler.

If the commutation was not found yet, the Motor On procedure will indicate that to the real time, where a commutation search procedure will take place. During this procedure, which might take a long time (a few hundred milliseconds), the profiler or auxiliary reference cannot command the motor to move. During this time the **SO** command will return zero.

The **SO** command indicates whether the servo is enabled, allowing the user (profiler) to command the motion, or is not yet enabled, preventing any reference command to be executed.

After the application initiates motor enable, it must continually check the **SO** command until the value is 1.

Another situation where the motion is prevented for a long time while the motor is enabled is when a brake is defined via **OL[].** The brake is released after time defined via **BP[]** command. In this case **SO** indicates 0 until the brake time is exhausted and the drive is ready for profiling.

Disabling the motor

MO = 0 disables the motor. This is the idle state of the drive. The power stage is disabled, and no current flows in the motor. The disabling procedure cannot fail and the servo will always be disabled – **SO** = 0. If the axis is used as a vector profiler and the vector is in motion (see **GR**, **EV**, **VS** commands) than the vector must be stopped before the axis vector can be disabled. If the drive is in DS-402 Remote mode (**AF[1]=1)** and the **MO**=0 is set than the mode will be switched to local mode (**AF[1]** will be automatically set to 0) and the servo will be disabled. Note that in this situation the **AF[1]**=1 must be set by the host to have the control of the motor enable procedure (and DS-402 control word).

If a brake is defined via **OL[]**, the **SO** is set to 0 only after the brake engage time elapsed (**BP[]**).

When setting the **MO**=0, the DS-402 state machine indicates Switch On Disable state.

When the servo is disabled, the drive performs various tasks that are impossible when the motor is on, including the following:

- Resetting the drive to default (RS command)
- Downloading new firmware or User Programs
- Saving or loading parameters in the flash memory

- Modifying setup data that cannot be modified on-the-fly, such as the commutation parameters (**CA[*N*]**) and unit mode (**UM**)

The servo drive is automatically disabled when a motor fault (**MF**) is captured. An attempt to enable the motor may fail if the conditions of the fault still exist.

The SO command On when the Motor is Off

SO is set immediately to 0 when the motor is off. **SO** remains set to 1 in cases in which a brake is applied. A brake is defined by **OL[]** which sets the relevant digital input and **BP[]** command that sets the brake time. In this case **MO** indicates 0 while **SO** indicates 1, informing the application that the servo is On.

Enabling the motor via DS-402 and via MO=1

Enabling the motor via DS-402 is done by the Control word (**Object 0x6040**). The control word is processed by the drive when the mode is Remote mode otherwise, it is ignored and adequate EMCY message will be transmitted (when control word is used via PDO). The remote/local mode is indicated via DS-402 status word bit 9.

In general, the same procedure of enabling and disabling the servo via DS-402 is performed via the **MO**=1 command. Actually, the drive imitates the exact state machine sequence that is executed when the DS-402 control word is used.

Remarks

- If **UM** = 3, an automatic torque can be applied while **MO** = 1 by setting **SC[8]**
- **MO** = 1 will be called automatically if one of the inputs is defined as Inhibit/Enable function (**IL[]** = 0 or 1)
- In auxiliary mode (**RM** = 1) the motor might move immediately with respect to the auxiliary reference, which might be an analog input
- Motor enable state can be detected via digital output, see **OL[]** command function 6,7

References

MF, **SR**, **CD**, **BP[]**, **EE[]**, **SO**, **OL[]**, **IL[]**

MP[] – Motor Parameters

MP[] specifies the motor's physical parameters

CANopen/CoE

Elmo Alias Object: 0x31C9

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	≥0
Index Range	1 to 7
Default	See the table below
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

Indices

The following table details the **MP[]** entries:

Index	Description	Default	Units	Notes
1	Motor resistance R_s	0.5	Ohm	
2	Motor self-inductance L_s	0.001	H or Webers/A	0.01 x 1Volt second/Amp
3	Back EMF constant K_e	0.01	V/el.Hz	Volt (peak) of one phase per electrical cycles/sec. Setting of K_e is required for correct Hot Plugging algorithm functionality. It is recommended to disable the Hot Plugging (**XA[2]**), if K_e is unknown.

Index	Description	Default	Units	Notes
4	**Number of motor pole pairs** **In rotary: pole pairs per revolution** **In linear: usually equal to one**	0		The motor must be off to change the setting. Changing the setting resets commutation. This parameter must be set, cannot operate motor with value zero
5	Motor equivalent inductance ratio $\frac{L}{L_s}$	1.5	-	L_s : self-inductance L_m : mutual inductance $L = L_s - L_m$: equivalent motor phase inductance
6	Mechanical inertia J	1	$Kg * m^2$	
7	Reserved	0		Volatile, any value is reset to 0 on the next power up cycle (or **LD** command)

Notes

Hot Plugging is the ability to enable the motor when it is moving. **XA[2]** defines the allowed electrical speed [Electrical cycles/second] for Hot Plugging. To increase this speed above the default value (5 Hz), it is recommended to set the back EMF constant K_e of the motor (**MP[3]**), so that the first voltage command is suitable for this speed.

References

MQ[] – Capture Mux Quad Counter Parameter

MQ[1] – MQ[6] selects the Quad Counter used as position signal for Capture module.

CANopen/CoE

Alias Object Ranges: **MQ[]** 0x31CA

Attributes

Attribute	Description
Type	Unsigned short, RW
Source	All
Restrictions	None
Range	In accordance with the Index
Index range	1 - 6
Default	In accordance with the Index
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The command is relevant for the Platinum Quartet drive. The command has no effect for Platinum standard drive. Each Index has possible Values of 0 to 8 described in the table following the Indexes

Indexes

Index	Quad Counter input signal	Default	Range
1	Axis 1 Quad Counter Mux to select the position input from A1 Quad A or A1 Quad B (reserved)	0	0
2	Axis 2 Quad Counter Mux to select the position input from A2 Quad A or A2 Quad B (reserved)	0	0
3	Axis 3 Quad Counter Mux to select the position input from A3 Quad A or A3 Quad B (reserved)	0	0
4	Axis 4 Quad Counter Mux to select the position input from A4 Quad A or A4 Quad B (reserved)	0	0

Index	Quad Counter input signal	Default	Range
5	Axis 5 Quad Counter Mux to select the position input from A1 or A2 Quad B	0	0, 2, 4
6	Axis 6 Quad Counter Mux to select the position input from A3 or A4 Quad B	0	0, 6, 8

Write access to reserved indexes has no effect and does not return an error.

Read access to reserved indexes returns zero.

Value Definition of each Index

Value	Description
0	The command has no effect, the mux is set in accordance with sensor initialization
1	A1 Quad A counter
2	A1 Quad B counter
3	A2 Quad A counter
4	A2 Quad B counter
5	A3 Quad A counter
6	A3 Quad B counter
7	A4 Quad A counter
8	A4 Quad B counter

References

CR[], GI[]

MR[N] – Motion Repetitive

MR[*N*] are parameters for the special motion mode. Special motion modes are enhancement modes to the point-to-point motion mode (**UM=5**).

CANopen/CoE

Elmo Alias Object: 0x31CB

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 4
Default	0
Unit modes	**UM = 5**
Non-Volatile	No
Axis Related	Yes

Remarks

The following options are supported special modes:

- Point-to-point repetitive motion. In this mode the point-to-point motion is repeated until the user stop command (e.g. **ST** command). Delay can be defined between motions.
- Point-to-point set of set points. In this mode every new motion is added to a buffer, and starts when the previous motion ends.
- Point-to-point set of set points blend mode. In this mode every new motion is added to a buffer, and is blended with the previous motion, i.e. the present motion decelerates or accelerates according to the next speed performing motion with a smooth blend between motions.

The motion begins by setting the **MR[]** command and initiating **BG** command. The following commands are applied to stop this mode:

- **KR** command stops the motion after the last segment is finished. The command does not stop the mode (**MR[1]** remains active)
- **ST** command stops the motion immediately and resets the mode (**MR[1]=0**)
- **KL** command disables the servo and resets the mode (**MR[1]=0**)

Indices

The following table describes the **MR[*N*]** entries.

<table>
<tr><th>Index</th><th>Description</th><th>Type</th><th>Default</th><th>Restrictions</th></tr>
<tr><td>1</td><td><table><tr><th>Value</th><th>Mode Description</th></tr><tr><td>0</td><td>Disable</td></tr><tr><td>1</td><td>Point-to-point repetitive mode, from present position (profiler) to target position. The target position can be either absolute (PA) or relative (PR) according to the last commanded values.</td></tr><tr><td>2</td><td>Point-to-Point repetitive mode, from MR[3] absolute position to MR[4] absolute position.</td></tr><tr><td>3</td><td>Point-to-Point repetitive mode, from MR[3] absolute position to MR[3]+MR[4] absolute position.</td></tr><tr><td>4</td><td>Point-to-point buffered mode</td></tr><tr><td>5</td><td>Point-to-point buffered blend mode</td></tr></table></td><td>Double</td><td>0</td><td>Profiled position mode (PTP)</td></tr>
<tr><td>2</td><td>Delay between motions (in addition to target time)
Applicable only if MR[1]==1/2/3</td><td>Double</td><td>0</td><td>0-16777215 [mSec]</td></tr>
<tr><td>3</td><td>First position, depend on MR[1] value
Applicable only if MR[1]==2/3</td><td>Double</td><td>0</td><td></td></tr>
<tr><td>4</td><td>Second position, depend on MR[1] value
Applicable only if MR[1]==2/3</td><td>Double</td><td>0</td><td></td></tr>
</table>

References

UM, **KR**, **SR**

MS[N] – Motion Status

MS[N] reports the status of the motion with respect to the profiler state and the actual feedback.

CANopen/CoE

Elmo Alias Object: 0x31CC

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	None
Range	0 to 3
Index range	1: Main profiler status 2: Superimposed profiler status 3: Phasing profiler status 4: Simple socket profiler status
Default	No
Unit modes	1, 2, 5, 6
Non-Volatile	No
Axis Related	Yes

Remarks

The motion status **MS[1]** (**MS**) refers to the state of the profiler according to the specific mode of operation and the auxiliary reference. Motion status refers to the profiler state rather than the physical movement. The profile can be active when there is no motion and vice versa, the profiler can be at rest while the motor is moving. Physical movement can be monitored via **SR** bit 23.

- Motion status is set to 3 to indicate that the motor is disabled and no profiler is active.
- Motion status is set to 2 when the profiler system is initiated.
- Motion status is set to 1 when the software command has reached the target, but feedback has not reached the target and is not within the target window boundaries.
- Motion status is set to 0 when the feedback reached the target and is within the target window boundaries. Refer to the explanations below on Time Dependent Motion Modes and External Reference.
- Motion status is depending in Operation Mode (**Object 0x6061**) and HW switch state (FLS, RLS, HW STOP switch)

The motion status values are mode-dependent according to the following description.

Indices

Refer to **PA[N]** command for more details.

Torque modes

- The CANopen Profile Torque mode
- Elmo's **TC** command

MS Value	Description	Note
0	1. Actual torque is within the torque window (**TR[5]**, **TR[6]**) when there is not active Halt or Quick Stop 2. Halt or Quick stop is active and actual speed is zero (within the defined zero window)	**TR[5]** and **TR[6]** are reflected in objects 0x2506, 0x32C7.5 and 0x2507, 0x32C7.6.
1	The torque command (**DV[10]**) has reached the torque target but actual torque is not within the torque window.	
2	1. The torque command differs from the torque target. 2. Motion stopped with HW switch	
3	The motor is disabled.	

Velocity modes

- The CANopen Profile Velocity mode
- Elmo's **JV** command

MS Value	Description	Note
0	1. The actual velocity has reached the target and is within the velocity window (**TR[3]**, **TR[4]**) when there is no active Halt, Quick Stop or HW switch. 2. Halt or Quick stop is active, no HW Switch sensed and velocity command is zero.	**TR[3]** and **TR[4]** are reflected in objects 0x606D, 0x32C7.3 and 0x606E, 0x32C7.4.
1	The velocity command (**DV[2]**) has reached the velocity target (the profiler is at rest) but actual velocity is not within the velocity window.	
2	1. The velocity command has not reached the velocity target. 2. Motion stopped with HW switch	
3	The motor is disabled.	

Position modes

- The CANopen Profile Position mode
- Elmo's **PA** and **PR** commands

MS Value	Description	Note
0	1. The actual position has reached the target within the position window (**TR[1]**, **TR[2]**) when there are no Quick Stop or Halt active	**TR[1]** and **TR[2]** are reflected in objects 0x6067, 0x32C7.1 and 0x6068, 0x32C7.2.

MS Value	Description	Note
	and Speed command and Reference Speed Command are zero 2. Quick Stop or Halt is active, there is no HW Switch active and Speed command and Reference Speed Command are zero	
1	The position command (**DV[3]**) has reached the position target but actual position is out of position window .	
2	The position command is in motion, the position command (**DV[3]**) has not reached the position target.	
3	The motor is disabled.	

In profile position mode, when a position limit is reached e.g. Software Position Limit or FLS\RLS, **MS** does not change. This means that if for example the limit switch is set while **MS** is 2, the motor stops but **MS** remains 2.

Time-dependent Motion Modes

In motion modes which are time-dependent (Interpolated Position, Cyclic Synchronous Position) the following behavior occurs:

- **In Interpolated Position Mode (0x6061=7) MS** reports 0 when the velocity command and external reference velocity command (follower/ECAM, refer to **EM[1]**, **RM**, **EI** commands) are 0 due to following reasons: there are no change in Master set points, Halt, Quick Stop, Software limit.
- **In Interpolated Position Mode (0x6061=7) MS** reports 2 when the velocity command is not 0 or HW switch is sensed
- **In Cyclic Synchronous Modes** (0x6061 = 8,9 or 10) MS reports 2
- **In all time-dependent modes** motion status is set to 3 when the motor is disabled.

Simple Profilers

Simple profiler motion status **MS[N]** (**N=2,3,4**) refers to the state of the specific simple profiler and does not take into account actual feedback.

- Motion status is set to 3 when the motor is disabled.
- Motion status is set to 2 when the profiler is initiated and is on the move.
- Motion status is set to 0 when target is reached

References

TC, **TR[]**, **DV**, **JV**

N1…N2[N] – Non-Linear Sub Tables

N1…N2[*N*] is used to address the 256 entries of the non-linear table **NT[N]**.

CANopen/CoE

Elmo Alias Object: 0x31D5, 0x31D6

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restriction	None
Range	NA
Default	None
Index range	1 to 250
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

The following table describes the relationship between **NT** and **N1….N2[*N*]** commands:

N# command	NT command	CANopen object
N1[1…250]	**NT[1…250]**	0x31D5
N2[1…6]	**NT[251…256]**	0x31D6

References

NT[N]

NF[] – Non-Linear Float

NF[] specifies the non-linear float table used for various implementations.

CANopen/CoE

Elmo Alias Object: 0x31E3

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	See the table below.
Index Range	1 to 12
Unit modes	See the table below.
Non-Volatile	Yes
Axis Related	Yes

Remarks

NF[1] and **NF[2]** are available for velocity and for position control. The value is the current, used at beginning of motion to overcome friction.

NF[10] and **NF[11]** - This algorithm enables the definition of a dead-zone area (with hysteresis), around the axis commanded desired position location. This algorithm is usually used to avoid limit cycle near zero, in High Friction applications

This algorithm zero the factor of velocity feed forward (defined by FF[2]), if the absolute distance from the motion final target is less then (or equal to) **NF[12]**.

Indices

The following table defines the **NF[N]** entries defined by the command index **(N)**.

Index	Description	Units	Default	Range	Comments
0	Reserved				
1	Specifies the current that will be used at the beginning of positive motion	Amperes	0	0 to **PL[1]**	
2	Specifies the current that will be used at the beginning of negative motion.	Amperes	0	-**PL[1]** to 0	

<table>
<tr><th>Index</th><th colspan="3">Description</th><th>Units</th><th>Default</th><th>Range</th><th>Comments</th></tr>
<tr><td rowspan="7">3</td><td colspan="3">Cogging compensation configuration</td><td></td><td rowspan="7">0</td><td rowspan="7">0 to 7</td><td rowspan="7"></td></tr>
<tr><td>Bits</td><td colspan="2">Description</td><td></td></tr>
<tr><td rowspan="3">0…1</td><td>0x1</td><td>Cogging compensation by sine wave</td><td></td></tr>
<tr><td>0x2</td><td>Cogging compensation by NT[N] table</td><td></td></tr>
<tr><td>0x3</td><td>Cogging compensation by mechanical position</td><td></td></tr>
<tr><td rowspan="2">2</td><td>0x0</td><td>Disable cogging compensation</td><td></td></tr>
<tr><td>0x1</td><td>Enable cogging compensation</td><td></td></tr>
<tr><td>4</td><td colspan="3">Cogging compensation sine amplitude</td><td>Amperes</td><td>0</td><td>0 to CL[1]</td><td></td></tr>
<tr><td>5</td><td colspan="3">Cogging compensation sine harmonics in electrical cycle</td><td></td><td>0</td><td>0 to 12</td><td></td></tr>
<tr><td>6</td><td colspan="3">Cogging compensation sine offset</td><td>Internal electrical angle units</td><td>0</td><td>-4095 to 4095</td><td></td></tr>
<tr><td>7</td><td colspan="3">Speed integral drain</td><td>Amperes/ second</td><td>0</td><td>>=0</td><td>Can be overwritten by object 0x3605, if GS[12]=1</td></tr>
<tr><td>8</td><td colspan="3">Position error symmetrical dead zone</td><td>Position UU</td><td>0</td><td>>=0</td><td></td></tr>
<tr><td>9</td><td colspan="3">Number of cogging cycles per revolution</td><td></td><td>0</td><td>>=0</td><td></td></tr>
<tr><td>10</td><td colspan="3">Dead zone algorithm min value</td><td rowspan="3">Position UU</td><td>0</td><td>>=0</td><td rowspan="3">Feature for high friction systems, Enable\Disable by AB[9]</td></tr>
<tr><td>11</td><td colspan="3">Dead zone algorithm max value</td><td>0</td><td>>=0</td></tr>
<tr><td>12</td><td colspan="3">Zero feed forward (FF[2]) abs position error</td><td>0</td><td>>=0</td></tr>
</table>

References

PL[], GS[], AB[], FF[]

NT – Non-Linear Table

NT specifies the non-linear table used for various implementations.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	NA
Range	NA
Index range	1 to 256
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

At the present time it is used for cogging compensation.

Indices

The following table describes the available options for **NT[]**.

Index	Description	Values	Comments
1 to 256	Current added to current control command for cogging compensation	Amperes	

References

NF[]

OB[N] – Output Bits

OB[*N*] sets and resets a general-purpose output bit.

CANopen/CoE

Elmo parameters objects: 0x3203

Attributes

Attribute	Description
Type	Unsigned short, Array, Read/Write
Source	All
Restrictions	**Reflect the OP command in a bit oriented manner. OB[1] reflects bit 0 in OP and OB[32] reflects bit 31 in OP.**
Range	0, 1
Index range	1 to 32
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

When **OB[*N*]** is queried, it reflects the **OP** command bits 0 to 31, respectively.

The **OB[N]** command allows the setting or getting of specific bits of the **OP** parameter.

For example, if digital output 2 is defined as general-purpose (refer to the **OL[]** command), the command **OB[2]** = 1 will set the digital output to active position (depending on the logic level of this output).

OB[1] to **OB[32]** represent the **OP** command register bits 0 to 31.

For example:

OB[1], like **OP** bit 0, represents the General-Purpose Output 1 level value.

OB[4], like **OP** bit 3, represents the General-Purpose Output 4 level value.

Gold drives support a variety of digital outputs. The number and details of these digital outputs is specified in the drive's Installation Guide. The value of **OB[*N*]** varies according to the logic state of the output even if the output does not exist.

The **OB[*N*]** syntax may be more convenient than **OP** for setting individual outputs. However, it is not appropriate for the synchronized setting of several output bits.

Setting of **OB[*N*]** will not affect the digital outputs which are not defined as General Purpose. Setting a non-General Purpose output does not affect the output, does not burst and alarm.

OB[*N*] reflects the logical values of the outputs. It does not however inform the physical level. The physical level depends on the method the output is connected externally.

OB[] command are not affected by **OL[]** command polarity bit (bit 0). Polarity bit has impact on physical output only.

When the drive reboots (power-up or during firmware download), output ports are set internally to 'not conduct'. This way no transition will occur during boot up.

Outputs 14, 15, 16 (bits 13-15) are connected to hardware PORT C outputs, where output 14 is PortC_A output 15 is PortC_B, output 16 is PortC_Index. These outputs can be used for functions defined with **OL[]** when relevant **GO[]** command is set to 0.

Indices

The following table describes the **OB[*N*]** entries.

Index	Description	Type	Values	Restrictions
1 to 4	**If the digital output is defined as a general-purpose output, OB[1]…OB[4] return the value of a digital output 1 to digital output 4.** Otherwise, it returns 0.	Long	0, 1	
5 to 13	Reserved, return 0	0		
14 to 16	If digital output is defined as general purpose output, it returns the value of digital output 14 to digital output 16.	Long	0,1	
17	If at least one of the digital outputs is mapped to the Amplifier OK function: • when the drive is ready to be enabled (no amplifier exception such as under voltage which prevents **MO=1**), **OB[17]** returns 1; • when the drive is not ready, **OB[17]** returns 0. **If none of the digital outputs is mapped to the Amplifier OK function, OB[17] returns 0.**	Long	0, 1	
18	**If at least one of the digital outputs is mapped to the Brake function:** • when the brake is engaged, **OB[18]** returns 1. • when the brake is released, **OB[18]** returns 0. **However, if none of the digital outputs is mapped to the Brake function, OB[18] returns 0.**	Long	0, 1	

Index	Description	Type	Values	Restrictions
19	**If at least one of the digital outputs is mapped to the Servo enable/disable function, see SO[] command:** • when **SO** = 1, **OB[19]** returns 1 • when **SO** = 0, **OB[19]** returns 0 **However, if none of the digital outputs is mapped to the Servo enable/disable function, OB[19] returns 0.**	Long	0, 1	
20	**If one of the digital output is mapped to Fault indication:** • when **MF** is set to a value different than 0, **OB[20]** returns 1. • when **MF** is set to 0, **OB[20]** returns 0. **However, if none of the digital output is mapped to Fault indication, OB[20] returns 0.**	Long	0, 1	
21	**If one of the digital output is mapped to Target Reached indication:** • when **MS** is set to 0, **OB[21]** returns 1. • when **MS** is set to 1,2 or 3 **OB[21]** returns 0. (**MS=0** is reflected in bit 10 of object 0x6041 Statusword) **However, if none of the digital output is mapped to Target Reached, OB[21] returns 0.**	Long	0, 1	
21 to 32	Reserved, return 0	0		

References

OP, **OL[N], GO[N]**

OC[] – Output Compare

OC[*N*] allows the generation of pulses when a predefined position is reached.

CANopen/CoE

Alias objects: 0x3204

Attributes

Attribute	Description
Type	Double
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 20, for Output Compare module 1 21 to 40, for Output Compare module 2
Default	According to array index
Unit modes	All
Non-volatile	Yes, exclusive **OC[1]**, **OC[21]**
Axis related	Yes

Remarks

The Platinum drive supports two user programmable Advanced Output Compare modules AOC1 and AOC2 that can be programmed via the **OC[1]…..OC[20] and OC[21]…..OC[40]** commands respectively. The modules can operate simultaneously.

For detailed information about the Output Compare mode refer to the MAN-P-ADMINGUIDE.

AOC1 and AOC2 modules can be connected to any PORT (Port A, Port B or Port C) where the configuration of the selected PORT is based on the feedback socket via **OC[6]** (for compare module 1) or **OC[26]** (for compare module 2).

Note:
The following text refers to **OC[1]** to **OC[20]**, but also applies to the **OC[21]** to **OC[40]** command unless stated differently.

Output Compare operates with following sensors:

- Encoder Quad, Port A, Port B, Port C. Sensor ID =1
- Pulse and Direction sensor, Port A, Port B. Sensor ID =11
- Up and Down counter sensor, Port A, Port B. Sensor ID =12
- Analog Sin/Cos sensor, Port B, Sensor ID =3

- Serial absolute - Hiperface Port A & Port B sensor – based on the Sine\Cosine signals of Port B, Sensor ID = 28
- If emulation is used then the Quad Emulation FB socket can also be used (**EM[8]**)

The **AOC1 and AOC2** generate a train of pulses according to a predefined position value.

The **OC[1]** enables and disables the mode and operates in following operation modes:

OC[1]=1: Absolute Start Position with Fixed Delta Position Mode	The first pulse is generated by the initialized absolute position defined by **OC[2]** (i.e. **PX**=**OC[2]**) and continues at position intervals specified by the **OC[3]** value (i.e. **Compare Position(k)**=**OC[2]**+k*__OC[3]__, where k is the number of successful compared occurrences (k=1,2…).
OC[1]=3: Absolute Position Table, Toggle Mode.	In this mode positions are taken from a table. The table includes pairs of absolute positions. Each pair includes: • Start position – the absolute position to start the pulse, and • End position – the absolute position to end the pulse.
OC[1]=4: Absolute Position Table with Duration Time Mode	In this mode the positions are taken from a table. The table includes Start Position in which the pulse generation starts The pulses duration is indicated in **OC[4]**. The number of pulses can be set via **OC[14]** - the Pulse repetition.
OC[1]=5: Software Start Mode	In this mode the AOC1 and AOC2 generate pulses by a software command and as soon as the mode starts. Pulse duration, and pulse repetition and repetition period are set via **OC[4], OC[14], OC[16]** respectively . The main purpose of this mode is for debugging and both AOC must be activated i.e., **OC[1]**=5; **OC[21]**=5 in order for the pulses to be generated. There are two conditions: • Standalone where every channel generates pulses by its initialization command and • Synchronized where both channels start to generate pulses synchronized to the last initialization command (**OC[1]=5** or **OC[21]=5**)
OC[1]=6: Hardware Auto Increment	This mode is used to generate pulses in very small intervals between pulses. The mode is similar to **OC[1]**=1 of ,where the delta position interval denoted by **OC[3]** is handled by the hardware in an Auto Increment mode**.** The mode can be used when Delta position **OC[3]** meets condition: -32768 <= **OC[3]** <= 32767.

OC[1]=7: Absolute Position Table Toggle Cyclic Mode	This is the same mode as **OC[1]=3**, but with option to refill table on Output Compare enable state. For details refer to the MAN-P-ADMINGUIDE.
OC[1]=8: Absolute Position Table with Duration Time Cyclic Mode	The same mode as **OC[1]=4**, but with option to refill table on Output Compare enable state. For details refer to the MAN-P-ADMINGUIDE.
Advanced output compare mode	In this mode one of the AOC modules is generating the strobe for the other AOC module. For this both AOC1 and AOC2 must be activated. The position compare event is generated by one of them according to **OC[18]** and **OC[38]**. The mode is available in modes **OC[1]/OC[21]=1, 6 or 4.** Refer to the MAN-P-ADMINGUIDE for more details.
Compare Mask Mode	This mode allows the user to mask one of the AOC modules based on pulses that are generated by the other AOC. This means: AOC1 and AOC2 are both active, and output pulses of AOC1 are masked by AOC2 pulses, or vise-versa. This mode is done with settings of **OC[1]/OC[21]**=1,3,5 and **OC[13]/OC[33]**=1. Refer to the MAN-P-ADMINGUIDE for more details.
Pulse Repetition Sub-Mode	An optional pulse repetition can be activated via **OC[14]**. The Pulse repetition means that after the compare event a defined number of pulses can be generated with defined time differences between the pulses. Note that this sub-mode is not supported in **OC[1]**=3**.**
Pulse generation Delay	All modes can use a time delay for the first pulse generation after the strobe is received. The delay is defined via **OC[15]** for ACO1 and **OC[35]** for AOC2.

“Generate pulse” or “Active high” means that the hardware will set the output to ‘1’ – causing current to flow through the opto-coupler.

Be aware that long time duration pulses can cause a number of pulses to overlap. The drive gives no warning on such situations.

Homing and capture modes (i.e., DS-402 Homing mode, DS-402 Touch-Probe, **HM[1]** or **HF[1]**) can operated when the output comparison operates on the same feedback source. In this case the high level application has to consider that applying homing offset may result in undesirable pulses.

In **Absolute Position Table Toggle Mode,** the positive/negative value of the position interval (**OC[3]/OC[23]**) should be set according to the desired direction. When the direction is positive (increasing **PX** values), the position intervals should be positive; otherwise, they should be negative. If an axis moves in opposite direction to **OC[3]/OC[23],** no pulses are generated, regardless of using module.

In the **table single direction mode** (**OC[10]**=1,2), the pulses are generated according to the order in the table from first position index (**OC[8]**) to the last position index settings (**OC[9]**). This order is maintained even if the axis moves in the direction opposite to the specified direction (**OC[10]**). Note, that the number of pulses which is expected to be generated and defined within the range of **OC[8]** to **OC[9]** then only one pass through the table generates the pulses.

The first position must be at least **HS*TS** microseconds from the activation position (the drive position when sending the **OC[1]** command),

Changing the motion direction, in both direction mode can be done during HS*TS microsecond, faster change may result in loss of pulses where no more pulses will be generated.

In table modes compare, the table's data is set in user units and is converted from user units to counts during the activation of the comparison mode. An additional activation of comparison in the table mode **must be preceded by refilling the table** with the user unit's positions, or the setting of **OC[11]/OC[31]** to 1.

In **table mode,** if both directions are selected (OC[10]=0), the comparison operates infinitely and will only end with the **OC[1]=0** command.

In **table modes** in both directions, the pulse generating sector is in the table boundaries (the table's index is not rolled). If the movement passes the last position in the table, the next pulse position to generate is the last table position. If the movement passes the first position in the table, the next position generated is the first position in the table.

When the table mode is used (**OC[1]**=3/4), the user can edit it via **GV[]**commands. Sizes of the tables are **1024** entries.

The Output compare and the Capture are using the same physical table for position. **GX, G1 – G5** commands are used to address the table and retrieve the captured position of **HM[]** and **HY[]**, and **GV[]** is used to address the table and set the desired Compare values for the **OC[]**.

The table includes 1024 entries, and the user must be noted to select the proper indices for captures and compare modes to prevent usage of the same table index for different purposes.

The drive does not check if indices are used in different modes.

Changing sensor position during an output comparison, will **not** re-evaluate the comparison points in the compare table automatically. After sensor position change (via Homing or setting of **PX**) the output compare must be reinitialized to prevent offset, and if table mode is selected the compare table positions must be converted – see **OC[11]\OC[31]**.

Output Port C_A, Port C_B and Port C_I, are fast outputs allowing accurate pulse length and response time. Outputs 1-8 are slow outputs (relatively) and might distort the pulse timing, adding additional time to the pulse.

In case that the compare feedback is not Quad encoder, the compare function cannot be activated and **OC[1]/OC[21]** returns error 103 "Output Compare sensor not Quad Sensor".

Indices

The following table describes the **OC[]** entries.

AOC1/ AOC2 Index 1…20 Index 21…40	Description		Default Value
1, 21	**Write Values**		0
	0	disable output compare	
	1	Absolute Start Position with Fixed Delta Position Mode.	
	2	Reserved (returns error "out of range")	
	3	Absolute Position Table Toggle Mode.	
	4	Absolute Position Table with Duration Time Mode.	
	5	Software Start Mode. *Note 10	
	6	Absolute Start Position with hardware Auto Increment Mode. ***Note 4**	
	7	Absolute Position Table Toggle Cyclic Mode.	
	8	Absolute Position Table with Duration Time Cyclic Mode**.**	
	Read Values		
	-1	No more pulses are being generated because the number of pulses/table entries specified in **OC[5]** has been reached	
	0	Output Compare module is disabled	
	1	In Absolute Start Position with Fixed Delta Position Mode: The Output Compare function has started, but the absolute position **OC[2]** has not yet been reached, therefore, the train of pulses has not begun. In Absolute Position Table Toggle Mode, Absolute Position Table Toggle Cyclic Mode, Absolute Position Table with Duration Time Mode, Absolute Position Table with Duration Time Cyclic Mode: The Output Compare function has started, but the first position in the table has not yet been reached, therefore, the train of pulses has not begun.	
	2	In Absolute Start Position with Fixed Delta Position Mode: The Output Compare function has started and the absolute position **OC[2]** has been reached, but the number of pulses entries specified in **OC[5]** has not been reached; therefore, the train of pulses has begun. In Absolute Position Table Toggle Mode, Absolute Position Table Toggle Cyclic Mode, Absolute Position Table with Duration Time Mode, Absolute Position Table with Duration Time Cyclic Mode: The Output Compare function has started, the first position in the table has been reached, but the last position has not been reached; therefore, the train of pulses has begun.	

AOC1/ AOC2 Index 1…20 Index 21…40	Description	Default Value
2, 22	The absolute position of the first pulse in user units (depend on the feedback selected). This value cannot exceed the modulo limit in the same direction of motion i.e. **|OC[2]|-|PX|** must be positive. Relevant only for Absolute Start Position with Fixed Delta Position Mode **(OC[1]=1)** and Absolute Start Position with hardware Auto Increment Mode **(OC[1]=6).** Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	0
3, 23	The hardware position intervals between subsequent pulses (in user units). The positive/negative value of **OC[3]** should be set according to the encoder motion. When the direction is positive (increasing PX/Sensor-position value) **OC[3]** should be positive; otherwise, it should be negative. Relevant only for Absolute Start Position with Fixed Delta Position Mode **(OC[1]=1)** and Absolute Start Position with hardware Auto Increment Mode **(OC[1]=6).** The value cannot be set to 0 Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	1000
4, 24	N: Pulse duration in microseconds Minimum value 1 Maximum value 1310.6 **(*Note 2)** Relevant only for Absolute Start Position with Fixed Delta Position Mode **(OC[1]=1),** Absolute Start Position with hardware Auto Increment Mode **(OC[1]=6)** and for Absolute Position Table with Duration Time Mode **(OC[1]=4)**	100
5, 25	N: number of pulses to generate. **0:** infinite output compare mode (train of pulses will end only with the **OC[1]=0 command).** Max value is 65534 Relevant only for Absolute Start Position with Fixed Delta Position Mode **(OC[1]=1)** and Absolute Start Position with hardware Auto Increment Mode **OC[1]=6.** Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	0
6, 26	Output Compare source signal: 0: Output compare on Position-Feedback. 1-8: Output compare on socket number (1-8). In both cases the socket must be configured as one of the following: Encoder Quad, Analog Sine/Cosine or Hiperface sensor. Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	0
7, 27	**OC[7] - Output Compare 1, OC[27] - Output compare 2 See the next Table Index.** The command indicates next table index to be used for when Output Compare is active. Refer to Platinum Administrative Guide, Object 0x360D. Relevant only for table modes **OC[1]/OC[21] = 3, 4, 7, 8**	0

AOC1/ AOC2 Index 1…20 Index 21…40	Description	Default Value
8, 28	Tables first position index. Relevant only for table modes **OC[1]=3|4.** Validated only at mode enable. Min value 1, Max value 1024 Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	1
9, 29	Tables last position index. Relevant only for table modes **OC[1]=3|4.** Validated only at mode enable. Min value 1, Max value 1024 Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	1
10, 30	Axis direction:	0
	0 — Both directions. In this case the compare mode is infinite (the train of pulses will stop only with the **OC[1]=0 command).**	
	1 — Positive direction only.	
	2 — Negative direction only.	
	For details refer to the MAN-P-ADMINGUIDE. Relevant only for table modes **OC[1]=3|4.** Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	
11, 31	Convert table positions **0 –** convert table position entered by user from user units to Hardware (sensor) units (including error mapping if enabled **(PC[])), at OC[1]** enable command. 1 – Do not convert the table positions from user units to Hardware units. Relevant only for table modes **OC[1]=3|4.** Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	0
12, 32	Number of pulses N generated since mode enabled. Note that the value is not reset N=0… 2147483647. See *Note 3 Not relevant for advanced mode if the module is the slave **(OC[18\38] = 2)**	0
13	0 – AOC1 connected to OC1 1 – AOC1 connected to OC1 & OC2. See *Notes 7, 8 Value 1 is relevant for the modes where both software modules **OC[1] and OC[2]** are initialized	0
33	0 – AOC2 connected to OC2 1 – AOC2 connected to OC1 & OC2 Value 1 is relevant for the modes where both software modules **OC[1] and OC[2] are initialized. See *Notes 7, 8**	0
14, 34	Number of repetitions N: N=0 – Infinite number of pulses N=1…254 – N pulses	1

AOC1/ AOC2 Index 1…20 Index 21…40	Description	Default Value
	Not relevant for Table Toggle mode **(OC[1]=3), ignored** In Absolute Position Table with Duration Time Mode **(OC[1]=4)** infinite number of pulses cannot be set, returns error 7 "Bad mode initial data"	
15, 35	Delay from compare event to output pulse in microseconds. Max value 1310.7 See *Note 1 Not relevant for Table Toggle Mode **OC[1]=3**	0
16, 36	Period of pulse repetition per compare event in microseconds. Min value 0.04, Max value 1310.7 Relevant only if **OC[14]/OC[34] >=2** Must be set greater than Pulse width **OC[4]/OC[24], else OC[1]/OC[21]** returns error "Bad initialization data"**, error code 7** Not relevant for Table Toggle Mode **OC[1]=3**	200
17, 37	Output Compare pulse start polarity: 0 – Low level start polarity, positive pulses (Set **GO[n]=0;OL[n]=1**; before enable Output Compare with **OC[1]**) 1 – High level start polarity, negative pulses (Set **GO[n]=0;OL[n]=0**; before enable Output Compare with **OC[1]**)	0
18, 38	Advanced output compare mode: 0 – no advanced mode 1 – Master 2 – Slave Possible settings are: **OC[18]=1, OC[38]=2 –** advanced mode synchronized with **OC1 (OC1 is Master, OC2 is Slave)** **OC[18]=2, OC[38]=1 -** advanced mode synchronized with **OC2 (OC2 is Master, OC1 is Slave)** Can be set to non-zero value only for **OC[1]/OC[21]=1, 4 or 6.** In rest modes it is ignored. ***Note 9**	0
19, 39	Software Start Mode parameter: 0 – stand alone, 1 – synchronized **(Relevant only for software Start Mode OC[1]=5)** ***Note 10**	0
20, 40	Digital output assignment. Defines which digital output or Port C pin is assigned to the output pulses. ***Note 11**	0

Notes

Note 1. **OC[15]/OC[35]**, Delay from compare event to output pulse: Max delay supported by is 65535 which is 1.3107 milliseconds.

Note 2. **OC[4]/OC[24]**, Pulse width: Max pulse width supported is 65535 which is 1.3107 milliseconds.

Note 3. The value of **OC[12]** is the number of generated pulses. The user may write any value at any time also while the OC is active. When the mode is not enabled (**OC[1]=0**) read value equals to write value. When the mode is enabled (**OC[1]!=0** command) the read value is the written value + number of pulses already generated. The value reported by **OC[12]** is not dependent on pulse repetition number (**OC[14]**), it is related only to the number of Position Compare events.

Note 4. The mode **OC[1]=6** is the mode of **Absolute Start Position with Fixed Delta Position Mode** similar to **OC[1]=1** , but with **HW Auto Increment.** The mode can be used when Delta position **OC[3]** meets the condition: -32768 <= **OC[3]** <= 32767. Pulse repetition is not supported in this mode. The mode is not supported by Titanium drive for ports B1 and B2.

Note 5. The mode **OC[1]=6** is applicable under the condition that between the delta time, the consequent pulses are less than TS time.

Note 6. When using Advanced mode Note, first must be sent all desired parameters **OC[2]...OC[19], OC[22]...OC[39]**, after that initialization commands **OC[1]** and **OC[21]**

Note 7. There are some restrictions for using the feature in the Platinum drive, and the possible combinations for non-advanced mode are:

- Axis 1 **OC[1]** & Axis 2 **OC[1]**
- Axis 1 **OC[21]** & Axis 2 **OC[21]**

Note 8. There are some restrictions for using the feature, in the **Platinum Quartet drive**, and the possible combinations for non-advanced mode are:

- Axis 1 **OC[1]** & Axis 2 **OC[1]**
- Axis 1 **OC[21]** & Axis 2 **OC[21]**
- Axis 3 **OC[1]** & Axis 4 **OC[1]**
- Axis 3 **OC[21]** & Axis 4 **OC[21]**

Note 9. Titanium drive does not support Output Compare Advanced mode. OC[18], OC[38] must be set to 0.

Note 10. Titanium drive does not support Software Start Synchronous mode. OC[19], OC[39] must be set to 0

Note 11. **OC[20]/OC[40]** commands are different for Platinum, Quartet and Titanium drives. See following tables:

Platinum drive	
OC[20]/OC[40] value	**Output assigned to Output Compare**
0	No output Write access has no effect
1, 3	Output 1, 3
2, 4 - 8	Reserved Write access has no effect
9	Port C A

Platinum drive	
10	Port C B
11	Port C Index

Note: In the following three tables, Axis 1 – Axis 4 is regarded as AOC1 output – AOC4 output

Quartet drive					
Output Compare 1 (OC[1]) Digital Output assignment					
OC[20] value	0, 5.....11	1	2	3	4
Axis 1	Output 2	Output 2	Output 4	Output 6	Output 8
Axis 2	Output 4	Output 2	Output 4	Output 6	Output 8
Axis 3	Output 6	Output 2	Output 4	Output 6	Output 8
Axis 4	Output 8	Output 2	Output 4	Output 6	Output 8

Quartet drive				
Output Compare 1 (OC[1]) Port C Assignment				
OC[20] value	12	13	14	15
Axis 1 – Axis 4 (AOC1 – AOC4)	Port C1_A	Port C1_B	Port C2_A	Port C2_B

Quartet drive					
Output Compare 2 (OC[21]) Digital Output assignment					
OC[40] value	0, 5.....11	1	2	3	4
Axis 1	Output 4	Output 2	Output 4	Output 6	Output 8
Axis 2	Output 2	Output 2	Output 4	Output 6	Output 8
Axis 3	Output 8	Output 2	Output 4	Output 6	Output 8
Axis 4	Output 6	Output 2	Output 4	Output 6	Output 8

Quartet drive				
Output Compare 2 (OC[21]) Port C assignment				
OC[40] value	12	13	14	15
Axis 1 – Axis 4 (AOC1 - AOC4)	Port C1_A	Port C1_B	Port C2_A	Port C2_B

Titanium drive		
Digital Outputs and Ports assignment		
Titanium drive OC[20]/OC[40] value	**Connected DOUT or Port**	**Remarks**
0 (default)	Output Compare 1 connected to DOUT 1 Output Compare 2 connected to DOUT 2	
1	DOUT 1	
2	DOUT 2	
3	DOUT 3	
4	DOUT 4	
5	Not used, write access has no effect	
6	Not used, write access has no effect	
7	Not used, write access has no effect	
8	Not used, write access has no effect	
9	DOUT 9, OP bit 8 (Port 1 A_A)	
10	DOUT 10, OP bit 9 (Port 1 A_B)	
11	DOUT 11, OP bit 10 (Port 1 A_I)	
12	DOUT 12, OP bit 11 (Port 1 B_I)	
13	DOUT 13, OP bit 12 (Port 2 A_A)	
14	DOUT 14, OP bit 13 (Port 2 A_B)	
15	DOUT 15, OP bit 14 (Port 2 A_I)	
16	DOUT 16, OP bit 15 (Port 2 B_I)	

References

GO[], GV[], EA[N]

OL[N] – Output Logic

OL[] specifies the digital output function and logic. The command is axis related.

CANopen/CoE

Elmo Alias Object: 0x320D

Attributes

Attribute	Description
Type	Unsigned short, Bit field, Read/Write
Source	All
Restrictions	None
Range	0 to 11
Index	1 to 16
Default	0xFFFF
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

OL[*N*] is a bit-field command which allows the user to map any of the uncommitted digital outputs of the drive to any function and desired logic level.

The function description is described in Digital Output Function Description.

The *logic level* determines the relation between the output activation and the current flow from the drive. The actual hardware functionality of the output is hardware dependent.

Positive logic (active high) means that if the function is activated, the drive sets 1 for the relevant digital output pin and the opto-coupler is conducting.

Negative logic (active low) means that if the function is activated, the drive sets 0 for the relevant digital output pin and the opto-coupler is not conducting.

Platinum Quartet drive

The Platinum Quartet drive has 8 digital outputs distributed between 4 axes as presented in following table.

Axis	Digital Output	Related OL[] command
1	OUT1	Ax1.OL[1]
	OUT 2	Ax1.OL[2]
2	OUT 1	Ax2.OL[3]

Axis	Digital Output	Related OL[] command
	OUT 2	Ax2.OL[4]
3	OUT 1	Ax3.OL[5]
	OUT 2	Ax3.OL[6]
4	OUT 1	Ax4.OL[7]
	OUT 2	Ax4.OL[8]

Indices

Index	Description
1 to 4	DOUT functions described in table below are available when **GO[]** is set to 0
5,6	Setting **OL[]** has no effect
7,8	DOUT functions described in table below are available when **GO[]** is set to 0
9 to 13	Setting **OL[]** has no effect
14 to 16	**For Platinum drive only** Port C is used. DOUT functions described in table below are available when **GO[]** is set to 0

Titanium 2 Axes Drive

Some pins of ports A1 and A2, can be used as Digital Outputs in the case when they are not used for sensor signals.

The following Digital Outputs are connected to ports:

Axis 1 Port B_I	Axis 1 Port B_A	Axis 1 Port A_I	Axis 1 Port A_A
DOUT 12 OP bit 11	DOUT 11 OP bit 10	DIN 10 OP bit 9	DOUT 9 OP bit 8

Axis 2 Port B_I	Axis 2 Port B_A	Axis 2 Port A_I	Axis 2 Port A_A
DOUT 16 OP bit 15	DOUT 15 OP bit 14	DOUT 14 OP bit 13	DOUT 13 OP bit 12

Values

The following table describes the bit-field entries of **OL[]** for logic and function.

Bit	Description	Type	Values	
0	Logic level	Boolean	0	Active low
			1	Active high
1 to 15	Function behavior * See the detailed description in Digital Output Function Description below.	Unsigned short	0	General Purpose
			1	**AOK function**
			2	Brake
			3	**Servo State (SO)**
			4	**Motor Fault (MF)**
			5	**Target Reached (MS)**
			6…32766	Reserved
			32767	Ignore, no function is mapped to this DOUT

Possible Values of OL[N]

The following table lists the possible values of **OL[*N*]**.

Command Value	Logic Level	When Active…..
OL[*N*] = 0	Low	Output is general-purpose.
OL[*N*] = 1	High	Output is general-purpose.
OL[*N*] = 2	Low	**AOK indicates that the drive is ready for use.**
OL[*N*] = 3	High	**AOK indicates that the drive is ready for use.**
OL[*N*] = 4	Low	Brake feature is active.
OL[*N*] = 5	High	Brake feature is active.
OL[*N*] = 6	Low	Servo enable/disable indication
OL[*N*] = 7	High	Servo enable/disable indication
OL[*N*] = 8	Low	Motor was disabled due to a fault.
OL[*N*] = 9	High	Motor was disabled due to a fault.
OL[*N*] = 10	Low	Target Reached indication.
OL[*N*] = 11	High	Target Reached indication.
OL[*N*] = 12…32765	-	Reserved
OL[*N*] = 65534	Low	**Ignore, no function is mapped to this DOUT**
OL[*N*] = 65535	High	**Ignore, no function is mapped to this DOUT, default**

Note:
Changing the Logic level without changing the assigned function of a digital output (for example **OL[1]** change from 0 to 1) will cause the physical output to change, without effecting the related **OB[]** value and related bit in **OP**. Two examples are presented in following tables.

OL[1]	OB[1]	OP.0	LED
0 (GP output, active level low)	0	0	ON
1 (GP output, active level high)	0	0	OFF
0 (GP output, active level low)	1	**1**	OFF
1 (GP output, active level high)	1	**1**	ON

OL[1]	Actual amplifier state	OB[17]	OP.16	LED
3 (AOK, Active high)	OK	1	1	ON
2 (AOK, Active low)	OK	1	1	OFF
3 (AOK, Active high)	**Set XP[13]=0 to get "Overvoltage state"**	0	**0**	ON
2 (AOK, Active low)	**Set XP[13]=0 to get "Overvoltage state"**	0	**0**	OFF

Digital Output Function Description

Function 0: General purpose

The output is general-purpose (has no special automatic function) and can be set or reset by the **OP** or **OB[N]** command from any source.

Function 1: AOK

The **Amplifier OK** function indicates that the physical condition of the drive allows the motor to be enabled. If a digital output is assigned to **AOK** it will be automatically reset to 0 if an amplifier fault occurs. An amplifier fault includes any of the following:

- Under voltage
- Overvoltage
- Safety state
- Short protection
- Over temperature

For more details about amplifier faults please refer to the **MF** command.

Function 2: Brake

The **Brake** function is an automatic function which logically sets the output according to the brake parameter time definition when the motor is either enabled (to disengage the brake) or disabled (to engage the brake). Refer to the **BP[N]** command for more details.

Function 3: Servo Enable

The output will be logically set in cases in which the servo is enabled. Refer to the **SO** command for more details.

Function 4: Motor Fault

The output will be logically set in cases in which the motor aborted to freewheel. Refer to **MF** command for more details. The output is reset to 0 when the motor is re-enabled or when a "Fault Reset" request is sent from the "Fault State" in the CANopen state machine.

In cases in which motor enable is requested by the **MO** = 1 command or by the CANopen state machine, the fault is reset even if the motor on procedure returned an error. For example, if during the motor on procedure it was detected that the safety switch is not conducting, the pervious fault indication is cleared and the output is set to 0.

Function 5: Target Reached

The output will be logically set in cases where the target set-point was accomplished. In such cases, the **MS** command indicates 0 and bit 10 of DS-402 Status Word (object 0x6041) is set as well. The output is logically set to 0 when motion starts. Refer to the **MS** command for more details.

Function 32767: Ignore Input

The output will be logically set to 0. **OP** and **OB[n]** commands return 0 in bit related to this output. **OL[n]** command returns 0xFFFF regardless of the digital output logic that was set (**OL[n]** = 0xFFFF or 0xFFFE).

Notes

- **OP** and **OB[]** command are not affected by **OL[]** command polarity bit (bit 0). Polarity bit has impact on physical output only.
- The Output Compare function is hardware-dependent and requires handling with the **OC[*N*]** and **GO[*N*]** commands. Please refer to these relevant commands.
- For outputs that are not supported by the product, the relevant index (the **OL** index) will be accepted, but ignored.
- In Emulation function, the logic level of the relevant output (**OL[14]** to **OL[16], TBD**) determines the emulation logic level. Refer to the **EA[N]** command for more details.
- If **GO[*N*]** is defined for Emulation, the relevant **OL[N]** function should not be activated.
- Platinum drive has one set of physical digital outputs DOUT1 to DOUT16 that are controlled by the axis related commands **AxN.OL[n]**, **AxN.OP**, **AxN.OB[n]** (where N – number of axis 1…4, 7, 8, 14…16, n – number of DOUT 1…16). In accordance with this, it is not recommended to assign the same DOUT for different axes for any function different from "Ignore" or GPO (like **Ax2.OL[3]**=1; **Ax3.OL[3]**=7; **Ax1.OL[3]**=3;). However, if the recommendation was disregarded, and the same DOUT was assigned, it should be noted that physical DOUTs are controlled by the commands related to axis with largest number regardless of the order the commands were entered (in our example: **Ax3.OL[3]**=7). Indications in **AxN.OP**, **AxN.OB[n]** are independent for every axis.
- When any DOUT is assigned to "Ignore" for all axes, it is disabled , "no current"

References

OP, **OB[N]**, **MO**, **OC[N]**, **GO[N]**, **EA[N]**

OP – Output Port

OP specifies the digital output port which can be used to set or reset the Digital Output from any source.

CANopen/CoE

Elmo Alias Object: 0x3211

Attributes

Attribute	Description	
Type	Unsigned long, Read/Write	
Source	All	
Restrictions	NA	
Range	Bit 0	General-purpose output 1 level Quartet drive: Axis 1 General-purpose output 1 level
	Bit 1	General-purpose output 2 level Quartet drive: Axis 1 General-purpose output 2 level
	Bit 2	General-purpose output 3 level Quartet drive: Axis 2 General-purpose output 1 level
	Bit 3	General-purpose output 4 level Quartet drive: Axis 2 General-purpose output 2 level
	Bit 4	Setting of this bit has no effect on output pin Quartet drive: Axis 3 General-purpose output 1 level
	Bit 5	Setting of this bit has no effect on output pin Quartet drive: Axis 3 General-purpose output 2 level
	Bit 6	General-purpose output 7 level (not implemented in HW) Quartet drive: Axis 4 General-purpose output 1 level
	Bit 7	General-purpose output 8 level (not implemented in HW) Quartet drive: Axis 4 General-purpose output 2 level
	Bit 8	**For Platinum and Quartet drives:** Reserved **For Titanium drive:** General-purpose output 9 X1 Port A_A
	Bit 9	**For Platinum and Quartet drives:** Reserved **For Titanium drive:** General-purpose output 10 X1 Port A_B

Attribute	Description	
	Bit 10	**For Platinum and Quartet drives:** Reserved **For Titanium drive:** General-purpose output 11, Port A Index
	Bit 11	**For Platinum and Quartet drives:** Reserved **For Titanium drive:** General-purpose output 12 X1 Port B_Index
	Bit 12	**For Platinum and Quartet drives:** Reserved **For Titanium drive:** General-purpose output 13, X2 Port A_A
	Bit 13	**For Platinum drive:** General-purpose output 14 level (Port C_A) **For Titanium drive:** General-purpose output 14, X2 Port A_B
	Bit 14	**For Platinum drive:** General-purpose output 15 level (Port C_B) **For Titanium drive:** General-purpose output 15, Port A Index
	Bit 15	**For Platinum drive:** General-purpose output 16 level (Port C_Index) **For Titanium drive:** General-purpose output 16, X2 Port B Index
	Bit 16	Amplifier OK output indication
	Bit 17	Break output indication
	Bit 18	Servo ON output indication, see command **SO[]**
	Bit 19	Motor Fault indication
	Bit 20	Target Reached (In Position) indication
	Bit 21-31	Reserved
Index range	NA	
Default	0	
Unit modes	All	
Non-Volatile	No	
Axis Related	Yes	

Remarks

Refer to **OL[]** command for detailed information about the output optional functions.

Outputs 14, 15, 16 (bits 13-15) are connected to hardware PORT C outputs, where output 14 is PortC_A output 15 is PortC_B, output 16 is PortC_Index. These outputs can be used for functions defined with **OL[]** when relevant **GO[]** command is set to 1.

Set values for all general -purpose digital outputs as defined in the **OL[]** command.

Querying **OP** indicates which digital output is logically activated.
For example, if digital output is defined as general-purpose and the user sets **OP** = 8, digital output 4 becomes active, and, depending on its logic level configuration, the specific output is set or reset.

Note that the **OP** command are not affected by **OL[]** command polarity bit (bit 0). Polarity bit has impact on physical output only.

Platinum drives support a variety of digital outputs. The number and details of these digital outputs is specified in the drive's Installation Guide.

OP does not affect the digital output pins otherwise defined as general-purpose.

The **OB[*N*]** command can be used to access (set and read) individual digital outputs rather than the whole port.

When any of the uncommitted digital outputs is defined as general-purpose, the physical state of the output depends on the previous **OL[], OP[]** command setting.

The **OB[*N*]** syntax may be more convenient than **OP** for setting individual outputs. However, it is not appropriate for the synchronized setting of several output bits. If a synchronized setting of several digital outputs is desired, use the **OP** command.

If the Output Compare (**OC[*N*]**) function is active, the defined output compare output is overridden by this function.

References

OB[N], **OL[N]**, **GO[N]**

PA[N] – Position Absolute

PA[N] specifies the absolute target positions for the drive profiles.

CANopen/CoE

Emo Alias Object: Not Available

PA[1] Target_position: 0x607A

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	• Motor must be on (except for **PA[4]**). • Effective on the next call to **BG[N]** • **PA[1]** must be within the ranges defined by **VH[3]**, **VL[3]**
Range	-2^{52} to $(2^{52}-1)$
Index range	1 to 4
Default	0
Unit modes	**UM = 3; UM = 5; UM = 6; UM = 7**
Non-Volatile	No
Axis Related	Yes

Remarks

Main profile (N=1):

PA command w/o index is equivalent to PA[1].

On the next BG[1] after applying PA[1] command:

The motion control will switch to the position control loop.

The drive will execute the Point-To-Point (PTP) motion according to AC[1], DC , SF, SP[1] and FS.

When PA[1] is set and BG[1] is commanded, the motion mode reflected in object 0x6061 is modified to 1 (Profile Position).

Object 0x607A will be overridden by the PA[1] value. Refer to the BG command for more details.

Simple (low capability) profiles (N=2,3,4).
On the next BG[N] after applying PA[N] command (N=2,3,4) the corresponding profile will be built according to AC[N] and SP[N]. SF, FS and DC commands are not supported for these profiles.

If UM = 3 or 6, PA[N] determines the target position in electrical angle units (1 pole pair is 360 electrical degrees, which are denoted by 4096 electrical ticks).

The motor will abort if the feedback position is higher than HL[3] or lower than LL[3].

The PA[N] value can be given in user-defined position units specified by the FC command.

Indices

The following table describes the profiles defined by the command index **(N)**.

Index	Profile name	Profile description
1	Main	The main drive profile (DS-402)
2	Superimposed	The profile is used for: • Superimposing the main profiler motion by the additional motion • Overcoming motion jumps when external reference generator is used (follower or ECAM)
3	Phasing	The profile is used during follower and ECAM mode to create a smooth phase shift in the master position of a slave axis.
4	Socket profile	The profile running in a socket at control interrupt, can be used for all functions of the socket.

References

JV, **FS**, **SP**, **AC**, **DC**, **UM**, **PR**, **PO**, **FC**

PC[] – Error Mapping Correction

PC[*N*] configures and enables the mode of the position error correction (AKA Error mapping)

CANopen/CoE

Manufacturer specific profile area objects: 0x3228

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 19
Default	According to array index
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

Error correction mapping is used to correct non-linear mechanical position errors.

This command enables the user to set error mapping parameters and to enable/disable error mapping mode.

The mode can be activated using a linear table, where the axis main feedback position is taken as an absolute entry in the table. The mode can also be activated as a cyclic mode (modulo), where the position is calculated for each modulo value, as defined by the user, producing endless cyclic position correction.

The table entries are the corrections which need to be made in a specific feedback location.

Platinum drive supports 1D and 2D error correction mapping modes

For more details on different error mapping options, please refer to the Error Mapping Correction chapter in the Platinum drive Administrative Manual.

Error mapping cannot be enabled during the following scenarios:

- While DS-402 homing is in progress (**Object 0x6060** sub mode 6)
- **HM**/**HF** are operational with homing mode,
  i.e., **HM[1]** > 0 and **HM[5]** != 2 or **HF[1]** > 0 and **HF[5]** != 2.

The following operations cannot be performed when error mapping is enabled:

- DS-402 homing
- Setting the position of the main position sensor (**PX** = xx, **PU=xx**)
- Activating the **HM**/**HF** with homing mode, i.e., **HM[2]** != 2 or **HF[2]** != 2.

While error mapping is enabled, all captured positions in **HM**/**HF** or Touch-probe are after the correction was considered. (i.e., positions after correction).

To edit the correction table, use the **ET[]** command.

The error mapping correction array **ET[N]** is also used by the ECAM. The user must make sure that there is no overlapping between ECAM and error correction portions of the **ET** array. This is not checked by the drive.

The overall corrected position (abscissa + error and ordinate + error) must be rising monotonously. This limitation directly implies that the correction table uniquely defines all positions, i.e., for each corrected position there is one and only one actual sensor reading that satisfies the following relation:

Corrected Position = Actual Position + Error Correction

This limitation is not checked by the drive before enabling error mapping. It is up to the user to verify this.

Indices

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Default</th><th>Range</th></tr>
<tr><td rowspan="8">1</td><td>Value</td><td>Operation</td><td rowspan="8">0</td><td rowspan="8">0 to 6</td></tr>
<tr><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>Enable linear mode</td></tr>
<tr><td>2</td><td>Enable cyclic (modulo) mode</td></tr>
<tr><td>3</td><td>Enable 2D linear (non-modulo for both dimensions) mode</td></tr>
<tr><td>4</td><td>Enable 2D cyclic (modulo X dimension) mode</td></tr>
<tr><td>5</td><td>Enable 2D cyclic (modulo Y dimension) mode</td></tr>
<tr><td>6</td><td>Enable 2D cyclic (modulo both dimensions) mode</td></tr>
<tr><td>2</td><td colspan="2">Reserved</td><td></td><td></td></tr>
<tr><td>3</td><td colspan="2">Reserved</td><td></td><td></td></tr>
<tr><td>4</td><td colspan="2">First index of the correction table (Dimension X in the case of 2D correction)</td><td>1</td><td>1 to 1999</td></tr>
<tr><td>5</td><td colspan="2">Last index of the correction table (Dimension X in the case of 2D correction)</td><td>2</td><td>2 to 2000</td></tr>
<tr><td>6</td><td colspan="2">Correction table position gap (Dimension X in the case of 2D correction)</td><td>10</td><td>$1<$PC[6] $<(2^{52}-1)$</td></tr>
<tr><td>7</td><td colspan="2">Error mapping start position, in user-defined units. (Dimension X in the case of 2D correction)</td><td>0</td><td>-2^{52} to $(2^{52}-1)$</td></tr>
<tr><td>8</td><td colspan="2">Reserved</td><td></td><td></td></tr>
<tr><td>9</td><td colspan="2">Axis number of dimension X (row).</td><td>1</td><td>1…6</td></tr>
<tr><td>10</td><td colspan="2">Reserved (for First Index dimension Y)</td><td></td><td></td></tr>
</table>

Index	Description	Default	Range
11	Number of rows in the table (Dimension Y size)	2	2 to 2000
12	Correction table position gap for dimension Y	10	1 to (2^{52}-1)
13	Error mapping start position (in user-defined units) for dimension Y	0	-2^{52} to (2^{52}-1)
14	Reserved		
15	Axis number of dimension Y.	2	1…6
16 - 19	Reserved		

Notes

- Correction table position gap is a position distance between two consequent points in the table in user units
- Indices 9 -15 are relevant for modes **PC[1]**=3-6 (2D error mapping) only
- Activating the mode with a reserved values returns error 21 "Out of range"
- Writing value to a reserved indices has no effect and does not return error
- Read access with the reserved indices returns 0.

References

ET[]

PE – Position Error

PE reports the position error.

CANopen/CoE

Elmo Alias Object: 0x322A

Attributes

Attribute	Description
Type	Double, Read-only
Source	All
Restrictions	None
Range	NA
Index range	NA
Default	0
Unit modes	**UM = 5, 7**
Non-Volatile	No
Axis Related	Yes

Remarks

The **PE** command returns the instantaneous position tracking error, in user defined units.

In main feedback position mode (**UM** = 5), **PE** reports the following:

PE = DV[3] – PX

If the absolute value of **PE** exceeds **ER[3]**, the motion is aborted, and the motion fault code **MF** = 256 (0x100) is set. If **MO** = 0, or if the position controller is not used for example in velocity or current or stepper mode (**UM** = 1, 2 or 3), **PE** returns 0.

References

XM[*N*], **ER[*N*]**, **MF**, **UM**

PL[N] – Current Peak Limit

PL[] specifies the peak limit current and peak limit duration.

CANopen/CoE

Elmo Alias Object: 0x3231

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	See below
Index range	1-3
Default	See table below
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

This parameter is used to protect the motor (or the drive) from overcurrent and to protect the load from excessive torque. The motor current command is limited by the continuous (**CL[1]**) and peak (**PL[1]**) current limits. The drive allows the current to be higher than the continuous **CL[1]** limit for a limited time period, after which it is clamped internally to **CL[1]**. The time limit duration depends on a number of factors explained below.

As a general rule, when a step current command is given from **Icmd=0** to **Icmd=PL[1]**, the actual current will be equal to **PL[1]** (amperes) for a limited period of time (defined by **PL[2]** in seconds), after which it will be clamped to **CL[1]**. The motor current command remains limited to **CL[1]** until enough time has elapsed for the average requested current command to fall below 90% of **CL[1]**.

The **LC** flag indicates that the current is limited to its continuous limit.

The torque limits **PL[1]** and **CL[1]** may be changed dynamically while the motor is enabled.

It should be noted that in "R" type drives where (**MC[2]**=**MC[4]**), the continuous current limit **CL[1]** can be set as follows:
0 <=**CL[1]**<= **PL[1]** <= **MC[2]**.

Indices

The following table describes the **PL[*N*]** entries.

Index	Description	Units	Range	Default
1	Defines the motor maximum peak current, in amperes.	Amperes	**0 to MC[2]**	
2	Defines the motor maximum peak duration, in seconds.	Seconds	0.1 to 1000	
3	Dynamic Brake current command, in amperes.	Amperes	**0 to MC[2]**	
4	Reserved			
5	Temperature threshold for peak current limit	Celsius	0 to 120	84
6	Current per temperature factor for peak current limit	Amp/Celsius	0 to **MC[2]**	0
7	Voltage threshold for current derating See notes.	Volts	NA, if > **BV**, saturated to **BV**	85
8	Voltage derating current slope	Amp/Volt	NA, if >**MC[2]**, saturated to **MC[2]**	0

Notes

- It is recommended to define a **PL[1]** value that can be achieved. It is not recommended to set **PL[1]** > **BV/R**, where **BV** is the DC power supply voltage and R is the motor resistance. **PL[1]** value should be small enough so that at peak current there is enough voltage to drive current changes. Otherwise, at large currents the drive's response rate will be limited by voltage saturation and the controller's performance will decrease.
- The allowed peak current may be saturated at a level lower than the **PL[1]** value when the PWM frequency is increased with the **XP[2]** command.
- The peak duration time **PL[2]** specifies the time required to switch from the peak limit to the continuous limit, when current command is stepped from zero to **PL[1]**.
  The user can set the peak duration time as specified in the table above.
  Note however, that the drive internally limits the actual peak time according to the following formula, in order to protect the overall power:

$$\mathbf{PL}[2] \le \frac{(\mathbf{MC}^2[2] - \mathbf{MC}^2[4]) \cdot \mathbf{MC}[3]}{\mathbf{PL}^2[1] - \mathbf{CL}^2[1]}$$

References

CL[], **LC**, **MC**, **TC**, **XP**

PO – Positioning Options

PO specifies:

- The optional behavior of the **PA** and **PR** commands.
  - The stop manager bypass options
  - Interpolation type in Cyclic Synchronous Position Mode

CANopen/CoE

Elmo Alias Object: 0x3234

Positioning_Option_Code: **PO[1]**: 0x60F2

Attributes

Attribute	Description
Type	Unsigned Short, Read/Write
Source	All
Restrictions	**PO[1] - effective on the next call to BG** **PO[2], PO[3] - motor must be off**
Range	According to bit-field definitions
Index range	NA
Default	0
Unit modes	**UM = 5; UM = 6; UM = 7**
Non-Volatile	Yes
Axis Related	Yes

Indices

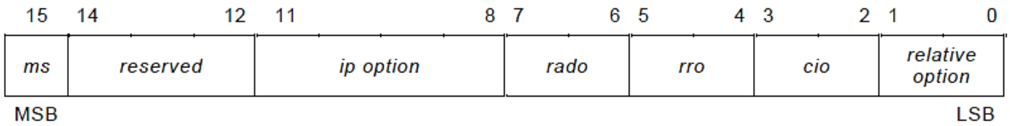
The following table defines the **PO[N]** entries defined by the command index **(N)**.

<table>
<tr><th>Index</th><th colspan="2">Description</th><th>Default</th><th>Values</th><th>Comments</th></tr>
<tr><td>1</td><td colspan="2">Positioning option code as it is defined by the DS-402 object 0x60F2</td><td>0</td><td></td><td>Details see below</td></tr>
<tr><td rowspan="5">2</td><td colspan="2">Cancel Stop Manager features:</td><td rowspan="5">0</td><td rowspan="5">0, 2, 4, 6</td><td rowspan="5">In Interpolated Position Mode and in Cyclic Synchronous Position Mode (DS-402 object 0x6061 is 7 or 8) the acceleration limiting, denoted by SD, is bypassed regardless of bit 1 setting.</td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>Bypass acceleration limiting</td></tr>
<tr><td>2</td><td>Ignore FLS, RLS, HW Stop & SW position limits (VH[]3\VL[3]) in Interpolated Position and all Cyclic</td></tr>
</table>

Index	Description	Default	Values	Comments
	Synchronous modes (DS-402 Object 0x6061 is 7, 8, 9, 10)			
3	Interpolation type in Cyclic Synchronous Position Mode **Value** – **Description** 0 – Linear interpolation 1 – Cubic interpolation 2 – Quadratic interpolation	0	0, 1, 2	Cubic interpolation can be selected, if **Object 0x60B1** (velocity offset) is mapped. Quadratic interpolation can be selected, if **Object 0x60B1** (velocity offset) is **not** mapped.

Remarks

The following diagram shows the bit-field structure of a **PO[1]** value:



LEGEND
ms = manufacturer-specific
rro = request-response option
cio = change immediately option
rado = rotary axis direction option

The *relative option* bits control the behavior of the **PR[1]** command as detailed in the following table:

Value of Bits 0 and 1	Remarks
0x0 (default)	Positioning moves are performed relative to the preceding (internal absolute) target position or relative to the present location if there is no preceding target position (after Motor On).
0x1	Positioning moves are performed relative to the actual position demand value, that is, the output of the trajectory generator.
0x2	**Positioning moves are performed relative to the position actual value (PU).**
0x3	Reserved.

The *change immediately option (cio)* bits are described in the following table:

Value of Bits 2 and 3	Remarks
0x0 (default)	The drive device readapts the actual motion to the new target position immediately.
0x1	The actually performed positioning task will be continued and blended with the newly commanded task when the target position is reached.
0x2	Reserved.
0x3	Reserved.

The *rotary axis direction options (rado)* bits define the behavior of the position modulo:

Mode	Value of Bits 6 and 7	Remarks
Normal positioning	0x0 (default)	Normal rotary positioning is similar to linear axis positioning. If the position range limits (object **0x607B** or **XM[1]** and **XM[2]**) are achieved or exceeded, the input value wraps automatically to the other end of the range. Movement greater than the modulo is possible only with this bit combination.
Negative movement	0x1	Positioning only in the negative direction. If the target position is higher than the actual position, the axis moves over the minimum limit of the position range **(XM[1]** or Object **0x607B.1** to the target position.
Positive movement	0x2	Positioning only in positive direction. If the target position is lower than the actual position, the axis moves over the maximum limit of the position range **(XM[2]** or Object **0x607B.2** to the target position.
Positioning with shortest way	0x3	Positioning with the shortest way to the target position. **Note:** If the difference between the actual value and the target position in a 360° system is 180°, the axis moves in the positive direction.

References

PA, **PR**, **BG**, **XM[]**

PP – Protocol Interface

PP[] programs all communication parameters.

CANopen/CoE

Elmo Alias Object: 0x3235

Mappable objects: None

Attributes

Attribute	Description
Type	Parameter, unsigned long, Read/Write
Source	All
Restrictions	**MO=0 for PP[1]**
Range	See table below
Default	See table below
Index Range	1 to 31
Unit modes	All
Non-Volatile	See table below
Axis Related	No
Attribute	None

Remarks

The **PP[]** command sets and retrieves the communication parameters for RS232, Ethernet, TCP-IP/UDP and CANopen protocols. When modifying a communication parameter, e.g. RS-232 baud-rate, the new variable will affect the communication only when a relevant activate request is received. This means that setting a parameter will not cause a loss of communication.

The following table informs the triggers that allow to modify the communication according to the protocol:

Communication	Request to modify communication
RS232 main	**PP[1]**=1
RS232 auxiliary	**PP[2]**=2, In Titanium - Aux 1
RS232 auxiliary 2	**PP[2]=4 (Titanium drives)**
CAN	**NMT 0x81** (communication reset) **NMT 0x82** (application reset)
Ethernet (TCP-IP/UDP)	PP[1]=3

Notes:

- **For Titanium drives:**
  RS232 Main can be used for Interpreter, SIL Rx\Tx, Sensors Rx (e,g, Gyro)
  RS232 Auxiliary 1 can be used for Interpreter, SIL Rx\Tx, Sensors Rx (e,g, Gyro)
  RS232 Auxiliary 2 can be used for SIL Rx\Tx and Sensors Rx (e,g, Gyro)
- After a complete firmware download, the host sends the **BU** command that resets the drive, by resetting the power stage internally and activating the new firmware image. During this procedure the communication to the drive is lost and the parameters are loaded from the FLASH memory. When the power returns (typically taking a few hundred milliseconds), the host should resume the communication to the drive according to the loaded parameters.
- **PP[1]** setting, the actual communication parameters change will take affect after 1[sec]. The response to the **PP[1]**=x command may be sent with new communication parameters depending on the communication channel load.

Indices

The following table describes the **PP[N]** entries:

Index	Description	Values		Non-Volatile	Restrictions
1	Type of communication	0 OR >3	None	No	
		1	Main RS232		
		2	Aux RS232		
		3	Ethernet		
		4	Aux 2 RS232		Titanium Drives
2	Main RS232 Baud rate	0	4,800	Yes	
		1	9,600		
		2	19,200		
		3	38,400		
		4	57,600		
		5	115,200		
		6	230,400 (default)		
		7	460,800		
		8	921,600		
		9	1,843,200 (1.8 Mbps)		

Index	Description	Values		Non-Volatile	Restrictions
		10	3,686,400 (3.68Mbps)		
3	Reserved	0		No	
4	Main RS232 Parity	0	None (default)	Yes	
		1	Even		
		2	Odd		
5	Reserved	0		No	
6	Aux RS232 Baud rate	0	4,800	Yes	In Titanium Aux 1 RS232
		1	9,600		
		2	19,200		
		3	38,400		
		4	57,600		
		5	115,200		
		6	230,400 (default)		
		7	460,800		
		8	921,600		
		9	1,843,200 (1.8 Mbps)		
		10	3,686,400 (3.68Mbps)		
7	Aux RS232 Parity	0	None (default)	Yes	In Titanium Aux 1 RS232
		1	Even		
		2	Odd		
8	Aux 2 RS232 Baud	Same as Aux 1 baud rate		Yes	Titanium drives Only
9	Aux 2 RS232 Parity	0	None (default)	Yes	Titanium drives Only
		1	Even		
		2	Odd		
10-12	Reserved	0		No	None
13	CANopen Device ID	1 to 127		Yes	Default 127

Index	Description	Values		Non-Volatile	Restrictions
14	CANopen Baud rate	0	1,000,000	Yes	Long values
		1	500, 000 (default)		
		2	250,000		
		3	125,000		
		4	100,000		
		5 to 7	50, 000		
15	CANopen group	1 to 128		Yes	Default 128 Note that 128 disables group functionality
16-22	Reserved	0		No	None
23	EtherCAT OUT port configuration	0	EtherCAT Default	Yes	This parameter is evaluated at power-up only.
		0xA55A	Ethernet		
24	Ethernet – IP	0x00000000 – 0xFFFFFFFF Format Example: For IP 192.168.1.2 set value to 0xC0A80102		Yes	This parameter is affected in case **PP[1]**=3 or after power-up. Default: 192.168.01.02
25	Ethernet – Net Mask	0x00000000 – 0xFFFFFFFF Format Example: For IP 255.255.255.0 set value to 0xFFFFFF00		Yes	This parameter is affected in case **PP[1]**=3 or after power-up. Default: 255.255.255.0
26	Ethernet – Gateway	0x00000000 – 0xFFFFFFFF Format Example: For IP 192.168.1.1 set value to 0xC0A80101		Yes	This parameter effects the baud rate in case **PP[1]**=3 or after power-up. Default: 192.168.01.01
27-31	Reserved			No	

Notes

- The number of RS-232 stop bits has a fixed value of 1.
- The group ID number for CAN (**PP[15]**) defines the ID of the received message object. The response is transmitted by each node with its own ID (**PP[13]**). Setting **PP[15]** = 128 allows the user to cancel the CAN group ID.
- Refer to the **YS** command for the actual Ethernet and EtherCAT EoE addresses.
- Unused **PP[*N*]** parameters are reserved for compatibility with other Elmo drives.

References

YS

PR[N] – Position Relative

PR[N] specifies the relative target positions for the drive profiles.

CANopen/CoE

Emo Alias Object: Not Available

PR[1] Target_position: 0x607A

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	• Motor must be ON (except for **PR[4]**) • Effective on the next call to **BG[N]** • Target position after adding the **PR[1]** value must be within the ranges defined by **VH[3]**, **VL[3]**
Range	-2^{52} to $(2^{52}-1)$
Index range	1 to 4
Default	0
Unit modes	**UM = 3; UM = 5; UM = 6; UM = 7**
Non-Volatile	No
Axis Related	Yes

Remarks

Main profile (N=1):

PR command w/o index is equivalent to PR[1]

PR[1] command behaves according to the relative option bits of the PO command

On the next BG[1] after applying PR[1] command:

The motion control will switch to the position control loop.

The drive will execute the Point-To-Point (PTP) motion according to AC[1], DC , SF, SP[1] and FS.

When PR[1] is set and BG[1] is commanded, the motion mode reflected in object 0x6061 is modified to 1 (Profile Position).

Object 0x607A will be overridden by the target value. Refer to the **BG** command for more details.

Simple (low capability) profiles (N=2,3,4):
On the next BG[N] after applying PR[N] command (N=2,3,4) the corresponding profile will be built according to AC[N] and SP[N]. SF, FS and DC commands are not supported for these profiles.

If UM = 3 or 6, PR[N] determines the target position in electrical angle units (1 pole pair is 360 electrical degrees, which are denoted by 4096 electrical ticks).

The motor will abort if the feedback position is higher than HL[3] or lower than LL[3].

The PR[N] value can be given in user-defined position units specified by the FC command.

Indices

The following table describes the profiles defined by the command index **(N)**.

Index	Profile name	Profile description
1	Main	The main drive profile (DS-402)
2	Superimposed	The profile is used for: • Superimposing the main profiler motion by the additional motion • Overcoming motion jumps when external reference generator is used (follower or ECAM)
3	Phasing	The profile is used during follower and ECAM mode to create a smooth phase shift in the master position of a slave axis.
4	Socket profile	The profile running in a socket at control interrupt, can be used for all functions of the socket.

References

JV, **FS**, **SP**, **AC**, **DC**, **UM**, **PR**, **PO**, **FC**

PS – Get Program Status

PS[N] PS informs the status of the user program.

CANopen/CoE

Elmo_parameters_objects: 0x3238

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	-2147483648 to 2147483647
Default	0
Index Range	1 to 2
Unit modes	All
Non-volatile	No
Attribute	No

Remarks

PS[1] – Get user program status.

PS[2] - Get threads status.

PS[1]:

Drive can be in one of the following situation, in User Program aspect:

- No program was loaded ever or no program exist.
- User program was downloaded. It is not running but ready for execute command.
- User program is halted by HP command.
- User program is currently running.

PS[1] indicates each of the above description. The following table details these indications:

PS[1] value	Description
-2	Program is not compiled or does not exist. In case that program exists but could not be loaded from the non-volatile flash during boot-up (power up or drive reset) this would also be the indication.
-1	Program exists, in rest, and ready to be executed
0	Thread 0 (main) in halt state. This can be either break point (in debug mode) or after **HP** command
1	Program is running. This case would also be indicated in **SR** command bit 12.

PS[2]

PS[2] bits	Description	
0-3	Thread 0 status	
	0 (0x0)	Thread 0 is Halt
	1 (0x1)	Thread 0 is Active
	2 (0x2)	Thread 0 is Idle
4-7	Thread 1 status	
	0 (0x0)	Thread 1 is Halt
	1 (0x1)	Thread 1 is Active
	2 (0x2)	Thread 1 is Idle
8-11	Thread 2 status	
	0 (0x0)	Thread 2 is Halt
	1 (0x1)	Thread 2 is Active
	2 (0x2)	Thread 2 is Idle
12-15	Thread 3 status	
	0 (0x0)	Thread 3 is Halt
	1 (0x1)	Thread 3 is Active
	2 (0x2)	Thread 3 is Idle
16-19	Thread 4 status	
	0 (0x0)	Thread 4 is Halt
	1 (0x1)	Thread 4 is Active
	2 (0x2)	Thread 4 is Idle
20-23	Thread 5 status	
	0 (0x0)	Thread 5 is Halt
	1 (0x1)	Thread 5 is Active
	2 (0x2)	Thread 5 is Idle
24-31	Reserve	

References

CC, HP[], XC[], XC, KL[]

PU – Main Position in User-Defined Units

PU sets/reports the present position of the position feedback socket in user-defined units.

CANopen/CoE

Position_actual_value: 0x6064

Elmo Alias Object: 0x323A

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	None
Range	-2^{52} to $(2^{52} - 1)$
Index range	NA
Default	According to the value of the sensor. An incremental encoder starts counting from zero.
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

Setting the **PU** command acts as an immediate homing to the requested position.

PU cannot be set when:

- sensor socket is not defined via **SA[]**
- outside of position modulo denoted via **XM[1]**….**XM[2]**
- homing procedure is active (**HM[1]**, DS402 homing mode)
- position error mapping is active **(PC[1])**
- Motion Status is active (**MS==2 || MS==1**-, Statusword (0x6041))

References

PX, XM[], FC[]

PW[] – PHY Register Configuration

PW[] specifies the drive option to configure the PHY register value at power-up according to the connected PHY. The command is non-volatile, and saved in FLASH, non-axis related, and supports relevant registers only.

CANopen/CoE

Elmo Alias Object: 0x323C

Attributes

Attribute	Description
Type	Unsigned short, Read/Write
Source	All
Restrictions	AF[2] bit 3 set to 1
Range	See indices table below
Index range	1, 2
Default	**PW**[1] = 0xB, **PW**[2]=0x0003
Unit modes	All
Non-volatile	Yes
Axis related	No

Remarks

- Supported for PHY type reported by **WS**[68] as 2.
- Activated only in case **AF**[2] bit 3 is set, otherwise command is ignored

Indices

The following table describes the available options for PW[].

Index	Description	Type	Values	Restrictions
1	Register address	Unsigned short	0x000B	Read/Write
2	Register value	Unsigned short	0, 2, 3, 8	Read/Write

References

AF[], WS[]

PX – Main Position in Counts

PX sets/reports the present position of the position feedback socket in internal units (counts).

CANopen/CoE

Elmo Alias Object: 0x323D

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	None
Range	-2^{52} to $(2^{52} - 1)$
Default	According to the value of the sensor. An incremental encoder starts counting from zero.
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

PX command acts as an immediate homing to the specified value.

PX cannot be set when:

- socket is not defined via **SA[]**
- outside of position modulo denoted via **XM[1]….XM[2]**
- homing procedure is active (H**M[1]**, DS402 homing mode)
- position error mapping is active (**PC[1]**)
- Motion Status is active (**MS==2 || MS==1**-, Statusword (0x6041))

References

PU, XM[]

PZ – Additional Position in Counts

PZ sets/reports the present external reference position in user-defined units.

CANopen/CoE

Elmo Alias Object: 0x323F

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	None
Range	-2^{52} to $(2^{52} - 1)$
Default	According to the value of the external reference source – socket or axis profiler.
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

- External reference source is defined by **EM[16]** command

References

EM[]

QS – Quick Stop Deceleration

QS specifies the profile deceleration used to stop the main DS-402 profile when the quick stop function is activated.

CANopen/CoE

QS quick stop deceleration: 0x6085

Elmo Alias Object: 0x325C

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Range	1…10^{14}
Index range	1
Default	1,000,000,000
Unit modes	**UM = 2, 3, 5, 6, 7**
Non-Volatile	Yes
Axis Related	Yes

Remarks

The QS command specifies the configured deceleration used to stop the motor when the quick stop function is activated and the quick stop code **Object 0x605A** is set to 2 or 6. The quick stop deceleration is also used if the fault reaction code **Object 0x605E** is 2 and the halt option code **Object** 0x**605D** is 2.

The quick stop deceleration of the main profiler are subject to the limits of the **SD** value. If the **QS** value is higher than the **SD** value, the **SD** value is used, and the **QS** value is ignored.

The **QS** value can be given in user-defined units specified by the **FC** command.

References

AC, **SD**, **BG**, **PA**, **PR**, **JV**, **FC**

R1 – SIL User Parameters

R1 Array is a SIL general purpose array.

CANopen/CoE

0x3265, 0x22F3

Attributes

Attribute	Description
Type	long (signed 32bit)
Source	All
Restrictions	No
Range	Not relevant
Index range	254
Default	0
Unit modes	All
Non-volatile	Yes
Axis related	No

Remarks

This SIL user array can be interfaced from the following:

- ELMO communication interfaces
- SIL program API
- EtherCAT/CAN CoE (0x3265/0x22F3)
- EtherCAT/CAN PDO (0x22F3)

At power-up the array is in default value, i.e. all sub index's are 0.

R1 array can be saved, once at any time after power-up cycle, by **SV[8]** command OR by SIL API.

R1 array can be load from FLASH by **LD[8]** command OR by SIL API.

Indices

None

References

LD, SV.

R2 – SIL User Parameters

R2 Array is a SIL general purpose array.

CANopen/CoE

0x3266, 0x22F4

Attributes

Attribute	Description
Type	double
Source	All
Restrictions	No
Range	Not relevant
Index range	128
Default	0
Unit modes	All
Non-volatile	Yes
Axis related	No

Remarks

This SIL user array can be interfaced from the following:

- ELMO communication interfaces
- SIL program API
- EtherCAT/CAN CoE (0x3266/0x22F4)
- EtherCAT/CAN PDO (0x22F4)

At power-up the array is in default value, i.e. all sub index's are 0.0.

R2 array can be saved, once at any time after power-up cycle, by **SV[9]** command OR by SIL API.

R2 array can be load from FLASH by **LD[9]** command OR by SIL API.

Indices

None

References

LD, SV.

R3 – SIL User Parameters

R3 Array is a SIL general purpose array.

CANopen/CoE

0x3267

Attributes

Attribute	Description
Type	double
Source	All
Restrictions	No
Range	Not relevant
Index range	128
Default	0
Unit modes	All
Non-volatile	Yes
Axis related	No

Remarks

This SIL user array can be interfaced from the following:

- ELMO communication interfaces
- SIL program API

At power-up the array is in default value, i.e. all sub index's are 0.0.

R3 array can be saved, once at any time after power-up cycle, by **SV[10]** command OR by SIL API.

R3 array can be load from FLASH by **LD[10]** command OR by SIL API.

Indices

None

References

LD, SV.

R4 – SIL User Parameters

R4 Array is a SIL general purpose array.

CANopen/CoE

0x3268

Attributes

Attribute	Description
Type	double
Source	All
Restrictions	No
Range	Not relevant
Index range	128
Default	0
Unit modes	All
Non-volatile	Yes
Axis related	No

Remarks

This SIL user array can be interfaced from the following:

- ELMO communication interfaces
- SIL program API

At power-up the array is in default value, i.e. all sub index's are 0.0.

R4 array can be saved, once at any time after power-up cycle, by **SV[11]** command OR by SIL API.

R4 array can be load from FLASH by **LD[11]** command OR by SIL API.

Indices

None

References

LD, SV.

RA[] – Recorder Actual Length

RA[*N*] array reflects recording length related data.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Integer/read only
Source	Program, all except CoE
Restrictions	None
Range	long
Index range	1 to 3
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

Index	Description	Type	Values	Restrictions
1	**Actual recorded length in samples (same as WI[22]in Gold).** In case real-time recorder mode, this value is the size of each buffer.	Integer	**Buffer mode:** **0 - RA[2]** **Real-time mode:** **0 - RA[2]/2**	Valid only if recorder is in buffer or Real-time modes. For buffer mode: check that recorder is active or finished but there is valid data inside. For Real-Time mode: check only that recorder is active. (otherwise returns error : “Recorder not active”)
2	Max recorder buffer length in Bytes, supported by the drive.	Integer	$1 – 5 \times 10^6$	The recorder buffer length supported by the drive.
3	Recorder current sample counter	Integer	0 - RA[1]	Valid only if recorder is active in buffer mode

References

RC, BH, RR

RC – Define Recorder Variables

RC specifies which of the mapped signals are to be recorded.

CANopen/CoE

Elmo Alias Object: 0x3270

Attributes

Attribute	Description
Type	Integer, Read/Write
Source	All, except CoE
Restrictions	**Recorder inactive (RR=0 or RR=-1)**
Range	Bit field [bits 0 to 32]
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The drive can record a range of signals for performance verification and debugging. The first step of the recording process is to define the recorded variables by assigning a value to **RC** (a bit field). Each activated bit in the representation of **RC** defines a signal to be recorded. A valid **RC** value defines at least one recorded variable. **RC** can map up to thirty-two variables that are to be recorded.

The host can map many optional variables to any bit of **RC** from bit 0 to bit 32.

If the drive has stored previously recorded data, setting **RC** will invalidate this data. Invalidated data cannot be retrieved.

The total number of data points that can be recorded is fixed. Therefore, the number of points per signal depends on the number of signals that are recorded simultaneously: the more signals recorded, the fewer are the points that are available for each signal.

References

RG, **RL**, **RP[*N*]**, **RR**, **BH**

RF[] – Recorder Factors

RF[*N*] indicates the factor of the selected **RV[N]** recorded variables.

CANopen/CoE

Elmo Alias Object: 0x3273

Attributes

Attribute	Description
Type	Double
Source	Program, all except CoE
Restrictions	**RV[i] must be selected.**
Range	**According to the selected RV[i]**
Index range	1 to 16
Default	1 to 16, respectively
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

Indices

Index	Description	Type	Values	Restrictions
1 to 16	Recorder variable entry factor	Double	1 to 16, respectively	**RV[i] must be selected**

References

RC, **BH**, **RR, RV**

RG – Recorder Gap

RG specifies the frequency per sampling time that the recorder is activated.

CANopen/CoE

Elmo Alias Object: NA

Attributes

Attribute	Description
Type	Integer, Read/Write
Source	All, except CoE
Restrictions	**The recorder must be inactive (RR = 0 or RR = -1).**
Range	1 to 32767
Default	1
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

As the recorder has a limited storage capacity (16K), if it operates at the sampling time of the drive, the recorder will operate for a very short time. To achieve longer recording times, the time interval between consecutive data recordings must be increased. The **RG** parameter trades recording resolution for increased recording time. When **RG** = 1, the sampling time of the recorder is given by the **WS[29]** command.

Be aware that the recorder sampling time depends on the **TS/HS** value and the specific unit mode (**UM**).

On the conditions when:

- **RP[4]** = 0, the sampling time depends on **TS** value
- **RP[4]** != 0, the sampling time depends on **HS** value.

If the drive has stored a previously recorded data vector, setting **RG** will invalidate this data. Invalidated data cannot be retrieved.

References

RC, **RL**, **RP[*N*]**, **RR**, **BH**, **TS**, **WS[29]**, **UM**

RH[] – Recorder Header

RH[] uploads the recorder header

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Command, write-only
Source	All, except FoE
Restrictions	• Valid recorded data is ready (see **RR** command). **No other uploading sequence is performed (UL).**
Range	Set value of type unsigned char (1 byte) The command uploads header data.
Index range	1
Default	No
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The recorder allows the host to capture data of signals such as Position and Velocity. This information is uploaded by the host for any required analyzes. The recorder header contains all the relevant information of the signals such as size, type of signal, factors etc.

The **RH** command contains the recorder header. The header is built during the **RR** command (start recording). The data in the **RH** command as well as the data in the recorder buffer are available to upload until the next **RR** command is sent.

Please refer to **RR** command for further information about the recorder procedure.

The **RH** command is designed to optimize data transfer from the drive to the host while allowing fetching of the data in a simple Binary format or Hex-Binary text format.

Regardless of the recorder mode (buffered/Real-Time), the **RH** command supports the following upload formats:

<table>
<tr><th>RH index</th><th>Value to Set</th><th colspan="2">Description/Format</th></tr>
<tr><td rowspan="7">1</td><td rowspan="6">RH[1] = 1</td><td colspan="2">Recorder header upload data format is according to communication type:</td></tr>
<tr><td>Communication</td><td>Data Format</td></tr>
<tr><td>RS232</td><td>Hex-Binary</td></tr>
<tr><td>UDP</td><td>Hex-Binary</td></tr>
<tr><td>USB</td><td>Hex-Binary</td></tr>
<tr><td>CAN</td><td>Binary</td></tr>
<tr><td>RH[1] =0</td><td colspan="2">Recording header data will be uploaded in Binary format.</td></tr>
</table>

The Header Structure

Note that the number of bytes which are uploaded depend on the uploaded method which is selected via **RH[1]**. The table assumes that the uploaded method is binary (**RH[1]**=0). If the Hex-Binary format is selected, all bytes are doubled.

> **Note:**
> Hex-Binary format means that each number (byte) is transferred in two printable characters. For example, the value 10 (0x0A) is transmitted in two chars: 0x30 and 0x41. This means that in Hex-Binary format the total byte size is doubled relative to the binary format.

The recorder header information is described in the table below.

Name	Description	Bytes
Length of RH	Length of all data in RH buffer to upload.	0 - 3
Checksum	CheckSum 16 bit on all data from byte 12 until the end.	4 - 5
Record length	**Value of RA[1] command – actual length of recorded data.**	6 - 9
Gap value	**Value of RG command – recording gap.**	10 - 11
Sampling time	**Value of TS command – TS time.**	12 - 15
High speed time	**Value of HS command – HS time.**	16 - 17
Recorder signals	**Value of RC[1] command – defined recorded variables (bit field).**	18 - 21
Recorder map	**RV[] array – each cell of this array is 32 bit field (contains the variable's index in SigType and its axis/socket number).** **Note:** All cells of RV array will be uploaded here, where EAS should use the relevant cell that suit the RC command.	22 – 85

Name	Description	Bytes
	The length of RV array today is 16, which may change to 32 at a future time.	
Recorder factor	**All RF[] array – contains values of factors (type 'double') per each corresponding cell in RV.** **Note:** All cells of RV array will be uploaded here, where EAS should use the relevant cell that suit the RC command. The length of RV array today is 16, which may change to 32 at a future time.	86 – 213
Recorder & trigger parameters	**All RP[] array – contains recording parameters.** **Only the following cell's indexes of RP[] array will be uploaded:** **RP[1] – First Trigger index** **RP[2] – Pre-trigger percentage** **RP[3] – Trigger type** **RP[4] – TS/HS flag** **RP[6] – First trigger - digital input level** **RP[7] – First trigger – digital input mask** **RP[10] – Time value [msec] for start of recording.** **RP[16] – Second trigger index.** **RP[19] – Second trigger - digital input level** **RP[20] - Second trigger – digital input mask**	214 – 253
Trigger level (float)	**All FT[] array – contains floating point trigger's level (type double).**	254 – 285

All recording modes

During the uploading of the data, the drive can receive and execute other commands. However, the drive will not be able to reply the commands unless it is used in a communication channel that is different from the channel used by the **RH** request. For example, if the recorded data is uploaded from the RS232 communication channel, the USB communication channel can still be used to execute motion commands while the RS232 channel is uploading the data.

During the uploading the following commands cannot be executed:

- **PP[1]** command for engaging new communication parameters
- **FT[]**,**RC**, **RG**, **RL**,**RP[]**,**RR** and **RV[]** commands are used during the recorder setting.

The recorder can fetch global variables from the User Program as well. See the **RR** command and the User Program manual for further information.

References

FT[], **RC**, **RG**, **RL**, **RP[]**, **RR**, **RV[]**

RL – Recorder Length

RL specifies the total length of the recorded data.

CANopen/CoE

Elmo Alias Object: 0x3279

Attributes

Attribute	Description
Type	Integer, Read/Write
Source	All, except CoE
Restrictions	**The recorder must be inactive (RR = 0 or RR = -1).**
Range	**1 to RA[2]**
Default	**RA[2]**
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

If the recorder is set with a length larger than the maximum value (**RA[2]**), each signal will be set according to the formula

RL = RA[2]/ $\sum_{1}^{n}$ (**recorded variable size in bytes**).

The actual size of the recorded data is returned by the **WI[21]** or **RA[1]** commands.

If the drive has stored a previously recorded data vector, setting **RL** will invalidate this data. Invalidated data cannot be retrieved

Define the recording length for each recorded variable. The max allowed **RL** will be available for the recorder user (EAS) via **RA[]** command.

RL value depends on the recording mode:

- Buffer recording mode – **RL** defines the total recording length by the following equation: (Recording length) = (num recording variables) * **RL**
- Real-time recording mode – **RL** defines the length of a single buffer by the following equation: (Total Recording length) = (num recording variables) * **RL**.
  The maximum allowed recording length in this mode is half of the max possible recording length **RA[2]**.

References

RC, RG, RP[N], RR, BH

RM – Reference Mode

RM specifies the use of an external reference generator (ECAM/Follower).

CANopen/CoE

Elmo Alias Object: 0x327A

Attributes

Attribute	Description
Type	Short, Read/Write
Source	All, except CoE
Restrictions	None
Range	0, 1
Index range	NA
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

The following table describes the possible values of the **RM** command.

Command Value	Description
0	External reference generator is disabled.
1	External reference generator is enabled.

Notes

- When setting **RM=1**, the ECAM/Follower position output jump is prevented. The Velocity output can be changed in steps, e.g. in Follower mode or when the ECAM is enabled in the middle of the table.
- After setting **RM=0**, the ECAM/Follower position output maintains the latest value, with the velocity and acceleration outputs set to 0.
- Setting **RM**=1 resets ECAM LPF and FIR filters. For filters details refer to **EM[N]** and **KV[N]** commands.

References

IL, **MO**, **KV[N]**

RP – Recorder Parameters

RP[] determines the trigger for the recorder and the method by which the recorded data is transferred to the host.

CANopen/CoE

Elmo Alias Object: 0x327D

Attributes

Attribute	Description
Type	Parameter, Long
Source	All, except CoE
Restrictions	• According to the description table • The recorder must be inactive (**RR** = 0 or **RR** = -1).
Range	According to description table
Index range	0 to 20
Default	According to the description table
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

Trigger definitions

The recorder is started by a trigger event, which may be one of the following:

Trigger	Description
Immediate	Recorder starts immediately after the recording request is issued
Triggered by an analog signal	Recorder starts upon one of the following events
Positive slope	Signal crosses a prescribed level with a positive slope
Negative slope	Signal crosses a prescribed level with a negative slope
Window	Signal exits a window of two prescribed signal levels
Digital inputs	Digital inputs are switched to their active logic state as defined by the **IL[N]** command
Motion begins	A **BG** command, or activation of a hardware **BG** command (refer to the **IL[N]** command)
Time	TBD
On the fly	The recorder begins to record immediately after the recording request is issued and uploads data until it is stopped by request

Trigger delay

The trigger defines the event when the recorder starts buffering the data. For example, when the Actual Position reaches 10,000 counts or when the Velocity is either higher than 5,000 cnt/sec or lower than -3000 cnt/sec. The trigger delay indicates the history that the recorder buffer shall store prior to the required trigger event. The delay is given in percentage from the actual buffer size (refer to **RL** command).

Indices

The following table describes the **RP[*N*]** entries.

Index	Description	Type	Values			Restrictions
0	Not in use					
1	First trigger variable index, which is defined similarly to **RV[].** The trigger variable does not need to be one of the recorded variables.	Integer Bit field	Bits 0:15	Index of first recorder variable in the 'SigType' table.		
			Bits 16:31	Number of axis the recorder variable of first trigger is referring to.		
2	Pre-trigger, the percentage of the recorded signal that is recorded before the trigger event [%]	Integer	0 to 100			
3	Trigger type	Integer	Bits 0-7 Trig1	0	Immediate	
				1	Motion Begin	
				2	Positive slope	
				3	Negative slope	
				4	WindowExit	
				5	Greater	
				6	Smaller	
				7	Digital input Change (Raise/Fall)	
				8	Reserved	
				9	WindowEnter	
				10	WindowIn	
				11	WindowOut	
				12	Digital Input level (High/Low)	

Index	Description	Type	Values			Restrictions
			Bits 8-15 Trig2	0	Immediate	
				1	Motion Begin	
				2	Positive slope	
				3	Negative slope	
				4	WindowExit	
				5	Greater (From version 2.0.3.2 B00)	
				6	Smaller (From version 2.0.3.2 B00)	
				7	Digital input Change (Raise/Fall)	
				8	Reserved	
				9	WindowEnter	
				10	WindowIn	
				11	WindowOut	
				12	Digital Input level (High/Low)	
			Bits 16-31	0	No Operation Trigger	
				1	Operation And, for second trigger selection **see RP[16-20]**	
				2	Operation OR, for second trigger selection **see RP[16-20]**	
				3	Operation Chain, for second trigger selection **see RP[16-20]**	
4	Hs/Ts Flag	Integer	0 – The recording occurs every Ts sample. 1- The recording occurs every Hs.			
5	Reserved	Integer				

<table>
<tr><th>Index</th><th>Description</th><th>Type</th><th colspan="2">Values</th><th>Restrictions</th></tr>
<tr><td>6</td><td>Level of digital input when used as trigger for the recorder</td><td>Integer</td><td colspan="2">1 to 0xFFFFFFFF</td><td></td></tr>
<tr><td>7</td><td>Digital input mask, defines which digital inputs trigger the recorder</td><td>Integer</td><td colspan="2">1 to 0xFFFFFFFF</td><td></td></tr>
<tr><td>8</td><td>Lower buffer index for recorded data upload transmission</td><td>Integer</td><td colspan="2">0 to RA[1]</td><td rowspan="2">When RP[9] = RP[8] = 0, all of the buffer is transmitted.</td></tr>
<tr><td>9</td><td>Higher buffer index for recorded data upload trans-mission.</td><td>Integer</td><td colspan="2">0 to RA[1]</td></tr>
<tr><td>10</td><td>Reserved</td><td>Integer</td><td colspan="2">Reserved</td><td></td></tr>
<tr><td>11</td><td>Selected recorded signal for the BS[] command. Defined similarly to RC, but only 1 bit may be non-zero.</td><td>Integer</td><td colspan="2">0 to 65535</td><td>Recorder ready (RR==0); Selected signal is recorded.</td></tr>
<tr><td>12-15</td><td>Reserved</td><td></td><td colspan="2">Reserved</td><td></td></tr>
<tr><td rowspan="2">16</td><td rowspan="2">Second trigger variable index, which is defined similarly to RV[]. The trigger variable does not need to be one of the recorded variables.</td><td rowspan="2">Integer
Bit field</td><td>Bits 0:15</td><td>Index of second recorder variable in the ‘SigType’ table.</td><td rowspan="2"></td></tr>
<tr><td>Bits 16:31</td><td>Number of axis the recorder variable of second trigger is referring to.</td></tr>
<tr><td>17</td><td>Reserved</td><td>Integer</td><td colspan="2">Reserved</td><td></td></tr>
<tr><td>18</td><td>Reserved</td><td>Integer</td><td colspan="2">Reserved</td><td></td></tr>
<tr><td>19</td><td>Second trigger Level of digital input when used as trigger for the recorder</td><td>Integer</td><td colspan="2">1 to 0xFFFFFFFF</td><td></td></tr>
<tr><td>20</td><td>Digital input mask, defines which digital inputs trigger the recorder</td><td>Integer</td><td colspan="2">1 to 0xFFFFFFFF</td><td></td></tr>
</table>

Notes

- If the drive has stored a previously recorded data vector, setting **RP[*N*]** (with *N* equal to a number other than 8 or 9) will invalidate this data. Invalidated data cannot be retrieved.
- When the recorder trigger is set to Digital input **(RP[3]=5)** the trigger is armed when at least one of the masked digital input indication (IP command) marked by **RP[7]** was changed and the value of masked inputs are equal to **RP[6]: (IP&RP[7]) == RP[6])**.

- In case of operation trigger (OR/AND/CHAIN) , there is no meaning by setting trigger '0' (Immediate) as one of the triggers (first or second). Hence 'Immediate' trigger can be set only as Func1 (first trigger) and not as func2 or func3).
- Multi-axis: trigger variables (**RP[1]**, **RP[16]**) are axis related .
- In Buffer mode (refer to **RR** command), setting R**P[8]** & **RP[9]** to 0 indicates that the recorder buffer shall be uploaded in a single uploading sequence. Refer to **BH** command.
  In real time mode (refer to **RR** command) where 2 buffers are used: set **RP[8]** & **RP[9]** to any value between 0 to (**RL**-1) and for buffer 2 uploading, the range is from **RL** to (2***RL**-1).

References

RC, **RG**, **RL**, **RR**, **BH**, **BS[]**

RR[] – Recorder Activate/Status

RR starts the recorder, terminates an on-going recording process and retrieves the recorder status.

CANopen/CoE

Elmo Alias Object: 0x327F

Attributes

Attribute	Description
Type	Parameter, Integer
Source	All, except CoE and Program
Restrictions	The recorder must be inactive (**RR** = 0 or **RR** = -1)
Range	0 to 15
Index range	NA
Default	-1
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The **RR** command has the following options:

Value	Description
-1	Invalidate the recorder (all modes).
0	Stop the recorder (all modes)
1	Start recording upon the next **BG** command (buffer mode)
2	Start recording immediately (Buffer mode)
3	Arm the recorder with the trigger setting of the **RP[3]** command (buffer mode)
4-9	Reserved
10	Real-time recording auto mode start
11	Real-time recorder normal mode start
12	Real-time recorder single mode start
13-14	Reserved
15	Recording cyclic mode start

Note:
Setting a value to the RR invalidates the previous recorded buffer and starts a new recording session.
Setting 0 to RR, previously set to 0 (i.e. data ready) is ignored.

The integrity of all recorder parameters is checked when the recorder starts. The following table details the results of the check. If any of the following fail, the **RR** will be set to -1 and will abort with an appropriate error code (**EC**):

Request	Description
Ongoing recorder	Make sure that **RR** is either 0 or 1.
Number of recorded variables	Make sure that no more than TBD variables were selected.
Total record length	Make sure that the total required length does not exceed the max defined **RL (RA[xx]**??)
Recorder buffer	Make sure that the recorder buffer is available

The **RR** command may report the following values in buffered mode:

Value	Description
-1	There is no valid data in the recorder.
0	The recorder is ready or has finished and is ready with valid data (buffer recording mode).
1 to 3	The recorder is waiting for completion of the trigger event, respectively (buffer mode).

The **RR** command reports the following values in Real-Time mode:

Value	Description
-1	There is no valid data in the recorder. Recorder is not active.
0	Recorder is active data is not ready.
1	Recorder is active, buffer-1 data is ready for upload, samples **0-(RL-1)**.
2	Recorder is active, buffer-2 data is ready for upload, samples **RL-((2*RL)-1)**.
3	Recorder is active, both buffers are ready for upload.

Notes

The recorder buffer is shared with **UL** command that uploads data from the drive. When **UL** is used the **RR** is set automatically to -1.

References

BH, **RP[*N*]**, **RC**, **RG**, **RL**, **FT[N]**

RS[] – Reset Parameters

RS command sets data to the default values.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Not relevant
Source	All
Restrictions	**MO**=0 - Motor disabled (servo off) No user program active (**PS[]**) SIL is not active (**SR[2]**)
Range	Not relevant
Index range	3
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

Reset command sets the default value to all parameters.

If an error occurred during the reset process, the returned error is : 'BAD_DATABASE'. The **CD[1]** command retrieves the details of the error which contains object index, sub-index, **EC** of the particular error. Refer to the **CD** command for details.

Note:

- RS command stops Homing and Touch probe, if active
- After RS command was applied DS-402 Status word bits 10-13 are not relevant

Indices

The following table describes the **RS[*N*]** entries.

Index	Description Format
1	Reset non safety parameters to default value. Object 0x1011 can be used via fieldbus master.
2	Reset safety parameters to default value. When applied the safety password is reset as well. Setting of the safety password is performed via 0x2E01 object.

Index	Description Format
3	Reset SIL model parameters

References

LD, SV, **CD**

RV[] – Recorder Variables

RV[*N*] maps recorded variables to the recorder through the **RC** command.

CANopen/CoE

Elmo Alias Object: **0x3283**

Attributes

Attribute	Description
Type	Integer, Bit field
Source	Program, all except CoE
Restrictions	The recorder must be inactive.
Range	Lower 16bits: According to the index in the static variable table. Upper 16bits: 1 to actual number of axes.
Index range	1 to 32
Default	Lower 16bits: 1 to 16, respectively Upper 16bits: 0 in all cells. (This way the command does not change)
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

RV[*N*] maps recorded variables to the recorder through the **RC** command. By setting **RV[*N*]** = *X*, the variable is assigned to bit *N* - 1 of **RC** in the variable static table. The default mapping (Power On) of **RV[1]** to **RV[32]** behaves similarly to those in previous product lines. The full list of variables available to the recorder is stored in the serial flash memory of the Platinum drive and can be uploaded using the **UL** command.

Each cell in the **RV** array (a long 32bits) will become a bit field divided in two parts:

Lower 16bits (0:15) : will represent the index of the recording variable in the 'SigType' table.

Upper 16bits (16:31) : will represent the number of axis/socket to which the recording variable (from the lower 16 bits) is applied.

Indices

Index	Description	Type	Values		Restrictions
1 to 32	Recorder variable entry	Integer	Bits 0:15	Index of recorder variable in the 'SigType' table. Default values : 1 to 16, respectively	None

Index	Description	Type	Values		Restrictions
			Bits 16:31	Number of axis/socket the recorder variable is referring to. Default: 0 in all cells.	

References

RC, **BH**, **RR**, **RF**

RZ[] – SIL Operational Mode

The **RZ[]** is an interface for activating the SIL.

The Software In the Loop (SIL) is an embedded user program that is based on MATLAB Simulink® (Mathworks Embedded Coder®), downloaded to the drive and run at the drive deterministic real time context (high speed cycle).

CANopen/CoE

Alias Objects: **Object 0x3287**

Attributes

Attribute	Description
Type	Unsigned short, Read/Write
Source	All
Restrictions	None
Range	See table below
Index range	1 - 8
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The SIL is loaded to the drive memory only after power up cycle. If for any reason the SIL is stopped reload is required.

SR[3] bits 16-23 informs the SIL state.

The ability to run SIL depends on drive module.

Indices

The following table describes the **RZ[]** entries.

Index	Description	Value \ description	Notes
1	SIL operation mode. This mode will be activated when **RZ[2]** is set to 0.	165 (0xA5): Execute the SIL immediately.	
		187 (0xBB): Stop running SIL.	Re-enabling the SIL only after next power up.
		57005 (0xDEAD): Erase the SIL from non-volatile memory.	

Index	Description	Value \ description	Notes
2	SIL activation	0: SIL starts when **RZ**[1] is set to 165.	
		1-16: SIL is activated when the digital inputs set by **RZ**[2] are logically set.	Refer to **IL[]** and **IP** commands for more information
		165 (0xA5): Auto mode, SIL is executed immediately on power up.	
		*other values are reserved w/o error.	
3	Time delay to execution	0-65535: The execution of the SIL will be delayed in **RZ**[3] [mSec] after power up cycle completed.	
4	SIL to Drive data :copy mode	0: Copy the SIL date to drive prior to SIL execution. 1: Copy the SIL data to drive after SIL execution.	
8	SIL watchdog timeout. (endless lop protection)	The **HS** (high speed) cycles that are expected to complete SIL cycle. If the SIL will not complete with in the defined number of HS cycles, the SIL will be erased from the flash. Motor is disabled.	

References

SR

S1…S8[]/F1….F8[] – Sensor Socket Parameters

S1…8[]/F1….8[] sets the sensor feedback configuration.

S1…8[] - Sensor socket parameters (integer), are the long type sensor socket parameters.

F1….8[] - Sensor socket parameters (float), are the floating point sensor socket parameters.

CANopen/CoE

Alias Object Ranges: **S1…8[]** 0x3289 - 0x3290

F1….8[] 0x30B5- 0x30BC

Attributes

Attribute	Description
Type	**S1…8**: Long **F1…8**: Float
Source	All
Restrictions	According to array index
Range	According to array index
Index range	**S1…8**: 1 to 20 **F1…8**: 1 to 10
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

It should be noted that modification of the command should be performed using the EAS Quick or Expert Tuning interface. Manual modification may result in ambiguous results from the sensor or function that uses it.

The socket is the virtual entity that contains the sensor parameters and performs all the functionalities which are required from a sensor. There are up to 8 sockets.

Each sensor is provided with a unique ID, which is defined by **S1…8[1]** command. The following short table lists the **S1...8[]/F1...8[]** entries according to the selected sensor ID with links to the main indices table.

Sensor ID S1…8[1]	Sensor type
1	Encoder Quad
3	**Analog Sin/Cos**

Sensor ID S1…8[1]	Sensor type
4	Halls Only
5	Serial BiSS
6	Serial Panasonic
8	Virtual Sine Signal
9	Serial EnDAT
11	Encoder Pulse and Direction
12	Encoder Up/Down
15	Copy Main Profile
16	Analog Input
18	Serial SSI
19	Yaskawa
20	Gantry 4X
21	Serial Exclusive
22	Analog Resolver
25	Serial Sanyo/Nikon
26	Socket Profile
28	Hiperface
29	General Reference 0x3605
30	Speed Observer Simplified
34	Motor Simulation
36	BEMF Observer
38	Generic Serial IMU/Gyro
39	Mitsubishi
40	Acurolink
41	Openlink
42	DSL
43	Enhanced Serial IMU/Gyro Interface

Sensor ID S1…8[1]	Sensor type
44	SIL Sensor

Depending on the sensor type (and the ID), the sub-indices of each command (e.g., **F1[2]**) can have a different functionality. The following table describes the parameters with respect to the ID.

Indices

The following main indices table details the **S1...8[]/F1...8[]** entries according to the selected sensor ID.

Note that for some Titanium Drive sensors, the terminal name e.g. (Eqep0, 1, 2,…) is also used to describe the subindex and terminal.

Encoder Quad

S1…8[1] ID	Index	Description		Values		Restrictions / Notes
1 Encoder Quad	**S1[2]…8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				0	Port A, Secondary Quad 1	
				5	Emulation, Secondary Quad 1	
				9	Port B, Secondary Quad 2	
				10	Port C, Secondary Quad 2	
				11	Single-ended In 7&8, Secondary Quad 2	
				12	Single-ended GPIO 6&7, Secondary Quad 2	
				13	Emulation, Secondary Quad 2	
				17	Port B, Primary Quad 2	
				24	Port A, Primary Quad 1 & Secondary Quad 1	
				25	Port A, Primary Quad 1	

S1...8[1] ID	Index	Description		Values		Restrictions / Notes
				33	Port B, Primary Quad 2 & Secondary Quad 2	
				Quad Drives x 4		
				0	Port A1, Ax1 Secondary Quad 1	
				5	Emulation Secondary Quad 1 A1	
				6	Emulation Secondary Quad 1 A2	
				7	Emulation Secondary Quad 1 A3	
				8	Emulation Secondary Quad 1 A4	
				9	Port B1, Ax1 Secondary Quad 2	
				18	Port A2, Ax2 Secondary Quad 1	
				27	Port B2, Ax2 Secondary Quad 2	
				36	Port A3, Ax3 Secondary Quad 1	
				45	Port B3, Ax3 Secondary Quad 2	
				54	Port A4, Ax4 Secondary Quad 1	
				63	Port B4, Ax4 Secondary Quad 2	
				Titanium Drives 2X		
				0	X1. Port A1 Secondary Quad 1	
				5	X1. Port A1 Secondary Emulation Quad 1	
				6	X2. Port A2 Secondary Emulation Quad 1	
				9	X1. Port B1 Secondary Quad 2 (Eqep0)	

S1...8[1] ID	Index	Description		Values		Restrictions / Notes
				10	X2. Port A2 Secondary Quad 1	
				11	X2. Port B2 Secondary Quad 2 (Eqep2)	
				12	X1. Port A1 Secondary Quad 3	Reserved
				13	X1. Port B1 Emulation Quad 2 (Eqep0)	
				14	X2. Port B2 Emulation Quad 2 (Eqep2)	
		7	Speed calculation by:	0	Every edge	
				1	Every Channel A rising edge	
		8	Quad Interpreter	0	No Quad interpolation	
				1	Quad 10 bit interpolation	
		9-14	Reserved			
		15	Only safety use	0 – No, 1 - Yes		If yes, no RT read
	S1...8[3]	FIR filter window size		0 to 8		0 or 1 disable filter
	S1...8[4]	Sensor status (error) mask				
	S1...8[5]	Glitch filter [nSec]		40 to 5140		
	F1...8[1]	1/T calculation threshold [cnt/sec]		≥ 0		

Analog Sin/Cos

S1...8[1] ID	Index	Description		Values		Notes
3 Analog Sin/Cos	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				0	Port B, Secondary Quad 2	
				8	Port B, Primary Quad 2	
				16	Port B, Primary Quad 2 & Secondary Quad 2	
				Drives x 4		
				0	Port B1, Ax1 Secondary Quad 2	
				8	Port B2, Ax2 Secondary Quad 2	
				16	Port B3, Ax3 Secondary Quad 2	
				24	Port B4, Ax4 & Secondary Quad 2	
		7-8	Reserved			
		9	Software Capture	TBD		
		10-14	Reserved			
		15	Only safety use	0 – No, 1 - Yes		If yes, no RT read
	S1...8[3]	FIR filter window size		0 to 8		0/1 disable filter
	S1...8[4]	Sensor status (error) mask				
	S1...8[5]	Glitch filter [nsec]		40 to 5140		
	S1...8[6]	Multiplication factor		2 to 16		
	S1...8[9]	Sine offset [ADC]		-32768 to 32767		
	S1...8[10]	Cosine offset [ADC]		-32768 to 32767		
	S1...8[11]	Minimum signal amplitude [ADC]		3276 to 32768		

S1...8[1] ID	Index	Description	Values	Notes
	S1...8[12]	Maximum signal amplitude [ADC]	3276 to 32768	
	S1...8[13]	Low signal amplitude warning [ADC]	3276 to 32768	
	S1...8[14]	High signal amplitude warning [ADC]	3276 to 32768	
	S1...8[17]	Absolute position offset	Counts	
	F1...8[1]	Phase shift [deg]	-90 to 90	
	F1...8[2]	Sine gain multiplier	0.3 to 1.7	
	F1...8[3]	Quad only calculation threshold [Hz]	≥ 0	

Halls Only

S1...8[1] ID	Index	Description	Values	Notes
4 Halls Only	None		0	Depends on **H1...8[]** parameters

Serial BiSS

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th colspan="2">Values</th><th>Notes</th></tr>
<tr><td rowspan="10">5
Serial BiSS</td><td rowspan="10">S1...8[2]</td><td colspan="2">Bitwise Configuration Parameter</td><td colspan="2" rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Invert counting direction of sensor</td><td>0</td><td>Do not invert</td><td rowspan="2"></td></tr>
<tr><td>1</td><td>Invert the direction</td></tr>
<tr><td rowspan="6">1-6</td><td rowspan="6">Port and resource used by sensor</td><td colspan="3">Single Drive</td></tr>
<tr><td>0</td><td>Port A, Primary Abs</td><td></td></tr>
<tr><td>8</td><td>Port C, Secondary Abs</td><td></td></tr>
<tr><td>16</td><td>Port A, Primary Abs & Secondary Abs Monitor</td><td></td></tr>
<tr><td colspan="3">Quad Drives x 4</td></tr>
<tr><td>0</td><td>Port A1, Ax1 Secondary Abs</td><td></td></tr>
</table>

S1...8[1] ID	Index	Description		Values		Notes
				8	Port A2, Ax2 Secondary Abs	
				16	Port A3, Ax3 Secondary Abs	
				24	Port A4, Ax4 Secondary Abs	
		7	Sampling rate flag	0 – Every TS 1 – Every 2*TS		
		8	Error and warning bits logic	0 – Active low 1 – Active high		
		9	BiSS Mode	0 – BiSS C Mode 1 – BiSS B mode		
		10	Pure binary resolution flag	0 – Pure binary 1 – Non-binary		
		12	Renishaw compatible flag	0 – General BiSS 1 – Renishaw BiSS		Renishaw BiSS sets all parameters by sensor resolution
		13-14	Reserved			
		15	Only safety use	0 – No, 1 - Yes		If yes, no RT read
		19	Safe Biss	0 – No, 1 - Yes		Determined by safety configuration setting
	S1...8[3]	FIR filter window size		0 to 8		0/1 disable filter
	S1...8[4]	Sensor status (error) mask				
	S1...8[5]	Glitch filter [nSec]		40 to 5140		
	S1...8[6]	Clock Frequency [Hz]		1250000, 2500000, 5000000, 10000000		
	S1...8[7]	Sensor resolution		≥ 8		
	S1...8[8]	Multi-turn resolution		0 - 24		
	S1...8[9]	Protocol bits		14 - 64		Protocol bits include: Position, Status bits and CRC.

S1...8[1] ID	Index	Description	Values	Notes
				excluding: ACK, Start, CDS, Stop.
	S1...8[10]	Reduced bits	0 to 12	
	S1...8[11]	Position LSB bit number	0 to (**S1...8[9]**- **S1...8[7]**- **S1...8[8]**)	Last received bit is 0
	S1...8[12]	Error bit number	-1 to (**S1...8[9]**-1) not including **S1...8[13] and { S1...8[11] - (S1...8[11] + S1...8[7] + S1...8[8])}**	-1 means no error bit
	S1...8[13]	Warning bit number	-1 to (**S1...8[9]**-1) not including **S1...8[12] and { S1...8[11] - (S1...8[11] + S1...8[7] + S1...8[8])}**	-1 means no warning bit
	S1...8[14]	Non binary BiSS resolution 32 LSB	0 to 281474976710655	Total position is up to 2^{48}-1
	S1...8[15]	Non binary BiSS resolution 16 MSB		
	S1...8 [16]	Communication time [µsec]	Every TS: 15 – (TS-7) Every 2*TS: TS – (2*TS-7)	
	S1...8[17/18]	Absolute position offset **S#[17]** – Low, **S#[18]** – High	Counts	
	S1...8[19]	Position 2 resolution	24-32	Safe position number of bits

Serial Panasonic

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th colspan="2">Values</th><th>Notes</th></tr>
<tr><td rowspan="4">6
Serial
Panasonic</td><td rowspan="4">S1...8[2]</td><td colspan="2">Bitwise Configuration Parameter</td><td colspan="2" rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td>0</td><td>Invert counting direction of sensor</td><td>0

1</td><td>Do not invert

Invert the direction</td><td></td></tr>
<tr><td></td><td></td><td colspan="3">Single Drive</td></tr>
</table>

S1...8[1] ID	Index	Description		Values		Notes
		1-6	Port and resource used by sensor	9	Port C, Secondary Abs	
				Platinum Quad (x4)		
				0	Port A1, Ax1 Secondary Abs	
				8	Port A2, Ax2 Secondary Abs	
				16	Port A3, Ax3 Secondary Abs	
				24	Port A4, Ax4 Secondary Abs	
		7	Sampling rate flag	0 – Every TS 1 – Every 2*TS		
		8-15	Reserved			
		15	Only safety use	0 – No, 1 - Yes		If yes, no RT read
	S1...8[3]	FIR filter window size		0 to 8		0/1 disable filter
	S1...8[4]	Sensor status (error) mask				
	S1...8[5]	Glitch filter [nsec]		40 to 5140		
	S1...8[6]	Clock Frequency [Hz]		2500000		
	S1...8[7]	Sensor resolution		17, 20, 23		
	S1...8[8]	Multi-turn resolution		0, 16		
	S1...8[10]	Reduced bits		0 to 12		
	S1...8[16]	Communication time [µsec]		Every **TS**: 15 – (**TS**-7) Every 2*TS: **TS** – (2*TS-7)		
	S1...8[17/18]	Absolute position offset **S#[17]** – Low, **S#[18]** – High		Counts		

Virtual Sine Signal

S1...8[1] ID	Index	Description		Values		Notes
8 Virtual Sine Signal	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0		0	Do not invert	

S1...8[1] ID	Index	Description		Values		Notes
			Invert counting direction of sensor	1	Invert the direction	
		1-6	Reserved			
		7	Waveform step flag	0 – sine waveform 1 – step waveform		applies to sine A
		8-15	Reserved			
	S1...8[4]	Sensor status (error) mask				
	S1...8[5]	Units of command		1 – Current [A] 2 – Velocity [cnt/sec] 3 – Position [cnt]		
	F1...8[1]	Amplitude sine A				Units according to **S1...8[4]**
	F1...8[2]	Frequency sine A [Hz]				
	F1...8[3]	Amplitude sine B				Units according to **S1...8[4]**
	F1...8[4]	Frequency sine B [Hz]				
	F1...8[5]	Offset of signal				**Units according to S1...8[4]**
	F1...8[6]	Slope of offset				Units according to **S1...8{4]** per second

Serial EnDAT

S1...8[1] ID	**Index**	Description		Values		Notes
9 Serial EnDAT	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				0	Port A, Primary Abs	
				8	Port A, Secondary Abs	
				9	Port C, Secondary Abs	

S1...8[1] ID	Index	Description		Values		Notes
				16	Port A, Primary Abs & Secondary Abs Monitor	
				Drives x 4		
				0	Port A1, Ax1 Secondary Abs	
				8	Port A2, Ax2 Secondary Abs	
				16	Port A3, Ax3 Secondary Abs	
				24	Port A4, Ax4 Secondary Abs	
		7	Sampling rate flag	0 – Every TS 1 – Every 2*TS		
		8-12	Reserved			
		13	Read warning bit indication	0 - No 1 – Yes		
		14	Reserved			
		15	Reserved			
		16	EnDAT mode	0 – EnDAT 2.2 1 – EnDAT 2.1		
	S1…8 [3]	FIR filter window size		0 to 8		0/1 disable filter
	S1…8 [4]	Sensor status (error) mask				
	S1…8 [5]	Glitch filter [nsec]		40 to 5140		
	S1…8 [6]	Clock Frequency [Hz]		156000, 1250000, 2500000, 5000000		
	S1…8 [7]	Sensor resolution		≥ 8		
	S1…8 [8]	Multi-turn resolution		≥ 0		
	S1…8 [16]	Communication time [µsec]		TBD		
	S1…8 [17/18]	Absolute position offset **S1…8 [17]** – Low, **S1…8 [18]** – High		TBD		

Encoder Pulse and Direction

S1...8[1] ID	Index	Description		Values		Notes
11 Encoder Pulse and Direction	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				0	Port A, Secondary Quad 1	
				5	Emulation, Secondary Quad 1	
				9	Port B, Secondary Quad 2	
				10	Port C, Secondary Quad 2	
				11	Single-ended In 7&8, Secondary Quad 2	
				12	Single-ended GPIO 6&7, Secondary Quad 2	
				13	Emulation, Secondary Quad 2	
				17	Port B, Primary Quad 2	
				24	Port A, Primary Quad 1 & Secondary Quad 1	
				25	Port A, Primary Quad 1	
				33	Port B, Primary Quad 2 & Secondary Quad 2	
				Quad Drives x 4		
				0	Port A1, Ax1 Secondary Quad 1	
				5	Emulation Secondary Quad 1 A1,	
				6	Emulation Secondary Quad 1 A2,	

S1...8[1] ID	Index	Description		Values		Notes
				7	Emulation Secondary Quad 1 A3,	
				8	Emulation Secondary Quad 1 A4,	
				9	Port B1, Ax1 Secondary Quad 2	
				18	Port A2, Ax2 Secondary Quad 1	
				27	Port B2, Ax2 Secondary Quad 2	
				36	Port A3, Ax3 Secondary Quad 1	
				45	Port B3, Ax3 Secondary Quad 2	
				54	Port A4, Ax4 Secondary Quad 1	
				63	Port B4, Ax4 Secondary Quad 2	
				Titanium Drives 2X		
				0	X1. Port A1 Secondary Quad 1	
				9	X1. Port B1 Secondary Quad 2	
				10	X2. Port A2 Secondary Quad 1	
				11	X2. Port B2 Secondary Quad 2	
				12	X1. Port A1 Secondary Quad 3	Reserved
				5	X1. Port A1 Secondary Emulation Quad 1	
				13	X1. Port B1 Emulation Quad 2 (Eqep0)	
				6	X2. Port A2 Secondary Emulation Quad 1	
				14	X2. Port B2 Emulation Quad 2 (Eqep2)	

S1...8[1] ID	Index	Description		Values	Notes
		7-14	Reserved	0 – No, 1 - Yes	If yes, no RT read
		15	Only safety use	0 or 1 disable filter	
	S1...8[3]	FIR filter window size		0 to 8	
	S1...8[4]	Sensor status (error) mask			
	S1...8[5]	Glitch filter [nsec]		40 to 5140	
	F1...8[1]	1/T calculation threshold [cnt/sec]			

Encoder Up/Down

S1...8[1] ID	Index	Description		Values		Notes
12 Encoder Up/Down	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				0	Port A, Secondary Quad 1	
				5	Emulation, Secondary Quad 1	
				9	Port B, Secondary Quad 2	
				10	Port C, Secondary Quad 2	
				11	Single-ended In 7&8, Secondary Quad 2	
				12	Single-ended GPIO 6&7, Secondary Quad 2	
				13	Emulation, Secondary Quad 2	
				17	Port B, Primary Quad 2	

S1...8[1] ID	Index	Description	Values		Notes
			24	Port A, Primary Quad 1 & Secondary Quad 1	
			25	Port A, Primary Quad 1	
			33	Port B, Primary Quad 2 & Secondary Quad 2	
			Quad Drives x 4		
			0	Port A1, Ax1 Secondary Quad 1	
			5	Emulation Secondary Quad 1 A1,	
			6	Emulation Secondary Quad 1 A2,	
			7	Emulation Secondary Quad 1 A3,	
			8	Emulation Secondary Quad 1 A4,	
			9	Port B1, Ax1 Secondary Quad 2	
			18	Port A2, Ax2 Secondary Quad 1	
			27	Port B2, Ax2 Secondary Quad 2	
			36	Port A3, Ax3 Secondary Quad 1	
			45	Port B3, Ax3 Secondary Quad 2	
			54	Port A4, Ax4 Secondary Quad 1	
			63	Port B4, Ax4 Secondary Quad 2	
			Titanium Drives 2X		
			0	X1. Port A1 Secondary Quad 1	
			9	X1. Port B1 Secondary Quad 2	
			10	X2. Port A2 Secondary Quad 1	

S1...8[1] ID	Index	Description		Values		Notes
				11	X2. Port B2 Secondary Quad 2	
				12	X1. Port A1 Secondary Quad 3	Reserved
				5	X1. Port A1 Secondary Emulation Quad 1	
				13	X1. Port B1 Emulation Quad 2 (Eqep0)	
				6	X2. Port A2 Secondary Emulation Quad 1	
				14	X2. Port B2 Emulation Quad 2 (Eqep2)	
		7-14	Reserved			
		15	Only safety use	0 – No, 1 - Yes		If yes, no RT read
	S1...8[3]	FIR filter window size		0 to 8		0 or 1 disable filter
	S1...8[4]	Sensor status (error) mask				
	S1...8[5]	Glitch filter [nSec]		40 to 5140		
	F1...8[1]	1/T calculation threshold [cnt/sec]		≥ 0		

Copy Main Profile

15 Copy Main Profile	**S1...8[2]**	Bitwise Configuration Parameter			
		Bit	**Description**		
		0	Reserved		
		1 - 3	Axis used	0 – Axis 1 1 – Axis 2 2 – Axis 3 3 – Axis 4 4 – Axis 5 5 – Axis 6 6-7 - Reserved	Only if axis exists

		4 - 15	Reserved			
	F1...8[1]	Reserved				

Analog Input

S1...8[1] ID	Index	Description		Values		Notes
16 Analog Input	S1...8[2]	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	Invert counting direction of input	0	Do not invert	
				1	Invert the direction	
		4 - 6	Port and resource used by sensor	0	Analog input 1	
				1	Analog input 2	
				2-7	Reserved	
	S1...8[5]	Signal Type		1 – Current 2- Velocity 3 - Position		
	F1...8[5]	Enable/Disable Low pass filter		0 - Disable 1 - Enable		
	F1...8[1]	Filter frequency		1 – 0.8/(2*TS*10^(-6))		
	F1...8[2]	Filter damping		0.3 – 0.95		
	F1...8[6]	Signal gain				Units/Volt

Serial SSI

S1...8[1] ID	Index	Description		Values		Notes
18 Serial SSI	S1...8[2]	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	Reserved			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
				Single Drive		

S1...8[1] ID	Index	Description		Values		Notes
		1-6	Port and resource used by sensor	0	Port A, Primary Abs	
				8	Port A, Secondary Abs	
				9	Port C, Secondary Abs	
				16	Port A, Primary Abs & Secondary Abs Monitor	
				Quad Drives x 4		
				0	Port A1, Ax1 Secondary Abs	0
				8	Port A2, Ax2 Secondary Abs	
				16	Port A3, Ax3 Secondary Abs	
				24	Port A4, Ax4 Secondary Abs	
		7	Sampling rate flag	0 – Every TS 1 – Every 2*TS		
		8	Error and warning bits logic	0 – Active low 1 – Active high		
		9-10	Reserved			
		11	Gray code flag	0 – binary data 1 – gray code data		
		12-14	Reserved			
		15	Only safety use	0 – No, 1 - Yes		If yes, no RT read
	S1...8[3]	FIR filter window size		0 to 8		0 or 1 disable filter
	S1...8[4]	Sensor status (error) mask				
	S1...8[5]	Glitch filter [nsec]		40 to 5140		
	S1...8[6]	Clock Frequency [Hz]		312500, 625000, 1250000, 2500000		
	S1...8[7]	Sensor resolution		≥ 8		
	S1...8[8]	Multi-turn resolution		0-24		
	S1...8[9]	Protocol bits		8-64		Protocol bits include: Position,

S1...8[1] ID	Index	Description	Values	Notes
				Status bits and CRC. exclude: ACK, Start, CDS, Stop.
	S1…8[10]	Reduced bits	0 to 12	
	S1…8[11]	Position LSB bit number	0 to (**S1…8[9]**- **S1…8[7]**- **S1…8[8]**)	Last received bit is 0
	S1…8[12]	Error bit number	-1 to (**S1…8[9]**-1) not including **S1…8[13]** and { **S1…8[11]** - (**S1…8[11]** + **S1…8[7]** + **S1…8[8]**)}	-1 means no error bit
	S1…8[13]	Warning bit number	-1 to (**S1…8[9]**-1) not including **S1…8[12]** and { **S1…8[11]** - (**S1…8[11]** + **S1…8[7]** + **S1…8[8]**)}	-1 means no warning bit
	S1…8[16]	Communication time [µsec]	Every **TS**: 15 – (**TS**-7) Every 2***TS**: **TS** – (2***TS**-7)	
	S1…8[17 - 18]	Absolute position offset **S1…8 [17]** – Low, **S1…8 [18]** – High		Counts
	F1…8[1]	First clock low level duration[µsec]	≥ 0	If lower than half clock cycle, then automatically set to half clock cycle

Yaskawa

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th colspan="2">Values</th><th>Notes</th></tr>
<tr><td rowspan="6">19
Yaskawa</td><td rowspan="6">S1…8[2]</td><td colspan="2">Bitwise Configuration Parameter</td><td colspan="2" rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td>0</td><td>Reserved</td><td colspan="2"></td><td></td></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Invert counting direction of sensor</td><td>0</td><td>Do not invert</td><td rowspan="2"></td></tr>
<tr><td>1</td><td>Invert the direction</td></tr>
</table>

		1-6	Port and resource used by sensor	**Single Drive**		
				9	Port C, Secondary Abs	
				Drives x 4		
				0	Port A1, Ax1 Secondary Abs	0
				8	Port A2, Ax2 Secondary Abs	
				16	Port A3, Ax3 Secondary Abs	
				24	Port A4, Ax4 Secondary Abs	
		7	Sampling rate flag	0 – Every **TS** 1 – Every 2***TS**		
		22-23	Encoder Type	0 – Absolute 1 – Incremental 2-3 - Reserved		
	S1...8[3]	FIR filter window size		0 to 8		0 or 1 disable filter
	S1...8[4]	Sensor status (error) mask				
	S1...8[5]	Glitch filter [nsec]		40 to 5140		
	S1...8[6]	Clock Frequency [Hz]		4000000		
	S1...8[7]	Sensor resolution		For Absolute type: 16, 17, 20, 22, 24 For Incremental type: 13, 16, 17,20,2 2, 24		
	S1...8[8]	Multi-turn resolution		0, 16		
	S1...8[10]	Reduced bits		0 to 12		
	S1...8[16]	Communication time [µsec]		Every **TS**: 15 – (**TS**-7) Every 2***TS**: **TS** – (2***TS**-7)		
	S1...8[17 - 18]	Absolute position offset **S1...8 [17]** – Low, **S1...8 [18]** – High				Counts

Gantry 4X

S1...8[1] ID	Index	Description		Values	Notes
20 Gantry 4X	**S1...8 [2]**	Bitwise Configuration Parameter			
		Bit	**Description**		
		1-3	Master axis	0 – Axis 1 1 – Axis 2 2 – Axis 3 3 – Axis 4 4 – Axis 5 5 – Axis 6	Only if axis exists
		4-6	Slave axis	0 – Axis 1 1 – Axis 2 2 – Axis 3 3 – Axis 4 4 – Axis 5 5 – Axis 6	Only if axis exists
		9	Gantry resource	0 - 1st Gantry 1 - 2nd Gantry	
	S1...8 [8]	Gantry X1 position socket		0 – Master commutation socket 1 – Socket 1 2 – Socket 2 ..... 8 - Socket 8	
	S1...8 [9]	Gantry X2 position socket		0 – Slave commutation socket 1 – Socket 1 2 – Socket 2 ..... 8 - Socket 8	
	S1...8 [10]	Gantry X1 velocity socket		0 – Master commutation socket 1 – Socket 1 2 – Socket 2 ..... 8 - Socket 8	
	S1...8 [11]	Gantry X2 velocity socket		0 – Slave commutation socket	

S1...8[1] ID	Index	Description	Values	Notes
			1 – Socket 1 2 – Socket 2 ..... 8 - Socket 8	

Serial Exclusive

S1...8[1] ID	Index	Description		Values	Notes
21 Serial Exclusive	**S1...8 [2]**	Bitwise Configuration Parameter			
		Bit	**Description**		
		0	Reserved		
		1 - 3	Axis associated with socket	0 – Axis 1 1 – Axis 2 2 – Axis 3 3 – Axis 4 4 – Axis 5 5 – Axis 6 6-7 - Reserved	Only if axis exists
		4 - 5	Communication channel number	0 – Channel 1 1 – Channel 2 2 – Channel 3	
		6 - 7	Reserved		
		8	Drive designation	0 - Slave 1 - Master	
		9 - 13	Reserved		
		14-15	Reserved		Safety use only
	S1...8 [3]	FIR filter window size		0 to 8	0 or 1 disable filter
	S1...8 [4]	Sensor status (error) mask			
	S1...8 [5]	Data transfer options		0 – Sync only 1 – Attached Drive 2 - Gantry	
	S1...8 [6]	First transferred position selection. Note that the velocity that is transmitted is taken from this socket as well.		0x00 to 0x08 0x10 to 0x18 0x20 to 0x28	Bits 0-3: 0 – Main feedback position; 1 to 8 – Socket number

S1...8[1] ID	Index	Description	Values	Notes
				Bits 4-7: 0 – Main socket signal; 1 – Aux 1 socket signal; 2 – Aux 2 socket signal;
	S1...8 [7]	Second transferred position selection	0x00 to 0x08 0x10 to 0x18 0x20 to 0x28	Bits 0-3: 0 – Disabled; 1 to 8 – Socket number Bits 4-7: 0 – Main socket signal; 1 – Aux 1 socket signal; 2 – Aux 2 socket signal;
	S1...8 [8]	Gantry X1 position socket	0 – Master commutation socket 1 – Socket 1 2 – Socket 2 ..... 8 - Socket 8	0 - commutation socket

Analog Resolver

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th colspan="2">Values</th><th>Notes</th></tr>
<tr><td rowspan="6">22
Analog
Resolver</td><td rowspan="6">S1...8[2]</td><td colspan="2">Bitwise Configuration Parameter</td><td colspan="2" rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td>0</td><td>Invert counting direction of sensor</td><td>0
1</td><td>Do not invert
Invert the direction</td><td></td></tr>
<tr><td>1-8</td><td>Reserved</td><td colspan="2"></td><td></td></tr>
<tr><td>9</td><td>Software Capture</td><td colspan="2">TBD</td><td></td></tr>
<tr><td>10-12</td><td>Excitation signal's frequency</td><td colspan="2">0- 0.5*Ts
1- Ts
2-2*Ts</td><td></td></tr>
</table>

S1...8[1] ID	Index	Description	Values	Notes
		13-14 Reserved		
		15 Only safety use	0 – No, 1 - Yes	If yes, no RT read
	S1...8[3]	FIR filter window size	0 to 8	0/1 disable filter
	S1...8[4]	Sensor status (error) mask		
	S1...8[5]	Glitch filter [nsec]	40 to 5140	
	S1...8[6]	Multiplication factor	2 to 16	
	S1...8[9]	Sine offset [ADC]	-32768 to 32767	
	S1...8[10]	Cosine offset [ADC]	-32768 to 32767	
	S1...8[11]	Minimum signal amplitude [ADC]	3276 to 32768	
	S1...8[12]	Maximum signal amplitude [ADC]	3276 to 32768	
	S1...8[13]	Low signal amplitude warning [ADC]	3276 to 32768	
	S1...8[14]	High signal amplitude warning [ADC]	3276 to 32768	
	S1...8[15]	Excitation signal's offset[ticks]	0 to (0.5*Ts)/(1 tick)	1 tick = 20 nanosecond
	S1...8[16]	Resolver pole pairs number	>0	
	S1...8[17]	Absolute position offset	TBD	
	F1...8[1]	Phase shift [deg]	-90 to 90	
	F1...8[2]	Sine gain multiplier	0.3 to 1.7	

Serial Sanyo/Nikon

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th colspan="2">Values</th><th>Notes</th></tr>
<tr><td rowspan="6">25
Serial
Sanyo/Nikon</td><td rowspan="6">S1...8[2]</td><td colspan="2">Bitwise Configuration Parameter</td><td colspan="2" rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Invert counting direction of sensor</td><td>0</td><td>Do not invert</td><td rowspan="2"></td></tr>
<tr><td>1</td><td>Invert the direction</td></tr>
<tr><td rowspan="2">1-6</td><td rowspan="2">Port and resource used by sensor</td><td colspan="3">Single Drive</td></tr>
<tr><td>9</td><td>Port C, Secondary Abs</td><td></td></tr>
</table>

S1...8[1] ID	Index	Description		Values	Notes
		7	Sampling rate flag	0 – Every TS 1 – Every 2*TS	
		8-14	Reserved		
		15	Only safety use	0 – No, 1 - Yes	If yes, no RT read
	S1…8[3]	FIR filter window size		0 to 8	0/1 disable filter
	S1…8[4]	Sensor status (error) mask			
	S1…8[5]	Glitch filter [nsec]		40 to 5140	
	S1…8[6]	Clock Frequency [Hz]		2500000	
	S1…8[7]	Sensor resolution		≥ 8	
	S1…8[8]	Multi-turn resolution		≥ 0	
	S1…8[10]	Reduced bits		0 to 8	
	S1…8[16]	Communication time [µsec]		TBD	
	S1…8[17/18]	Absolute position offset **S#[17]** – Low, **S#[18]** – High		TBD	

Socket Profile

S1...8[1] ID	Index	Description	Values	Notes
26 Socket Profile	**S1…8[2…20]**	Reserved		
	F1…8[1]	Reserved		

Hiperface

S1...8[1] ID	Index	Description		Values		Notes
28 Hiperface	**S1…8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1 - 6	Reserved			

S1...8[1] ID	Index	Description		Values	Notes
		7	Hiperface Interpolation per analog period	0 to 5 bit interpolation 1 to 8 bit interpolation	
		8	Hiperface serial and analog signals direction correlation	0 – Signals correlated 1–Signals uncorrelated	
		9 - 14	Reserved		
		15	Only safety use	0 – No, 1 - Yes	If yes, no RT read
	S1...8[3]	FIR filter window size		0 to 8	0/1 disable filter
	S1...8[4]	Sensor status (error) mask			
	S1...8[5]	Glitch filter [nSec]		40 to 5140	
	S1...8[6]	Multiplication factor		2 to 16	
	S1...8[7]	Hiperface single-turn resolution		8 to 32	Max total serial resolution is 32 bits
	S1...8[8]	Hiperface multi-turn resolution		0 to 16	
	S1...8[9]	Sine offset [ADC]		-32768 to 32767	
	S1...8[10]	Cosine offset [ADC]		-32768 to 32767	
	S1...8[11]	Minimum signal amplitude [ADC]		3276 to 32768	
	S1...8[12]	Maximum signal amplitude [ADC]		3276 to 32768	
	S1...8 [13]	Low signal amplitude warning [ADC]		3276 to 32768	
	S1...8 [14]	High signal amplitude warning [ADC]		3276 to 32768	
	S1...8 [17]	Absolute position offset		TBD	
	F1...8[1]	Phase shift [deg]		-90 to 90	
	F1...8[2]	Sine gain multiplier		0.3 to 1.7	
	F1...8[3]	Quad only calculation threshold [Hz]		≥ 0	

General Reference 0x3605

S1...8[1] ID	Index	Description		Values	Notes
29 General Reference 0x3605	**S1...8[2]**	Bitwise Configuration Parameter			
		Bit	**Description**		
		0	Reserved		
		1 - 3	Axis used	0 – Axis 1 1 – Axis 2 2 – Axis 3 3 – Axis 4 4 – Axis 5 5 – Axis 6 6-7 - Reserved	Only if axis exists
		4 - 15	Reserved		
	F1...8[1]	Reserved			

Speed Observer Simplified

S1...8[1] ID	Index	Description		Values	Notes
30 Speed Observer Simplified	**S1...8[2]**	Bitwise Configuration Parameter			
		Bit	**Description**		
		0	Reserved		
		1 - 3	Axis used	0 – Axis 1 1 – Axis 2 2 – Axis 3 3 – Axis 4 4 – Axis 5 5 – Axis 6 6-7 - Reserved	Only if axis exists
		4 - 15	Reserved		
	F1...8[1]	Motor Resolution		> 0	
	F1...8[2]	Voltage Dead Band Threshold		> 0	

Motor Simulation

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th>Values</th><th>Notes</th></tr>
<tr><td rowspan="16">34
Motor
Simulation</td><td rowspan="6">S1...8[2]</td><td colspan="2">Bitwise Configuration Parameter</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td>0</td><td>Reserved</td><td></td><td></td></tr>
<tr><td>1-3</td><td>Axis used</td><td>0 – Axis 1
1 – Axis 2
2 – Axis 3
3 – Axis 4
4 – Axis 5
5 – Axis 6
6 - 7 - Reserved</td><td>Only if axis exists</td></tr>
<tr><td>4-6</td><td>Motor type</td><td>0 – 3-phase brushless
1 – DC motor
2 - 7 – Reserved</td><td></td></tr>
<tr><td>7-15</td><td>Reserved</td><td></td><td></td></tr>
<tr><td>S1...8[4]</td><td colspan="2">Sensor status (error) mask</td><td></td><td></td></tr>
<tr><td>S1...8[5]</td><td colspan="2">Motor magnetic pole pairs</td><td>≥ 0</td><td></td></tr>
<tr><td>S1...8[6]</td><td colspan="2">Motor resolution 32 LSB[count/rev]</td><td rowspan="2">0 to 281474976710655</td><td rowspan="2">Total position is up to 2^{48}-1</td></tr>
<tr><td>S1...8[7]</td><td colspan="2">Motor resolution 16 MSB[count/rev]</td></tr>
<tr><td>F1...8[1]</td><td colspan="2">Motor phase resistance [Ω]</td><td>> 0</td><td></td></tr>
<tr><td>F1...8[2]</td><td colspan="2">Motor phase inductance [Henry]</td><td>> 0</td><td></td></tr>
<tr><td>F1...8[3]</td><td colspan="2">phase Back EMF constant [V/Krpm]</td><td></td><td></td></tr>
<tr><td>F1...8[4]</td><td colspan="2">Torque constant [N*m/A]</td><td></td><td></td></tr>
<tr><td>F1...8[5]</td><td colspan="2">Mechanical inertia [Kg*m2]</td><td>> 0</td><td></td></tr>
<tr><td>F1...8[6]</td><td colspan="2">Viscous friction coefficient [N*m/(rad/sec)]</td><td></td><td></td></tr>
</table>

BEMF Observer

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th>Values</th><th>Notes</th></tr>
<tr><td rowspan="13">36
BEMF
Observer</td><td rowspan="8">S1...8[2]</td><td colspan="2">Bitwise Configuration Parameter</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>Bit</td><td>Description</td></tr>
<tr><td>0</td><td>Reserved</td><td></td><td></td></tr>
<tr><td>1 - 3</td><td>Axis used</td><td>0 – Axis 1
1 – Axis 2
2 – Axis 3
3 – Axis 4
4 – Axis 5
5 – Axis 6
6-7 - Reserved</td><td>Only if axis exists</td></tr>
<tr><td>4 - 6</td><td>Current type during initial phasing</td><td>0 - 3</td><td>0 – active current
1 – reactive current Method 1
2 – reactive current Method 2
3 – reactive current Method 3</td></tr>
<tr><td>8</td><td>Initial phasing: stepper mode</td><td>0, 1</td><td>0 – current stepper mode
1 – speed stepper mode (closed loop)</td></tr>
<tr><td>9</td><td>Initial phasing: current command source</td><td>0, 1</td><td>0 – defined by SC[1] [% of PL[1]]
1 – defined by HT[1/2/3] [A]</td></tr>
<tr><td>10 - 15</td><td>Reserved</td><td></td><td></td></tr>
<tr><td>S1...8[3]</td><td colspan="2">FIR filter window size</td><td>0 to 8</td><td>0/1 disable filter</td></tr>
<tr><td>S1...8[4]</td><td colspan="2">Sensor status (error) mask</td><td></td><td></td></tr>
<tr><td>S1...8[5]</td><td colspan="2">Observer switching gain</td><td></td><td></td></tr>
<tr><td>S1...8[6]</td><td colspan="2">Observer LPF frequency [Hz]</td><td></td><td></td></tr>
<tr><td>F1...8[1]</td><td colspan="2">Motor phase inductance [Henry]</td><td>> 0</td><td></td></tr>
</table>

S1...8[1] ID	Index	Description	Values	Notes
	F1...8[2]	Motor phase resistance [Ω]	> 0	

Generic Serial IMU/Gyro

S1...8[1] ID	Index	Description		Values	Notes
38 Generic Serial IMU/Gyro	**S1...8[2]**	Bitwise Configuration Parameter			
		Bit	**Description**		
		1	Header indication	0-1	
		2	Main velocity indication	0-1	
		3	Aux 1 velocity indication	0-1	
		4	Aux 2 velocity indication	0-1	
		5	Additional data 1 indication	0-1	
		6	Additional data 2 indication	0-1	
		7	Additional data 3 indication	0-1	
		8	Additional data 4 indication	0-1	
		9	Additional data 5 indication	0-1	
		10	Checksum indication	0-1	
	S1...8[6]	Protocol bytes		4-64	0/1 disable filter
	S1...8[7]	Time out[TS cycles]		1-255	
	S1...8[8]	Header Value			
	S1...8[9]	0-7	Header first byte	0-(**S1…8[6]** - 1)	
		8-15	Header number of bytes	0-4	
	S1...8[10]	0-7	Main first byte	0-(**S1…8[6]** - 1)	
		8-15	Main number of bytes	0-4	
	S1...8[11]	0-7	Aux 1 first byte	0-(**S1…8[6]** - 1)	
		8-15	Aux 1 number of bytes	0-4	
	S1...8[12]	0-7	Aux 2 first byte	0-(**S1…8[6]** - 1)	
		8-15	Aux 2 number of bytes	0-4	
	S1...8[13]	0-7	Additional data 1 first byte	0-(**S1…8[6]** - 1)	

S1...8[1] ID	Index	Description		Values	Notes
		8-15	Additional data 1 number of bytes	0-4	
	S1...8[14]	0-7	Additional data 2 first byte	0-(**S1...8[6]** - 1)	
		8-15	Additional data 2 number of bytes	0-4	
	S1...8[15]	0-7	Additional data 3 first byte	0-(**S1...8[6]** - 1)	
		8-15	Additional data 3 number of bytes	0-4	
	S1...8[16]	0-7	Additional data 4 first byte	0-(**S1...8[6]** - 1)	
		8-15	Additional data 4 number of bytes	0-4	
	S1...8[17]	0-7	Additional data 5 first byte	0-(**S1...8[6]** - 1)	
		8-15	Additional data 5 number of bytes	0-4	
	S1...8[18]	Checksum type		0-10	
	F1...8[1]	Main gain		!=0	
	F1...8[2]	Aux 1 gain		!=0	
	F1...8[3]	Aux 2 gain		!=0	
	PP[6]	Baud rate			

Mitsubishi

S1...8[1] ID	Index	Description		Values		Notes
39 Mitsubishi	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	**Reserved**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				9	Port C, Secondary Abs	
				Drives x 4		
				0	Port A1, Ax1 Secondary Abs	

S1...8[1] ID	Index	Description		Values		Notes
				8	Port A2, Ax2 Secondary Abs	
				16	Port A3, Ax3 Secondary Abs	
				24	Port A4, Ax4 Secondary Abs	
		7	Sampling rate flag	0 – Every TS 1 – Every 2*TS		
	S1...8[4]	Error Bitwise Mask				
	S1...8[5]	Input Filter[nsec]		40-5140		
	S1...8[6]	Clock Frequency[Hz]		2500000		
	S1...8[7]	Encoder bit resolution		26		
	S1...8[8]	Multi turn bits		0, 16		
	S1...8[10]	Reduced bits		0 - 12		
	S1...8[16]	Communication Time		Every TS: 15 – (TS-7) Every 2*TS: TS – (2*TS-7)		
	S1...8[17 - 18]	Absolute Position Offset		Counts		

Acurolink

S1...8[1] ID	Index	Description		Values		Notes
40 Acurolink	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	**Reserved**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				9	Port C, Secondary Abs	
		7	Sampling rate flag	0 – Every TS 1 – Every 2*TS		
		19	Safe Acurolink	0 – No 1 - Yes		
	S1...8[4]	Error Bitwise Mask				

S1...8[1] ID	Index	Description	Values	Notes
	S1...8[5]	Input Filter[nsec]	40-5140	
	S1...8[7]	Encoder bit resolution	17 - 24	
	S1...8[8]	Multi turn bits	0 - 16	
	S1...8[10]	Reduced bits	0 - 12	
	S1...8[16]	Communication Time	Every TS: 15 – (TS-7) Every 2*TS: TS – (2*TS-7)	
	S1...8[17 - 18]	Absolute Position Offset	Counts	

Openlink

S1...8[1] ID	Index	Description		Values		Notes
41 Openlink	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	**Reserved**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				9	Port C, Secondary Abs	
		7	Sampling rate flag	0 – Every **TS** 1 – Every 2***TS**		
		19	Safe Openlink	0 – No 1 - Yes		
	S1...8[4]	Error Bitwise Mask				
	S1...8[5]	Input Filter[nsec]		40-5140		
	S1...8[7]	Encoder bit resolution		17 - 24		
	S1...8[8]	Multi turn bits		0 - 16		
	S1...8[10]	Reduced bits		0 - 12		
	S1...8[16]	Communication Time		Every **TS**: 15 – (**TS**-7) Every 2***TS**: **TS** – (2***TS**-7)		
	S1...8[17 - 18]	Absolute Position Offset		Counts		

DSL

S1...8[1] ID	Index	Description		Values		Notes
42 DSL	**S1...8[2]**	Bitwise Configuration Parameter				
		Bit	**Description**			
		0	**Reserved**			
		0	Invert counting direction of sensor	0	Do not invert	
				1	Invert the direction	
		1-6	Port and resource used by sensor	**Single Drive**		
				9	Port C, Secondary Abs	
		7	Sampling rate flag	0 – Every TS 1 – Every 2*TS		
		19	Safe Dsl	0 – No 1 - Yes		
	S1...8[4]	Error Bitwise Mask				
	S1...8[5]	Input Filter[nsec]		40-5140		
	S1...8[7]	Encoder bit resolution		16 - 24		
	S1...8[8]	Multi turn bits		0 - 16		
	S1...8[10]	Reduced bits		0 - 12		
	S1...8[17 - 18]	Absolute Position Offset		Counts		

Enhanced Serial IMU/Gyro Interface

S1...8[1] ID	Index	Description		Values	Restrictions / Notes
43 Enhanced Serial IMU/Gyro Interface	**S1...8[2]**	Bitwise Configuration Parameter			
		Bit	**Description**		
		0	Reserved		
		1	Endianness type	0 – little 1 – big	
		2	Acceleration X sign	0 – not inverted 1 – inverted	
		3	Acceleration Y sign		
		4	Acceleration Z sign		
		5	Magnetometer X sign		

S1...8[1] ID	Index	Description		Values	Restrictions / Notes
		6	Magnetometer Y sign		
		7	Magnetometer Z sign		
		8..15	Reserved		
	S1...8[3]	FIR filter window size		0	Reserved
	S1...8[4]	Sensor status (error) mask			
	S1...8[5]	Reserved			
	S1...8[6]	Protocol configuration			
		Bits	**Description**		
		7...0	Number of protocol bytes		Number of bytes sent in the protocol
		15...8	Timeout		Number of TS Samples
		23...16	Header length		Number of header bytes
		31...24	Reserved		
	S1...8[7]	Header value			
	S1...8[8]	Check sum configuration			
		Bits	**Description**		
		7...0	Check sum first byte		
		15...8	Check sum number of bytes		
		23...16	Check sum type		
		31...24	Reserved		
	S1...8[9]	Angular velocity			Data type: • 0-float • 1-integer signed • 2- integer unsigned
		Bits	**Description**		
		7...0	Main velocity (Gyro X) first byte		
		12...8	Main velocity (Gyro X) number of bytes		
		13..15	Main velocity (Gyro X) data type		
		23...16	Aux 1 velocity (Gyro Y) first byte		

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th>Values</th><th>Restrictions / Notes</th></tr>
<tr><td></td><td></td><td>28...24</td><td>Aux 1 velocity (Gyro Y) number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>29..31</td><td>Aux 1 velocity (Gyro Y) data type</td><td></td><td></td></tr>
<tr><td></td><td>S1...8[10]</td><td colspan="2">Angular velocity / Acceleration</td><td></td><td>Data type:
• 0-float
• 1-integer signed
• 2- integer unsigned</td></tr>
<tr><td></td><td></td><td>Bits</td><td>Description</td><td></td><td></td></tr>
<tr><td></td><td></td><td>7...0</td><td>Aux 2 velocity (Gyro Z) first byte</td><td></td><td></td></tr>
<tr><td></td><td></td><td>12...8</td><td>Aux 2 velocity (Gyro Z) number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>13..15</td><td>Aux 2 velocity (Gyro Z) data type</td><td></td><td></td></tr>
<tr><td></td><td></td><td>23...16</td><td>Main current (Acc X) first byte</td><td></td><td></td></tr>
<tr><td></td><td></td><td>28...24</td><td>Main current (Acc X) number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>29..31</td><td>Main current (Acc X) data type</td><td></td><td></td></tr>
<tr><td></td><td>S1...8[11]</td><td colspan="2">Acceleration</td><td></td><td>Data type:
• 0-float
• 1-integer signed
• 2- integer unsigned</td></tr>
<tr><td></td><td></td><td>Bits</td><td>Description</td><td></td><td></td></tr>
<tr><td></td><td></td><td>7...0</td><td>Aux 1 current (Acc Y) first byte</td><td></td><td></td></tr>
<tr><td></td><td></td><td>12...8</td><td>Aux 1 current (Acc Y) number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>13..15</td><td>Aux 1 current (Acc Y) data type</td><td></td><td></td></tr>
<tr><td></td><td></td><td>23...16</td><td>Aux 2 current (Acc Z) first byte</td><td></td><td></td></tr>
<tr><td></td><td></td><td>28...24</td><td>Aux 2 current (Acc Z) number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>29..31</td><td>Aux 2 current (Acc Z) data type</td><td></td><td></td></tr>
<tr><td></td><td>S1...8[12]</td><td colspan="2">Orientation angle</td><td></td><td>Data type:
• 0-float
• 1-integer signed
• 2- integer unsigned</td></tr>
<tr><td></td><td></td><td>Bits</td><td>Description</td><td></td><td></td></tr>
<tr><td></td><td></td><td>7...0</td><td>Main position (Orientation X) first byte</td><td></td><td></td></tr>
</table>

<table>
<tr><th>S1...8[1] ID</th><th>Index</th><th colspan="2">Description</th><th>Values</th><th>Restrictions / Notes</th></tr>
<tr><td></td><td></td><td>12…8</td><td>Main position (Orientation X) number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>13..15</td><td>Main position (Orientation X) data type</td><td></td><td></td></tr>
<tr><td></td><td></td><td>23…16</td><td>Aux 1 position (Orientation Y) first byte</td><td></td><td></td></tr>
<tr><td></td><td></td><td>28…24</td><td>Aux 1 position (Orientation Y) number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>29..31</td><td>Aux 1 position (Orientation Y) data type</td><td></td><td></td></tr>
<tr><td></td><td>S1...8[13]</td><td colspan="2">Orientation angle</td><td></td><td>Data type:
• 0-float
• 1-integer signed
• 2- integer unsigned</td></tr>
<tr><td></td><td></td><td>Bits</td><td>Description</td><td></td><td></td></tr>
<tr><td></td><td></td><td>7…0</td><td>Aux 2 position (Orientation Z) first byte</td><td></td><td></td></tr>
<tr><td></td><td></td><td>12…8</td><td>Aux 2 position (Orientation Z) number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>13..15</td><td>Aux 2 position (Orientation Z) data type</td><td></td><td></td></tr>
<tr><td></td><td></td><td>30…16</td><td>Reserved</td><td></td><td></td></tr>
<tr><td></td><td>S1...8[14]</td><td colspan="2">Magnetometer</td><td></td><td>Data type:
• 0-float
• 1-integer signed
• 2- integer unsigned</td></tr>
<tr><td></td><td></td><td>Bits</td><td>Description</td><td></td><td></td></tr>
<tr><td></td><td></td><td>7…0</td><td>Magnetometer X first byte</td><td></td><td></td></tr>
<tr><td></td><td></td><td>12…8</td><td>Magnetometer X number of bytes</td><td></td><td></td></tr>
<tr><td></td><td></td><td>13..15</td><td>Magnetometer X data type</td><td></td><td></td></tr>
<tr><td></td><td></td><td>23…16</td><td>Magnetometer Y first byte</td><td></td><td></td></tr>
</table>

S1...8[1] ID	Index	Description		Values	Restrictions / Notes
		28...24	Magnetometer Y number of bytes		
		29..31	Magnetometer Y data type		
	S1...8[15]	Magnetometer and Additional data 1			Data type: • 0-float • 1-integer signed • 2- integer unsigned
		.	**Description**		
		7...0	Magnetometer Z first byte		
		12...8	Magnetometer Z number of bytes		
		13..15	Magnetometer Z data type		
		23...16	Additional data 1 first byte		
		28...24	Additional data 1 number of bytes		
		29..31	Additional data 1 data type		
	S1...8[16]	Additional data 2 and 3			Data type: • 0-float • 1-integer signed • 2- integer unsigned
		Bits	**Description**		
		7...0	Additional data 2 first byte		
		12...8	Additional data 2 number of bytes		
		13..15	Additional data 2 data type		
		23...16	Additional data 3 first byte		
		28...24	Additional data 3 number of bytes		
		29..31	Additional data 3 data type		
	S1...8[17]	Reserved			
	S1...8[18]				
	S1...8[19]				
	S1...8[20]				

S1...8[1] ID	Index	Description	Values	Restrictions / Notes
	F1...8[1]	Gyro X gain		
	F1...8[2]	Gyro Y gain		
	F1...8[3]	Gyro Z gain		
	F1...8[4]	Orientation X gain		
	F1...8[5]	Orientation Y gain		
	F1...8[6]	Orientation Z gain		
	F1...8[7]	X (Roll) and Y (Pitch) weight factor		Complimentary filter parameter
	F1...8[8]	Z (Yaw) weight factor		Complimentary filter parameter
	F1...8[9]	Reserved		
	F1...8[10]	Reserved		

SIL Sensor

S1...8[1] ID	Index	Description	Values	Notes
44 SIL Sensor	**S1...8[2...20]**	Reserved		
	F1...8[1]	Reserved		

References

H1...8[]

SA – Socket Function

SA sets which socket is used for different functions.

CANopen/CoE

Elmo Alias Object: 0x3292

Attributes

Attribute	Description
Type	Unsigned Short
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 14
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

It should be noted that modification of the command should be performed using the EAS Quick or Expert Tuning interface. Manual modification may result in ambiguous results from the sensor or function that uses it.

The following bit order applies to **SA** commands (except for **SA[11]**)

<table>
<tr><th>Bits</th><th colspan="3">Details</th><th>Range</th><th>Notes</th></tr>
<tr><td rowspan="3">0-3</td><td>If Bits 8-11 = 0</td><td colspan="2">Socket number</td><td>0 to 8</td><td>0 – the axis reference signals will result in 0</td></tr>
<tr><td>If Bits 8-11 = 1</td><td colspan="2">Axis number</td><td>0 to WS[1]</td><td>0 – the axis reference signals will result in 0;
Must be different from the axis number associated with present command: AXn.SA[#].0-3 !=n</td></tr>
<tr><td>If Bits 8-11 = 2</td><td colspan="2">Signal generator</td><td>1</td><td>0 – the axis reference signals will result in 0</td></tr>
<tr><td rowspan="4">4-7</td><td rowspan="4">If Bits 8-11 = 0</td><td colspan="2">Socket signal used for axis function</td><td>0 to 2</td><td></td></tr>
<tr><td>0</td><td>Main socket signal</td><td></td><td></td></tr>
<tr><td>1</td><td>Auxiliary 1 socket signal</td><td></td><td></td></tr>
<tr><td>2</td><td>Auxiliary 2 socket signal</td><td></td><td></td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="3">Details</th><th>Range</th><th>Notes</th></tr>
<tr><td rowspan="4"></td><td rowspan="4">If Bits 8-11 = 1</td><td colspan="2">Axis signal used for axis function</td><td>0 to 1</td><td></td></tr>
<tr><td>0</td><td>DS402 profile command</td><td></td><td>SA[4], SA[5], SA[6] – profile current command
SA[14] – profile position command
SA[1-3,7-13] – not applicable</td></tr>
<tr><td>1</td><td>Yaw controller current command</td><td></td><td>SA[4], SA[5] in Gantry applications only</td></tr>
<tr><td>2</td><td>Control reference</td><td></td><td>SA[4], SA[5], SA[6] – total current command
SA[14] – total position command
SA[1-3, 7-13] – not applicable</td></tr>
<tr><td rowspan="4">8-11</td><td colspan="3">Reference source</td><td>0 - 2</td><td></td></tr>
<tr><td>0</td><td colspan="2">Socket</td><td></td><td></td></tr>
<tr><td>1</td><td colspan="2">Axis</td><td></td><td></td></tr>
<tr><td>2</td><td colspan="2">Signal generator</td><td></td><td>Applicable for SA[5,6,12,13]</td></tr>
</table>

Examples:

- **SA[1]**=0x12 means that auxiliary signal 1 of socket 2 is used as position feedback
- **SA[4]**=0x112 means that Axis #2 Yaw controller current command is used as direct current reference

Indices

The following table details the **SA** entries.

Index	Description	Details
1	Position feedback socket selection	The motor must be off to change the setting.
2	Velocity feedback socket selection	
3	Commutation socket selection	The motor must be off to change the setting.
4	First direct current reference socket selection	
5	Second direct current reference socket selection.	Used also for identifying Gantry plant.
6	Yaw direct current reference socket selection	
7	Socket selection for position gain scheduling by position.	

Index	Description	Details
8	Touch Probe socket selection	The Socket and signal that is used for the position for the touch probe. Change is not allowed while capture is enabled
9	DS402 Homing mode socket selection.	The Socket and signal that are used for the DS402 homing mode position. If socket number is 0 (bits 0-3), the position feedback socket setting (**SA[1]**) is used for this function.
10	Gantry Yaw feedback socket selection	
11	Additional sockets used for sensor status (errors & warnings) reporting.	Select the required socket(s) number(s) as follows: For socket number [N] set bit number [N] in the sequence. i.e. To select socket #5, set bit #5 in the variable (“0x20“). To select socket #3, set bit #3 (“0x8“). The available range is 0 – 0x1FF. Multiple sockets can be selected. **Note:** Bit #0 is unused and should be kept “0”, otherwise an error will be triggered. Note that if the additional socket is used for errors then the motor will be disabled when an error occurs in this socket.
12	Position direct reference socket selection	The selected socket position value is used as a direct reference to the Position Controller. The motor must be off to change the setting.
13	Velocity direct reference socket selection	The selected socket position value is used as a direct reference to the Velocity Controller
14	Yaw control loop reference source selection	The optional values are detailed in the table above

Notes

- All default values are 0
- SA[4], SA[5], SA[12],SA[13]:
  - Stop manager functions do not apply for these entries.
  - In situations where the brake is used (**BP[]**) the reference command might be active regardless of the brake state.

References

SC[N] – Stepper Commutation

SC[N] gets and sets parameters used for the stepper, binary search and sensor-less commutation algorithms.

CANopen/CoE

Elmo Alias Object: 0x3294

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	See the table below
Index Range	1 to 16
Default	See the table below
Unit Modes	All modes (except for **SC[8]**, used only in stepper mode)
Non-Volatile	Yes
Axis Related	Yes

Remarks

With stepper commutation (**CA[17]** = 2) the commutation algorithm increases the current at the stepper angle 0 degree to a value define by **SC[1]**, waits until the speed is stable and then changes the stepper angle by **SC[10]** degrees, and waits again until the speed is stable. This is repeated until the correct commutation phase is found; at this point the commutation phase is saved. Then the motor moves back to the position before the commutation, and the unit mode changes to the original mode requested.

With binary search commutation (**CA[17]** = 3) the commutation angle converges until it is found. There is almost no movement in this process. To verify the binary search result set **SC[12]=**1.

With Gurley commutation (**CA[17]** = 6) the commutation algorithm simultaneously increases the current and the stepper angle until the Gurley position becomes valid, then waits **SC[3]** seconds. At this point the commutation phase is known and saved, the unit mode changes to the original mode requested.

With the Motor On in stepper mode (**UM** = 3), it automatically sets the current command for stepper mode.

The activation of **SC[N]** occurs in the next motor enable.

Indices

The following table describes the **SC[]** entries.

Index	Description	Unit	Default	Range	In Process
1	Desired current in the process	Percentage of the maximum current (**PL[1]**)	50	0 to 100	Stepper, binary search and

Index	Description	Unit	Default	Range	In Process
					sensorless commutation
2	Time to increase from 0 to the desired current	Seconds	0.5	0 to 12	Stepper, binary search and sensorless commutation
3	Time to stabilize the motor. **Stepper commutation:** If the speed is below the value of **SC[5]** for a period of **SC[3]** seconds, the motor is stable, and the commutation angle can be calculated. **Binary search:** After the algorithm finds the angle, it increases the current for **SC[3]** seconds, to make sure that the position in this angle will stabilize.	Seconds	1.0	0.001 to 6	Stepper, binary search and sensorless commutation
4	Reserved for compatibility		1.0	Reserved for compatibility	
5	Stepper commutation: Low speed defined as motor not moving.	Electrical cycles/seconds	1.0	0.01 to 10	Stepper commutation
6	Threshold of stepper angle	Electrical angle *4096/360	20	1 to 341	Binary search commutation
7	Minimum movement used to define the direction of motor movement	Electrical angle [degree]	2.0	0 to 180 (zero not included)	Binary search commutation
8	Automatic set current command at motor on in stepper mode (**UM** = 3)	Amperes	0	**-CL[1]** to **CL[1]**	Motor on in stepper mode
9	Reserved for compatibility		0		
10	Commutation test step verifying a movement (checking that motor is not blocked by a wall)	Electrical angle degree	30.0	-180 to 180	Stepper commutation
11	Stepper speed during the commutation	Electrical cycles/seconds	1.0	0.01 to 100	Stepper and sensorless commutation
12	Binary search verification		0		

Index	Description	Unit	Default	Range	In Process
	0 – Disabled				Binary search commutation
	1 – Enabled				
13	Unbalanced axis commutation		0		Stepper commutation
	0 – Disabled				
	1 – Enabled				
14	Time to increase stepper velocity from 0 to **SC[11]** (if **UM=1**)	Seconds	0.5	0 to 100	Sensorless commutation
15	Velocity to switch from stepper mode to closed loop	Counts/seconds	500000	0 to 1.0e11	Sensorless commutation
16	Velocity controller integral initial value after switching from initial phasing to vector control	[A]	0		Sensorless commutation

References

PL[], **CA[]**, **MO**, **UM**

SD – Stop Deceleration

SD[N] specifies the deceleration used during emergency stops.

CANopen/CoE

Elmo Alias Object: 0x3295

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	None
Range	1.....10^{14}
Index range	NA
Default	1e9
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

The **SD** value is the deceleration used by profilers during emergency stop caused by **ST[N]** command, and during emergency stops, such as Limit Switch and Stop Switch. It limits the total acceleration and deceleration of the profilers and auxiliary references. **SD** should be set to the maximum acceleration and deceleration that the motor can force on the load.

On **BG[1]**, **SD** overrides the CANopen maximum acceleration/deceleration (0x60C5 /0x60C6) and the Quick Stop deceleration (0x6085).

AC[1] or **DC[1]** will be overridden by **SD** values, if they are higher.

The **SD** value can be given in user-defined units specified by the **FC** command.

References

AC, **BG[]**, **DC**, **FC[]**, **PA[N]**, **SD[]**,

SF – Smooth Factor

SF specifies the motion smoothing factor, in milliseconds, for PTP and jogging.

CANopen/CoE

Elmo Alias Object: 0x3297

Attributes

Attribute	Description
Type	Short, Read/Write
Source	All
Restrictions	None
Range	Define in Table
Index Range	1, 2
Default	0
Unit modes	UM=2, 5, 6, 7
Non-Volatile	Yes
Axis Related	Yes

Remarks

The **SF** command smooths the motion to prevent sharp and high dynamic speed changes. **SF** actually builds the acceleration for the specified milliseconds, allowing the acceleration to affect the motion in moderate portions. When **SF**=0, all the requested acceleration is used by the profiler to build the speed command.

Maximal smoothing factor is actually limited by the moving average buffer of 2048 entries of **HS*TS** µsec each.

The **SF** is saturated to the maximum permitted value as a result of the 2048 buffer length. Setting **SF** for a higher value than the max range as denoted from the **HS*TS** will not result in error code.

Indices

The following table describes the **SF[]** entries.

Index	Description	Unit	Default	Range	Comments
1	Motion smoothing factor	1 millisecond	0	0 to $\frac{2048 * HS * TS}{1000}$	

Index	Description	Unit	Default	Range	Comments
2	Motion smoothing factor resolution extension	$HS * TS$ µsec	0	0 to $\left(\frac{1000}{HS * TS} - 1\right)$	Defines smoothing factor fractional

Example

To define smoothing factor of 2.6 msec (**TS**=100, **HS**=2) set its integer and fractional parts:

- Integer part **SF[1]**=2 [msec];
- Fractional part **SF[2]**=3 : $HS * TS * SF[2] = 0.6 msec$.

References

AC, DC, PA, PR, JV, JP

SG[N] – Signal Generator Parameters

SG[] sets the signal generator parameters.

CANopen/CoE

Alias Objects: 0x3298

Attributes

Attribute	Description
Type	Float
Source	All
Restrictions	According to index
Range	According to index
Index range	1 to 9
Default	0
Unit modes	All
Non-volatile	No
Axis related	No

Remarks

Injection points for the signal generator output are defined by **SA[4], SA[5], SA[12]** and **SA[13]** commands.

Indices

The following table details the **SG[]** entries.

<table>
<tr><th>Index</th><th colspan="2">Description</th><th colspan="2">Values</th><th>Notes</th></tr>
<tr><td rowspan="10">1</td><td colspan="2">Bitwise Configuration Parameter</td><td colspan="2"></td><td rowspan="10"></td></tr>
<tr><td>Bit</td><td>Description</td><td colspan="2"></td></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Invert direction</td><td>0</td><td>Do not invert</td></tr>
<tr><td>1</td><td>Invert the direction</td></tr>
<tr><td>1-6</td><td>Reserved</td><td colspan="2"></td></tr>
<tr><td rowspan="2">7</td><td rowspan="2">Waveform step flag</td><td>0</td><td>Sine waveform</td></tr>
<tr><td>1</td><td>Step waveform</td></tr>
<tr><td rowspan="2">8</td><td rowspan="2">Enable signal generator</td><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>Enable</td></tr>
<tr><td>9-15</td><td>Reserved</td><td colspan="2"></td></tr>
</table>

Index	Description	Values		Notes
2	Signal generator units	1	Current [A]	
		2	Velocity [UU/sec]	
		3	Position [UU]	
3	Amplitude sine A			**Units according to SG[2]**
4	Frequency sine A [Hz]	0 to 5000		
5	Amplitude sine B			Units according to **SG[2]**
6	Frequency sine B [Hz]	0 to 5000		
7	Offset of signal			Units according to **SG[2]**
8	Slope of offset	> 0		Units according to **SG[2]** per second
9	Start/Stop signal generation	0	Stop signal generation	
		1	Start signal generation	

References

SA

SN – Serial Numbers

SN specifies the serial numbers of the drive.

CANopen/CoE

Elmo Alias Object: 0x329F

DS-402 objects: 0x1018

Attributes

Attribute	Description
Type	unsigned long, Read – only
Source	All
Restrictions	None
Range	0 to 0xFFFFFFFF
Default	None
Index Range	4
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

None

Indices

The following table describes the **SN[*N*]** entries.

Index	Description
1	Vendor ID
2	Product code
3	Revision number
4	Serial number

References

SO – Servo Enabled

CANopen/CoE

When **SO** is set to 1, the Operation Enable state is reported in object Statusword **0x6041**

Elmo Alias Object: **0x32A0**

Attributes

Attribute	Description
Type	Short, Read
Source	All
Restrictions	None
Range	0 1 (when the drive is ready to handle a profiler set point)
Index range	NA
Default	
Unit modes	NA
Non-Volatile	No
Axis Related	Yes

References

MO

SP[N] – PTP Profiler Speed

SP[N] specifies the configured velocity normally reached at the end of the acceleration ramp during a profiled motion. **SP[N]** is valid for both directions of motion.

CANopen/CoE

Profile_velocity SP[1]: 0x6081

Elmo Alias Object: 0x32A1

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	Effective on the next call to **BG[N]**
Range	0.....10^{11}
Index range	1 to 4
Default	100000
Unit modes	**UM** = 5
Non-Volatile	Yes
Axis Related	Yes

Remarks

SP[N] is used during PTP motion to limit the speed of the profiler.

SP[1] overrides CANopen object 0x6081.

The **SP[N]** value can be given in user-defined units specified by the **FC** command.

Indices

The following table describes the profiles defined by the command index **(N)**.

Index	Profile name	Profile description
1	Main	The main drive profile (DS-402)
2	Superimposed	The profile is used for: • Superimposing the main profiler motion by the additional motion • Overcoming motion jumps when external reference generator is used (follower or ECAM)
3	Phasing	The profile is used during follower and ECAM mode to create a smooth phase shift in the master position of a slave axis.

Index	Profile name	Profile description
4	Socket profile	The profile running in a socket at control interrupt, can be used for all functions of the socket.

References

AC, **DC**, **PA**, **BG**, **FS**, **FC**

SR – Status Register

SR reports the status of different functions in the drive.

SR[1] reports the ELMO Status Register

SR[2] reports the Extended status register

SR[3] reports STO status register

CANopen/CoE

For compatibility reasons, Object 0x1002 (Manufacturer status register) returns the result of **SR[1]** for Axis 1. Object 0x3607 is the map-able alias object. Each axis can be read with 0x800 offset (AX2.SR[1] is read via 0x3E07.1).

Note:
Object 0x6041 and also **SW** command return the status of the DS-402 state machine and motion.

Elmo Alias Object: 0x32A3, 0x3AA3, 0x42A3, 0x4AA3, 0x52A3, 0x5AA3

Attributes

Attribute	Description
Type	Read, Bit-field
Source	All
Restrictions	None
Range	None
Default	None
Index Range	1 to 3
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

SR returns the drive status. Most of the information returned by **SR** can be retrieved by other commands, and the purpose of the **SR** command is to assemble this status in a single variable.

For the visual effect of the Status Indices below refer to the section 3.4.14 Status LED Blinks in the MAN-P-ADMINGUIDE manual.

Indices

SR[1] provides read access to ELMO Status Register

The following table details the different functions in the bit-field format.

Note:	Unused bits are reserved and are set to 0.

Bits	Description	Values		Notes
0 to 3	Amplifier Status - reports the instantaneous state of the power drive.	See details in the table below.		In cases in which the value differs from 0, the motor cannot be enabled; the drive status LED is constant RED. Digital outputs mapped to **AOK** will be set. See **OL[]** for details regarding the **AOK** function.
4	The servo is enabled. The reference command can be processed by the profiler (**SO**)	0	The servo is not enabled, **SO** = 0	In some cases, the **MO** value can be 1 while **SO** is still 0. The status differs between the two.
		1	The servo is enabled, **SO** = 1	
5	Reference Mode	0	External reference generator disabled	Refer to **RM** command
		1	External reference generator enabled	
6	A fault occurred while the motor was enabled.	See the **MF** command		This bit is cleared during the Motor Enable procedure.
7	If Elmo's homing or capture sequence is active.	0	**HM[1]** and **HF[1]** are not active	The command does not reflect the DS-402 homing mode or the DS-402 touch probe function.
		1	**HM[1]** or **HF[1]** is active	
8 to 11	Reports the actual profiler according to the motion mode.	0	No motion was selected	Bits actually reflect the "Mode of operation display" - object 0x6061 of CANopen DS-402 as well as **Object 6061.** Profilers are activated after **BG** or according to DS-402 profiling method. The **RS** command resets bits 8…11 , meaning "no profiler mode selected"
		1	Profile position mode (**PA\PR\JP**)	
		2	NA	
		3	Profile Velocity mode (**JV**)	
		4	Profile Torque mode (**TC**)	
		5	NA	
		6	Homing mode (**DH**)	

Bits	Description	Values		Notes
		7	Interpolated position mode (CANopen only)	
		8	Cyclic sync position mode (EtherCAT only)	
		9	Cyclic sync velocity mode (EtherCAT only)	
		10	Cyclic sync torque mode (EtherCAT only)	
		11	Cyclic sync torque with commutation angle (EtherCAT only)	
12	User Program is running	0	No user program or program is not running	Refer to **PS** command for more details about user program states.
		1	The user program is running	
13	Current Limit is on	0	No Current Limit, Peak current (**PL[1]**) can be applied	
		1	Current is limited to **CL[1]**	
14	Warning status	0	No warning detected, **WE** = 0	Warning bit in Object 0x6041 is set. Refer to **WE[]** command
		1	At least one warning has been detected, **WE** ≠ 0	
15	Threshold status	0	No threshold exceeded, **WT** = 0	Warning bit in Object 0x6041 is set. Refer to **WT[]** command
		1	At least one threshold has been exceeded, **WT** ≠ 0	
16 to 17	Recorder Status For : • Buffered mode • RT mode (buff 1)	0	The recorder is not active	
		1	Waiting for a trigger	
		2	The recorder has completed its task. Valid data is ready for uploading	
		3	Recording is now active. Data is being fetched by the drive	

Bits	Description	Values		Notes
18	Target Reached	According to **TR[]** and the relevant motion mode (**Object 6041**) for profiler Torque, Profile Velocity or Profile Position mode, the bit will be set:		**Note:** **BG** clears this bit. The bit will be set when motor was enabled and no motion is detected (**MS** = 0) Bit is not relevant in Cyclic Synchronous modes.
		0	The target is not reached	
		1	The target is within the **TR[]** boundaries	
19 to 20	Recorder Status **RT** mode (buff 2)	0	The recorder is not active	
		1	Waiting for a trigger	
		2	The recorder has completed its task. Valid data is ready for uploading	
		3	Recording is now active. Data is being fetched by the drive	
21	Reserved			
22	Motor On (**MO**)	0	Motor is Off (**MO=0**)	
		1	Motor is On (**MO=1**)	
23	Motor movement	0	Standstill. No movement	Refer to objects 0x606F velocity threshold (Object 606F) and 0x6070 velocity threshold time (Object 6070)
		1	Active movement. Not in standstill	
24	Reserved			
25	STO-1 Status	0	STO-1 not active	The motor is disabled and in safe state when the STO is “Not Active”. The motor cannot be enabled in this state. The motor can be enabled only if both STOs are in “Active” state.
		1	STO-1 Active	
26	STO-2 Status	0	STO-2 not active	
		1	STO-2 Active	

Bits	Description	Values		Notes
27	STO OK (TBD)	0	No error in STO diagnostics	This bit is cleared during power-up only. **SR[3]** provides information about the state of the STO.
		1	STO diagnostic failed indicating that there is a failure in the STO mechanism. The drive is in safe state, the motor cannot be enabled. For more details on the STO error status refer to **Object 0x2300 \ 0x2301**.	
28	The profiler stopped due to a digital input switch state.	Either a Hard Stop, FLS, RLS function caused the profiler to stop. The bit is set only after the profiler speed is settled.		Input indication can be retrieved via **IP** command.
29	Reserved, 0			
30	PTP buffer is full	PTP buffer can store up to 4 targets. Bit 30 is set when there are 4 targets in the buffer		

* Unused bits are reserved and are set to 0.

The following table details the indication of the amplifier status bits in **SR** command.

A CAN EMCY message will be transmitted if the motor was enabled prior to the indication.

SR bits 0 to 3 Value (Hex)	Description	Type CAN EMCY (Hex)	Notes
0	All OK		
3 (0x3)	Undervoltage: The amplifier is measuring the voltage which is smaller than minimum required voltage.	5 3120	• The minimum allowed value is reported in the **WI[38]** command. • The actual bus voltage is reported in **AN[6]**.
5 (0x5)	Overvoltage: The amplifier is measuring a voltage which is higher than the maximum allowed value.	5 3310	• The maximum allowed voltage is reported in **OV** command. • The actual bus voltage is reported in **AN[6]**.
7 (0x7)	Safety: One or two of the safety inputs are in safety state.	5 FF20	Safety indications are reported in **SR** bit 27 (TBD)
9 0x9	Sensor error: Sensor is not ready or in error state	81 7300	More details per the sensor that is used are in **W#[1]** command.

SR bits 0 to 3 Value (Hex)	Description	Type CAN EMCY (Hex)	Notes
11 (0xB)	Short Protection: The current has exceeded a range which is considered as a phase to phase or phase to ground short.	3 2340	This instantaneous fault is measured by the hardware and typically cannot be recorded or indicated outside of the **MF** command.
12 (0xC)	Abort: Abort or Inhibit switches are active	21 5441	Refer to **IL**[] command.
13 (0xD)	Drive over-temperature: The drive is sensing a temperature that exceeds the maximum allowed temperature limit indicated via **WS[27**].	9 4310	The actual temperature is reported via **TI[1]** in Celsius and **TI[2]** in Fahrenheit command. Thresholds can be modified via **TF[]** command.
14 (0xE)	Motor over temperature: The motor temperature exceeded **WS[28**].	9 4210	
15 (0xF)	Additional Abort: The additional Abort input is activated	81 5442	Refer to **IL**[] command.

SR[2] provides read access to Extended status register

The following table details the different functions in the bit-field format.

Bits	Description	Values	Notes
0 to 2	Halls status	Bit 0 – HALL-A status Bit 1 – HALL-B status Bit 2 – HALL-C status	When Hall A, Hall B and Hall C have the same value, error 37 “Hall location conflict” is reported with **EC** command
3 to 7	Reserved		
8	Absolute Home Status	0: The absolute sensor offset does not match offset stored in the drive. 1: The absolute sensor offset match the offset stored in the drive	0 is also received when feature is not supported.
9	Sensor commutation adjusted	0: Commutation variables stored does not match to drive commutation variables. 1: Commutation variables of sensor & drive match.	0 is also received when feature is not supported.
10 to 15	Reserved		
16 to 23	SIL status	bit 16: 0 SIL not supported	

Bits	Description	Values	Notes
		1 SIL is supported bit 17: SIL loaded and initialized (if no SIL this bit is 0) bit 18: SIL running bit 19: SIL run time error (**EE[14]**) bits 20-23: Reserved	
24 to 31	Reserved		

SR[3] reports STO status register

The following table details the different functions in the bit-field format.

Bits	Description	Values	Notes
0	STO1 status	0 – STO1 not active 1 – STO1 active	
1	STO2 status	0 – STO2 not active 1 – STO2 active	
2	STO diagnostic error	0 – No Error 1 – STO Error	In case of an error bits 8-15 and 16-23 are not 0.
3 to 15	Reserved		
16 to 23	STO1 diagnostic error	0 – No Error !=0 Hardware error in STO1	
24 to 31	STO2 diagnostic error	0 – No Error !=0 Hardware error in STO2	

References

MF, **BZ[]**

ST[N] – Stop Profiler

ST[N] stops the profiler in stop deceleration.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Command
Source	All
Restrictions	None
Range	None
Index range	1,4
Default	None
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Indices

The following table details the **ST[N]** entries.

Index	Description
1	Stops main, superimposed and phasing profilers together
4	Stops socket profiler

Remarks

The **ST[N]** command will stop the profiler (software) immediately with deceleration specified by the **SD[N]** command.

ST[N] has no effect over the direct reference command.

When main profiler is in the Torque mode (**TC**), **ST[1]** will stop the profiler by forcing torque command to 0.

References

BG, **SD**, **TC**, **UM**

SV[] – Save Parameters

SV command Saves parameter's data to a dedicated section on FLASH memory.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Not relevant
Source	All
Restrictions	According to array index
Range	Not Relevant
Index range	1
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

This procedure reads the: "ParametersImageFile.h" and saves it into a dedicated address in Flash, the parameter's image with current parameter's values from RAM.

In case an error occurred during Reset process, the returned error is: 'BAD_DATABASE', and the detailed error which contains: object index, sub-index, EC of the particular error, is retrieved via **CD[1]** command.

References

RS, LD, CD

SW – Statusword

SW[N] provides access to DS-402 Statusword (Objects **0x6041, 0x6841, 0x7041, 0x7841, 0x8041, 0x8841**).

CANopen/CoE

Elmo parameters objects: 0x32A8

Map able objects to TPDO: 0x6041, 0x6841, 0x7041, 0x7841, 0x8041, 0x8841

Attributes

Attribute	Description
Type	Unsigned short, Read only
Source	All
Restrictions	None
Range	0 to 65535
Default	0
Index Range	0,1
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Indices

Write access to **SW[n]** returns error code 23 "Not assigned command"

Read access to **SW[0]** returns 1 (number of sub-indexes)

Read access to **SW[1]** returns the DS-402 Statusword

Note:
After RS command was applied DS-402 Status word bits 10-13 are not relevant.

Remarks

None

Notes

None

References

CW

T1…8[N] – Socket Operation

T1…8[*N*] is used to perform operations on the sensor in the socket.

CANopen/CoE

Alias Objects: 0x32AD…0x32B4

Attributes

Attribute	Description
Type	Long Long, Write-Only
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 12
Default	Depends on variable
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

For all sensors in the following **Indices** table, when applying the Restrictions/Notes, it is recommended to check the applicability using a specific sensor type.

Indices

The following table describes the **T1…8[*N*]** entries.

Index	Description	Units	Restrictions/Notes
1	**T#[1]=[input value]** Sets the current single-turn position to the input value	Counts	The motor must be off. Applies to the following **absolute serial encoders**: **Panasonic & Nikon** [input value] = (0) – Reset the single-turn. Other values not allowed. **EnDAT** [input value] = (-1) – Reset the single-turn to 0. [input value] = (0-max_resolution) – Add the value to encoder's position.

Index	Description	Units	Restrictions/Notes
			Tamagawa [input value] – Can be set to any value; reset the single-turn anyway. **Acurolink and Openlink, Hiperface DSL**: [input value] = (0-max sensor resolution) – Set the position to the input value.
2	**#[2]=[input value]** Resets the multi-turn counter.		The motor must be off. Applies to the following **absolute serial encoders:** Panasonic, Nikon, Endat, Yaskawa, Tamagawa Any input value is acceptable results in clear of multi-turn counter
3	**T#[3]=[input value]** Reset the errors and/or warnings of the sensor		The motor must be off. Applies to the following **serial encoders:** Panasonic, Nikon, Tamagawa, Mitsubishi, Yaskawa, Acurolink & Openlink, EnDAT. Any input value is acceptable results in clear of sensor errors/warnings.
4	**T#[4]=[input value]** Reset socket position to Hiperface serial position reading		The motor must be off. Applies for Analog Hiperface. Input value can be set to any value.
5	**T#[5]=[input value]** Starting calibration of the resolver excitation signal.		The motor must be off. Applies for Analog Resolver. Input value can be set to any value.
6	**T#[6]=[input value]** Input value = 0 – Terminates sine profile sensor Input value = 1 – Launch sine profile sensor		Applies only to Sine Signal sensor.
7	**T#[7]=[input value]** Retrieve sensor status. The command is to be used when the basic sensor status doesn't provide sufficient information.		Applies to the following **serial encoders:** **Endat 2.1/2** Input value = (1) – Record sensor warnings Input value = (2) – Record sensor error

Index	Description	Units	Restrictions/Notes
			Nikon Input value any value; record the extended status. **Endat 3.0** Input value = (1) – Record extended sensor errors and warnings Input value = (2) – Record sensor specific error Input value = (3) – Record sensor specific warning **Other encoders** Output the error code automatically. (T#[7] not required) when error/warning occurs: Hengstler Acurolink Hengstler Openlink General Biss Hiperface DSL Analog Hiperface Tamagawa Panasonic Incremental Panasonic Mitsubishi Yaskawa EnDAT 2.1/2 EnDAT 3.0 In order to display the status recorded from the sensor: **W1…8[4]** – for errors **W1…8[5]** – for warnings
8	**T#[8]=[input value]** Perform sensor power up sequence		The motor must be off. Applies for EnDAT 2.1/2, EnDAT 3.0 & Yaskawa. Input value can be set to any value.
9	**T#[9]=[input value]** Simulate sensor error		**TP[6]=1** Input value – simulated error number.

<table>
<tr><th>Index</th><th>Description</th><th>Units</th><th>Restrictions/Notes</th></tr>
<tr><td>10</td><td>T#[10]=[input value]
OEM operations – Read / Write from/to user’s defined memory areas in the sensor.
<table>
<tr><th>Value</th><th>Command</th></tr>
<tr><td>1</td><td>Read Sensor’s Resolution</td></tr>
<tr><td>2</td><td>Init a memory block (Push stack)</td></tr>
<tr><td>3</td><td>Re-Init a memory block (Pop stack)</td></tr>
<tr><td>4</td><td>Read memory block by OEM number and Index (zero indexed)</td></tr>
<tr><td>5</td><td>Write memory block by OEM number and Index (zero indexed)</td></tr>
<tr><td>6</td><td>Read OEM memory size</td></tr>
</table></td><td></td><td>NResults for OEM operations stored in WS[107],
Errors of OEM command in WS[108] and EE[21] for error code.
Supported sensors:
General Biss
Endat 2.1/2
Endat 3.0
Hengstler Acurolink
Hengstler Openlink
Hiperface DSL
Analog Hiperface</td></tr>
<tr><td>11</td><td>T#[11]=[input value]
Enables the Quadrature encoder correction algorithm
<table>
<tr><th>Value</th><th>Command</th></tr>
<tr><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>Enable</td></tr>
</table></td><td></td><td></td></tr>
<tr><td>12</td><td>T#[12]=[input value]
Sets multi-turn counter for Yaskawa absolute sensor.</td><td></td><td>Applies for Absolute Yaskawa.
Input value limited to 65535.</td></tr>
<tr><td>13</td><td>T#[13]=[input value]
Gantry main position calculation method
<table>
<tr><th>Value</th><th>Gantry position calculation</th></tr>
<tr><td>0</td><td>Average between master and slave positions (X1+X2)/2</td></tr>
<tr><td>1</td><td>Equal to master position X1</td></tr>
</table></td><td></td><td>The motor must be off.

Relevant to sensor IDs 20 and 21 – Serial Exclusive and Gantry 4X.</td></tr>
<tr><td>14</td><td>T#[14]=[input value]
Gantry socket to be equal to the slave axis position.</td><td></td><td>The motor must be off.

Input value = (0) - Forces slave position
Applied only to sensor IDs 20 and 21 – Serial Exclusive and Gantry 4X.</td></tr>
</table>

Index	Description	Units	Restrictions/Notes
			The command is used during Gantry homing procedure.

For more details of the **T#[10]** Command, refer to the Platinum Administrative Guide.

References

TC – Current Command

TC specifies the current (torque) command for the current control loop.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	The motor must be on.
Range	Torque limits (**-PL[1]** to **+PL[1]**)
Index range	NA
Default	0, Volatile. Cleared automatically when **MO** = 1
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

The TC command sets the motor current (torque) command, in amperes, in all modes. When the drive is in position or velocity mode, the TC command transfers it to current mode.

TC commands are accepted in the range permitted by the present torque command limits (refer to the **PL[*N*]** and **CL[*N*]** commands). If TC is set greater than **CL[1]**, after a few seconds, the current limit of the servo drive will drop to **CL[1]**. In this case **LC** will indicate 1, notifying that there is a current saturation state.

If TC is higher than **CL[1]** while at the limit (**LC** = 1), the command will fail.

TC defines the active current reference value IQ.

References

MO, **UM**, **IQ**, **ID**, **CL[]**, **PL[]**,**MC**, **LC**

TF[] – Set Temperature Parameters

TF[] set the drive and motor temperatures fault detection parameters.

CANopen/CoE

Manufacturer specific profile area objects: 0x32BB

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	According to index
Index range	1 to 6
Default	According to index
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

Indices

Index	Description	Default	Range	Units	Comments
1	Drive temperature: fault hysteresis	5	0 to 20	°C	
2	Drive temperature: warning threshold	100	10 to 125	°C	**Equivalent to XT[4] in Gold Drive**
3	Drive temperature: warning hysteresis	5	0 to 20	°C	
4	Motor temperature: fault hysteresis	5	0 to 20	°C	
5	Drive temperature: filter constant	10	1 to 1000	Hz	
6	Motor temperature: filter constant	10	1 to 1000	Hz	
7	Drive temperature fault threshold. Temperature threshold can be set to lower value than the nominal threshold.	32767	0 to 32767	°C	

References

TI[]

TI[] – Get Temperature

TI[] return the drive and motor calculated temperatures.

CANopen/CoE

Manufacturer specific profile area objects: **0x32BE**

Attributes

Attribute	Description
Type	Short, Read Only
Source	All
Restrictions	None
Range	Signed short value
Index range	1 to 4
Default	None
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

The Notes column within the Indices table below describe the situation where the temperature sensor is not detected. This is a hardware configuration or malfunction in the hardware bit.

Indices

Index	Description	Units	Type	Notes
1	Drive temperature	°C	Short	"-99" if sensor is not defined
2	Drive temperature	°F	Short	"-146" if sensor is not defined
3	Motor temperature	°C	Short	"-99" if sensor is not defined
4	Motor temperature	°F	Short	"-146" if sensor is not defined
5	Sensor internal temperature	Sensor depended	Short	Supported sensors: Endat2.2, Endat3.0, Mitsubishi, Yaskawa. Set bit 17 in **S#[2]** to read the raw data of the internal temperature sensor.

Index	Description	Units	Type	Notes
6	Sensor external temperature	Sensor depended	Short	Supported sensors: Endat2.2, Endat3.0. Set bit 18 in **S#[2]** to read the raw data of the external temperature sensor.

References

TF[]

TM[] - High Resolution Time Stamp

TM[] command provides a read and write high resolution time stamp measured in microseconds. The command is non-axis related. **TM** is related to the High-Resolution time stamp object (0x1003) used by the CANopen network master in order to synchronize the devices on the network to a global high resolution timer. The synchronization method is done via the "Producer-Consumer" method where the drive is the Consumer, and the network master is the Producer.

The Time Stamp can be used by the drive User Programs of Platinum drives in the system to synchronize functions or execute commands in a time-based manner.

CANopen

Alias Objects: 0x32C2

Attributes

Attribute	Description
Type	Unsigned long, Read-write
Source	All
Restrictions	None
Range	0…2^{32}-1
Index range	1
Default	None
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

TBD

References

TBD

TR[] – Target Radius

TR[] specifies the threshold which determines whether the motion parameters have reached their targets.

CANopen/CoE: see the table below

Elmo Alias Object: 0x32C7

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	None
Range	See the table below.
Index range	1 to 8
Default	See the table below.
Unit modes	**UM** = 2,5, 6, 7
Non-Volatile	Yes
Axis Related	Yes

Remarks

The Target Reached indication informs the host that the load has reached the target with respect to the criteria of window and time around the target.

TR[] is the interface for specifying the target and time window in the target and is in conjunction with the CANopen target reached definition for position, velocity, and torque.

When the feedback count meets the desired target window for the window time, the Target Reached bit is set.

Note:
The Target Reached indication is the same for either profile velocity mode or profile position mode. Depending on the motion mode (object 0x6061), the relevant procedure is evaluated.

The Target Reached bit is indicated in CANopen status word object 0x6041 bit 10. It can also be retrieved by the **SW** command, which is a mirror image of the status word.

Target Reached is also indicated in **SR** register bit 18.

Target Reached is evaluated every **HS*TS** µsec. In between readings, no indication is given.

Indices

The following table describes the **TR[]** entries and the relevant CANopen objects.

Index	Description	Values [units]	Default	CANopen object	Comments
0	Reserved				
1	Target Position window	[UU]	100	0x6067	-1 not active
2	Target Position Window time	[msec]	20	0x6068	
3	Target Velocity window	[UU/sec]	100	0x606D	
4	Target Velocity window time	[msec]	20	0x606E	
5	Torque Window	[1/1000 of rated torque]	40	0x32C7.5, 0x36210x3621	
6	Torque Window Time	[msec]	20	0x32C7.6, 0x3622	
7	Velocity Threshold window	[UU/sec]	100	0x606F	
8	Velocity Threshold window time	[msec]	20	0x6070	

References

SR

TS –Control Loops Sampling Time

TS specifies the sampling time of the drive, in microseconds, which is used as the update time of all control loops.

CANopen/CoE

Elmo Alias Object: 0x32C8

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	MO = 0, new value activated after drive reboot
Range	50.0, 62.5, 100.0, 125.0
Index range	NA
Default	Defined in Drive Info table
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

TS is the sampling time of the position, velocity and current control loops.

Motor enable is not allowed after a **TS** change until drive reboot. **WF[28]** gives the actual sampling time.

The selection of **TS** is a compromise between high servo performance and the scan loop (background) operations, such as the user program and interpreter responses. A low **TS** enables the drive to achieve more control bandwidth, but at the same time, it increases the computational burden, so that less computing power remains for the executing interpreter and user program commands.

The drive does not allow an excessively low value for **TS** to prevent an overflow of the required computing power.

When **TS** is modified, the controllers loop gains must be returned to prevent instability of the controllers (current, velocity and position), which may damage the drive and/or the motor. However, when changing **TS** you must run the **SV** and **LD** commands to recalculate the controller parameters according to the new **TS**.

The PWM frequency (in Hz) is calculated using the formula **XP[2]**/(2***TS**), and the current ripple frequency at the motor is twice as large.

References

WS[], WF[], XP[], HS

UD – Double User Interface

UD[N] specifies the double user variable array.

CANopen/CoE

Elmo Alias Object: 0x32DD

Mappable Objects: TBD

Attributes

Attribute	Description
Type	Parameter, double, Read/Write
Source	All
Restrictions	None
Range	-1.0e307 - 1.0e307
Default	0.0
Index Range	1 to 48
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

This is the data type range 8 bytes "double" with range $1.7^{\pm 308}$ which can be used as a User Interface to describe a variable array.

Users can use the 48 indexed entries of **UD[]** to keep double values in the non-volatile memory.

UD[] can be used for the user program as well. Users can modify the value and manipulate the user program flow in a simple way.

References

UI[N], UF[N], UO[N]

UF – Float User Interface

UF[N] specifies the float user variable array.

CANopen/CoE

Elmo Alias Object: 0x32DF

Map able objects: 0x2EF1, 0x2F01

Attributes

Attribute	Description
Type	Parameter, Float, Read/Write
Source	All
Restrictions	None
Range	-1.0e37 to 1.0e37
Index Range	1 to 48
Default	0.0
Unit modes	All
Non-Volatile	Yes
Axis Related	No
Attribute	None

Remarks

This is the data type range 4 bytes "float" with range $3.4^{\pm 38}$ which can be used as a User Interface to describe a variable array.

Users can use the 48 indexed entries of **UF[]** to keep floating-point values in the non-volatile memory.

UF[] can be used for the user program as well. Users can modify the value and manipulate the user program flow in a simple way.

References

UI[N]

UI – Integer User Interface

UI[N] specifies the Long user variable array.

CANopen/CoE

Elmo Alias Object: 0x32E2

Mappable objects: 0x2EF0, 0x2F00

Attributes

Attribute	Description
Type	Parameter, Long, Read/Write
Source	All
Restrictions	None
Range	-2147483648 to 2147483647
Index Range	1 to 48
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No
Attribute	None

Remarks

This is the data type range 4 bytes “long” which can be used as a User Interface to describe a variable array.

Users can use the 48 indexed entries of **UI[]** to keep Long values in the non-volatile memory.

UI[] can be used for the user program as well. Users can modify the value and manipulate the user program flow in a simple way.

References

UF[N]

UM[N] – Unit Mode

UM[N] specifies the highest allowed control loop (**UM[N]**=1; **UM[N]**=2; **UM[N]**=5; N=1,2) or special control mode (**UM[1]**=3; **UM[1]**>5).

CANopen/CoE

Elmo Alias Object: 0x32E6

Attributes

Attribute	Description
Type	Short, Read/Write
Source	All
Restrictions	The motor must be off. Note that control loops can be freely switched without disabling the motor.
Range	0 to 7 (excluding 4)
Default	1,2
Unit modes	0
Non-Volatile	Yes
Axis Related	Yes

Remarks

UM[1] is valid for the stand alone drive axis or master axis in in a network drive master-slave configuration.

UM[2] is applied to the master axis and defines which control loops are allowed to run at the slave drive in attached drive master-slave configuration.

Values

The following table describes the possible values and the modes associated with them.

Index	Value	Description	Comments
1	0	Axis control is disabled, motor cannot be enabled	Defines master or stand-alone axis control loops
	1	Current (torque) control loop. The reference is set directly by **TC**. Values are processed immediately in the next control interrupt.	
	2	Velocity control loop. The reference is set by **JV**. The values are processed by the controller only after the next **BG**. Cyclic Synchronous velocity mode can also be used in this mode. Setting **TC** forces torque loop without changing UM.	

Index	Value	Description	Comments
	3	Stepper open loop. No control loop beside the current. Use open loop electrical degrees given by **PA** for absolute movement, use **PR** for relative movement, and use **JV** for constant speed movement. The units are 4096 ticks for 1 pole pair. **TC** must be set to excite the motor phases that allow the movement.	
	4	Reserved.	
	5	Position control loop, Single or Dual. **PA** and **PR** are used to reference the control loop. Cyclic Synchronous Position mode can be used in this mode as well. Setting **JV** will force velocity control loop and velocity profile reference. Setting **TC** will force torque control loop and the amount of torque. This method can be used for a welding application.	
	6	Stepper open loop. In stepper open loop no control loop beside the current, uses open loop electrical degrees. The Current during movement profile or standstill depends on the **HT[]** and **FF[]** commands. The units are 4096 ticks for one pole pair.	
	7	Stepper closed loop. The position control is performed by changing the stepper angle. The Current during movement profile or standstill depends on the **HT[]** and **FF[]** commands.	
2	0	Axis control is disabled, motor cannot be enabled	Defines attached slave's control loops
	1	Current (torque) control loop.	
	2	Velocity and current control loops.	
	5	Position, velocity and current control loop.	

References

UO – 64bit Long User Interface

UO[N] specifies the 64bit Long user variable array.

CANopen/CoE

Elmo Alias Object: 0x32E8

Map able objects: TBD

Attributes

Attribute	Description
Type	Parameter, Long Long, Read/Write
Source	All
Restrictions	None
Range	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Default	0
Index Range	1 to 48
Unit modes	All
Non-Volatile	Yes
Axis Related	No
Attribute	None

Remarks

This is the data type range 8 bytes “long long” which can be used as a User Interface to describe a variable array.

Users can use the 48 indexed entries of **UO[]** to keep Long Long values in the non-volatile memory.

UO[] can be used for the user program as well. Users can modify the value and manipulate the user program flow in a simple way.

References

UF[N], UI[N], UD[N]

US[] – User Saturation Parameters

US[] specifies the User Saturation Parameters for the controller output.

CANopen/CoE

Elmo Alias Object: 0x32EC

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	NA
Range	0 to 100
Index range	1 to 4
Default	100
Unit modes	According to array index
Non-Volatile	Yes
Axis Related	Yes

Remarks

The **US[]** command sets user saturation parameters to limit the controller output. For example: the current controller output is voltage, which is limited by **US[1]**.

Indices

The following table describes the available options for **US[]**.

Index	Description	Units	Range	Restrictions
1	Limit PWM (voltage) command (max range is **WS[54])**	% of max PWM range	0 to 100	All unit modes
2	Limit the current integral. The limit is in addition to the anti-windup limit, which works on maximum PWM. Limiting the integral reduces overshoots allowing faster settling time and keeping the use of integral for small errors.	% of max PWM range	0 to 100	All unit modes
3	Limit the speed integral. The limit is in addition to the anti-windup limit, which works on maximum current.	% of maximum current **PL[1]**	0 to 100	In position and velocity mode

Index	Description	Units	Range	Restrictions
	Limiting the integral reduces overshoots allowing faster settling time and keeping the use of integral for small errors.			
4	Limit gantry's Yaw current	% of **PL[1]**	0 to 100	Only in gantry master

References

PL[], **UM**, **WS[]**

VE – Velocity Error

VE returns the velocity tracking error.

CANopen/CoE

Elmo Alias Object: 0x3302

Attributes

Attribute	Description
Type	Float, Read-Only
Source	All
Restrictions	None
Range	NA
Index range	1
Default	0
Unit modes	Position and velocity: **UM** = 2, 5, 7
Non-Volatile	No
Axis Related	Yes

Remarks

VE reports the present velocity tracking error: **VE** = **DV[2]** – **VX**

If a dual loop is used, the units of the velocity error are in the position sensor and not in the velocity sensor.

If the absolute value of **VE** exceeds **ER[2]**, the motion is aborted,
and the motion fault code **MF** = 128 (0x80) is set.

If **MO** = 0, or if the speed controller is not used (**UM** = 1, 3), **VE** returns 0.

References

PE, **VX**, **DV[*N*]**

VH[]/VL[] – High/Low Reference Limit

VH[] and **VL[]** specify the minimum and maximum acceleration, speed and position reference limits.

CANopen/CoE

VH[1] – 0x60C5

VL[1] – 0x60C6

VH[2] – 0x607F

VL[3] – 0x607D.1

VH[3] – 0x607D.2

VH - Elmo Alias Object: 0x3305

VL - Elmo Alias Object: 0x3309

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	**VH[3]** > **VL[3]**
Range	According to array index
Index range	**VH[*N*]**: N=1, 2, 3 **VL[*N*]**: N=1, 3
Default	According to array index
Unit modes	**VL[1]**, **VH[1]**: **UM** = 2, 3, 5, 6, 7 **VH[2]**: **UM** = 2, 3, 5 **VL[3]**, **VH[3]**: **UM** = 5, 6, 7
Non-Volatile	Yes
Axis Related	Yes

Remarks

In position mode (**UM** = 5) motor movement is enabled in both directions within the defined position reference range [**VL[3]**…**VH[3]**]. Position commands outside of the range set by the **VL[3]** and **VH[3]**) values are not accepted.

If feedback has been extended beyond those limits, the motor can be enabled by the user (**MO** = 1), but the motion can only be in the direction towards the reference limit range.

Speed commands outside of the range (-**VH[2]**…**VH[2]**) are truncated.

VH[2] is also used as a limits (maximum and minimum) to velocity controller command in position loop.

The final velocity command limit is influenced by the following parameters: **VH[2]** (**Objects** 0x607F) and Max Motor Speed (object 0x6080). The logic of the velocity limit depends on the motor type:

- In the case of a rotary motor: Velocity Limit = min(**VH[2]**, object 0x6080)
- In the case of a linear motor: Velocity Limit = **VH[2]**

Indices

The following table describes the **VH[]/VL[]** entries.

Index	Description	Default	Range	Units
0	Reserved			
1	The profile acceleration is limited by **VH[1]**; the profile deceleration is limited by **VL[1]**	10^{14}	1 to 10^{14}	User-defined
2	The reference to the speed controller is limited to the range [-**VH[2]**…**VH[2]**].	2000000000	0 to 10^{11}	User-defined
3	The reference to the position controller is limited to the range [**VL[3]**…**VH[3]**].	0	-2^{52} to $(2^{52}-1)$	User-defined

Notes

- If **VH[3] = VL[3] = XM[2] = XM[1] = 0**, neither the software position limit nor the position range limit affects the motion.
- The **VH[]** and **VL[]** values should be given in user-defined position units specified by the **FC** command.
- In Cyclic Synchronous Position mode and in Interpolated Position mode (**Object 0x6060**), the **VH[3]\VL[3]** are ignored if bit 2 of the **PO[2]** is set. Note that this setting of the **PO[2]** is the factory default.

References

XM[*N*], HL[*N*], LL[*N*], PO[N]

VP – Vector Actual Position

VP reports the present positions of the all axes included in the vector group.

CANopen/CoE

Elmo Alias Object: 0x330D

Attributes

Attribute	Description
Type	Double, Read
Source	All
Restrictions	None
Range	-2^{52} to $(2^{52} - 1)$
Index range	NA
Default	According to the value of the sensors.
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

- Prefix **GR#.** is used to address the specific group, for example, **GR2.VP**.
- **GR#.VP** command reports
  - Vector Cartesian coordinates, if kinematic transformation is defined (MCS position - see **EV** command);
  - Vector axes feedback positions, if kinematic transformation is not defined (ACS position)

References

AY[], EV, K1[], K2[], GR.

VR – Software Firmware Version

VR retrieves the software drive's version.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	String, read-only
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	None
Index Range	1-10
Default	NA
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The **VR** command reports the SW and HW version as a string. The format is presented in the following table in accordance with sub-index:

Indices

The following table describes the **VR[*N*]** entries:

Index	Description	Format	Alias object
1	Firmware Version	<Product Name> <Software Version><Build Number> <Date> Example: "Guitar 02.00.00.16 B00 25Aug2019"	0x200A.1
2	Safety CPU SW Version	SafetySw <Software Version><Build Number> <Date> Example: "SafetySw 02.00.00.08 B18 25Aug2019"	0x200A.2
3	Safety CPU HW Version	<P><Primary CPLD version><S><Secondary CPLD version> Example: "P24.1 S24.1"	0x200A.3
4	STO Version	<Software Version> <Build Number> <Date> Example: "01.00.00.00 B18 14Nov2018"	0x200A.4
5	Boot Version	Titan-Boot <Software Version> <Build Number> <Date> Example: " Titan-Boot 02.00.01.00 B12 11Dec2018"	0x200A.5
6	Anan Version	<Anan Software Version> <Build Number> <Date>	0x200A.6

Index	Description	Format	Alias object
		Example: "01.00.00.00 B03 23Apr2019"	
7	Drive Name	<Drive Name> Example: "Guitar"	0x200A.7
8	EtherCAT PRU firmware version	EtherCAT PRU Rev: <revision number> Build: <build number> Example: " EtherCAT PRU Rev: 2 Build: 4EE"	0x200A.9
9	Drive user name	String set by DN command Example: " Elmo Drive"	0x200A.10
10	Drive barcode	<Barcode>	0x200A.11

Note

All indices that are not listed include the following logic:

- A Read operation will return, for example "Error: vr[11] Drive Error 3: Bad Index " string.
- A Write operation will return an error, for example "Error: vr[8] Drive Error 23: Command cannot be assigned"

References

DN[]

VS – Vector Status

VS reports the status of the vector motion with respect to the motion status of the vector axes.

CANopen/CoE

Elmo Alias Object: 0x3310

Attributes

Attribute	Description
Type	Unsigned short, Read
Source	All
Restrictions	NA
Range	0 to 4
Index range	NA
Default	0
Unit modes	UM = 1, 7
Non-Volatile	No
Axis Related	No

Remarks

- Prefix **GRn.** is used to address the specific group, for example, **GR2.VS**

The following table describes the available options for **VS**.

Index	Command	Values		Description
1	Vector status	0	Target reached	Vector motion profile command has reached the target and all group axes feedbacks have reached their targets
		1	Profile done	Vector motion profile command has reached the target, but group axes feedbacks have not reached their targets yet
		2	Profile on motion	Vector motion profile is initiated
		3	Motor off	At least one of the group axes motor is disabled
		4	Vector off	Group vector is in disabled mode

References

AY[], GR, EV.

VU – Main Feedback Velocity in User-Defined Units

VU retrieves the velocity of the position sensor or velocity sensor, in user-defined units.

CANopen/CoE

Actual Velocity: 0x606C

Elmo Alias Object: 0x3312, 0x606C

Attributes

Attribute	Description
Type	Double, Read-Only
Source	All
Restrictions	None
Range	-2^{52} to $(2^{52} - 1)$
Index range	NA
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

In a velocity loop or a position loop with one sensor **VU** returns values in user-defined units.

In a dual loop, **VU** returns values in user-defined units of the position sensor. The sensor that returns is according to object 0x606A. If it is equal to zero, **VU** returns the velocity of the position sensor. If it is equal to 1, **VU** returns the velocity of the velocity sensor, but multiplied by a factor to be in the same units as the position sensor.

References

VX, **FC[]**

VX – Main Feedback Velocity in Counts per Second

VX retrieves the velocity of the position sensor or velocity sensor, in counts per seconds.

CANopen/CoE

0x606CElmo Alias Object: 0x3315

Attributes

Attribute	Description
Type	Double, Read-Only
Source	All
Restrictions	None
Range	-2^{52} to $(2^{52} - 1)$
Index range	NA
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

In a velocity loop or a position loop with one sensor **VX** returns values in counts per seconds.

In a dual loop, **VX** returns values in counts/sec in the units of the position sensor. The sensor that returns is according to object 0x606A. If it is equal to zero, **VU** returns the velocity of the position sensor. If it is equal to 1, **VU** returns the velocity of the velocity sensor, but multiplied by a factor to be in the same units as the position sensor.

References

VU, **FC[]**

W1…8[] – Warning and Error of Socket Report

W1…8[] obtains the warning and error status of the socket number.

CANopen/CoE

Alias Objects: 0x3319…0x3320

Attributes

Attribute	Description
Type	Unsigned Long, Read Only
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 5
Default	
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The **W1...8[]** commands obtains the status of the sensor attached to socket. The number of socket is denoted by a number 1 to 8 or the pound key (#), e.g., **W1** means status of sensor attached to socket 1. There are up to 8 sockets.

Interpretation of the value is according to the sensor in the socket.

Error Codes

Refer to chapter **Sensor Error Codes** for the error descriptions related to the error code.

Indices

The following table details the **W1…8[]** entries.

Index	Description	Default	Restrictions/Notes
1	Socket latched errors	0	Cleared when axis that uses sensor is Enabled - motor on
2	Socket warnings	0	Warnings reported from the socket analysis.
3	Socket active errors	0	Error reported from the socket analysis. These errors are mask-able – can be converted to warnings.

Index	Description	Default	Restrictions/Notes
4	Fetch the extended sensor status errors automatically	0	Reflects the error register from sensor's datasheet. Applies to the following **serial encoders:** General Biss Endat 2.2 Endat3 Acurolink Openlink Hiperface DSL Analog Hiperface Tamagawa Nikon Panasonic Incremental Panasonic Mitsubishi Yaskawa See **T1…8[7]** for further details.
5	Fetch the extended sensor status warnings	0	Reflects the warning register from sensor's datasheet. Applies to the following **serial encoders:** General Biss Endat 2.1/2.2 Endat3 Hiperface DSL Tamagawa Nikon Panasonic Mitsubishi Yaskawa See **T1…8[7]** for further details.
6	Get Yaskawa Multiturn value	-	Applies specific to Yaskawa Absolute Encoder with Multiturn
7	Get Endat 2.1/2.2 lower bits of Datum Shift value (32bit)	-	Applies only to Endat 2.1/2.2, this value will used in Safety SRA parameters.
8	Get Endat 2.1/2.2 higher bits of Datum Shift value (32bit)	-	Applies only to Endat 2.1/2.2, this value will used in Safety SRA parameters.

References

S#[]

WB[] – Bulk Commands

WB[*N*] allows setting of groups of subsequent commands and to retrieve their values.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Read / Write
Source	all
Restrictions	See restriction chapter
Range	None
Index range	1 to 30
Default	Default should include 4 first indices with up to 20 commands in each entry (depending on the status required by the EAS).
Unit modes	All
Non-Volatile	No
Axis Related	No

Indices

Index	Description	Type	Values		Restrictions
1-4	A string of Required commands to get their values separated by comma ‘,’ and using a semi-column for delimiter.	char	string		The entered string should be for example : WB[1] = SR, IQ, PE;
11-14	The data returned by the sequence of commands set respectively by **WB[1]** to **WB[10]**.	char	The data returned to the OutStream is binary data (as saved in the memory – LSB first).		Read only. The form of the returned data is : • Header (signature :A5, size) • Data (binary) • Footer (Checksum16bit, signature:5A) • Little Endians
21-24	Types of the subsequent commands set respectively by **WB[1]** to **WB[10]**.	char	1	char (1 byte)	Read only
			2	Unsigned char (1 byte)	
			3	Short (2 bytes)	

Index	Description	Type	Values		Restrictions
			4	Unsigned short (2 bytes)	
			5	Long (4 bytes)	
			6	Unsigned long (4 bytes)	
			7	Float (4 bytes)	
			8	Long long (8 bytes)	
			9	Unsigned long long (8 bytes)	
			10	Double (8 bytes)	

References

WE[] command – Warning Event

The **WE[]** command, Warning Events is a bit field which indicates when a warning event related to its specific bit occurs.

CANopen/CoE

NA

Attributes

Attribute	Description
Type	Unsigned long, Read only
Source	All
Restrictions	NA
Range	0…(2^{32})-1
Index range	1
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

The bits are mapped in the same order as in the **MF** command. Every bit in **WE[]** is set to 1 when the related **AE[4]** bit is set to 1 (meaning: treat fault as warning) and a related fault event occurs. In this case, the related **MF** bit should not be set. The Warning event does not disable the servo. A Warning event is treaded with non-latched policy, meaning a **WE[]** bit is auto reset when the conditions do not match this event anymore.

When **WE[]** indicates any bit set to 1, additional indications are presented in:

- **SW**: bit 7 - Warning
- **SR**: bit 14 – Warning Event

Indices

The following table describes the available options for **WE[]**.

Index	Description	Type	Values	Restrictions
1	Warning Event	Unsigned long	Bit field in the same order as in **MF**	

References

MO[], SO[], AF[], AE[], MF[], SW[], SR[], WT[]

WF – Miscellaneous Float Reports

The **WF** command provides information about the state and internal variables of the drive. Mainly relevant for Elmo internal use only.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Double, Read Only
Source	All
Restrictions	According to array index
Range	According to array index
Index range	Refer to table below
Default	
Unit modes	All
Non-Volatile	No
Axis Related	According to array index

Remarks

WF provides service personnel with a fairly comprehensive report, but these details are not normally required for defining an application.

Indices

The following table describes the **WF** entries.

Index	Description	Units\Values	Axis Related	Notes
4	PWM frame	[PWM clocks]	Yes	100***TS/XP[2]** Quartet: 250* **TS/XP[2]**
5	Bits per ampere	[bits/A]	Yes	MaxADCvalue/**MC**
6	Phase 1 calibration factor		Yes	From **DI**
7	Phase 2 calibration factor		Yes	From **DI**
8	Phase 3 calibration factor		Yes	From **DI**
9	Phase 4 calibration factor		Yes	From **DI**
10	Analog input 1 (primary ADC)	[V]	No	
11	Analog input 1 (secondary ADC)	[V]	No	

Index	Description	Units\Values	Axis Related	Notes
12	Phase 1 current ratio	[A/bits]	Yes	**WF[6]*MC**/MaxADCvalue
13	Phase 2 current ratio	[A/bits]	Yes	**WF[7]*MC**/MaxADCvalue
14	Phase 3 current ratio	[A/bits]	Yes	**WF[8]*MC**/MaxADCvalue
15	Phase 4 current ratio	[A/bits]	Yes	**WF[9]*MC**/MaxADCvalue
16	Rated torque factor		Yes	For safety use
17	Correlation in current experiment Σ(0.25*fTorqueCmdOld^2)		Yes	Relevant for Elmo internal use only
19			Yes	Relevant for Elmo internal use only
21	Filtered RMS current	Amperes	Yes	
23	Factor to convert analog input bits to volts. typically: 10.0/(2048.0 - 205.0)	Volts/bits	No	Relevant A2D set via **TW[25]**
24	Returns the internal offset of Analog Input	Volts	No	Relevant A2D set via **TW[25]**
28	Actual control cycle time (TS)	microseconds	No	
29	Recording cycle time (TS)	microseconds	No	For compatibility
31	Gantry Yaw loop cycle time (TS)		No	For compatibility
33	The instantaneous saturated high torque value	Amperes	Yes	Nominal **PL[1]**
34	The instantaneous saturated low torque value.	Amperes	Yes	Nominal **CL[1]**
35	Gantry differential position (yaw) feedback	UU	Yes	
36	ADC value at nominal Bus voltage	ADC	No	
38	Main position before Error Correction		Yes	
39	Main position after Error Correction		Yes	
50	Sum of Reference Signal used in tuner Σ(fTorqueCmdOld)	Internal torque units	Yes	Relevant for Elmo internal use only

Index	Description	Units\Values	Axis Related	Notes
53	Convert values from internal units to bus voltage	Volts/bits	No	**BV**/MaxADCvalue
54	Maximum PWM (voltage) command allowed in drive	[PWM clocks]	Yes	
55	Position Loop cycle time (TS)	microseconds	No	For compatibility
56	Minimum PWM (voltage) command allowed in drive	[PWM clocks]	Yes	
57	PWM (voltage) command range	[PWM clocks]	Yes	**WS[54]**-**WS[56]**
69	Average analog input value	[Volts]	No	Provide Value after measurement activation by **TW[94]**
82	Identification frequency	[Hz]		Relevant for Elmo internal use only
91	Signal #1 sine coefficient		No	Provide Valid info after **TW[81]**=N. N is a bit field that determines the vectors to identify. Relevant for Elmo internal use only
92	Signal #1 cosine coefficient		No	
93	Signal #2 sine coefficient		No	
94	Signal #2 cosine coefficient		No	
95	Signal #3 sine coefficient		No	
96	Signal #3 cosine coefficient		No	
97	Signal #4 sine coefficient		No	
98	Signal #4 cosine coefficient		No	
99	Signal #1 quality function		No	
100	Signal #2 quality function		No	
101	Signal #3 quality function		No	
102	Signal #4 quality function		No	
103	Offset of signal #1		No	
104	Offset of signal #2		No	
105	Offset of signal #3		No	
106	Offset of signal #4		No	
107	Dynamic brake instantaneous Bemf	Volts	Yes	
108	Instantaneous calculated power supplied by the drive	Watts	Yes	

References

WS – Miscellaneous Reports

The **WS** command provides information about the state and internal variables of the drive.

CANopen/CoE

Elmo_parameters_objects: 0x3334

Attributes

Attribute	Description
Type	Long, Read only
Source	All
Restrictions	None
Range	None
Index range	Refer to the table below
Default	None
Unit modes	All
Non-volatile	No
Axis related	According to array index

Remarks

WS[*N*] provides service personnel with a fairly comprehensive report, but these details are not normally required for defining an application.

Indices

The following table describes the **WS** entries.

Index	Description	Units Values	Axis Related	Notes
1	Number of Axes	None 1 to 6	No	
2	ANAN enabled	None 0 to 0xFFFFFFFF	No	For debug purposes, valid for ANAN versions only
3	CPU main clock frequency.	Hz 3e8 to 5e8	No	Refer to product user manual

Index	Description		Units Values	Axis Related	Notes
4	Application compilation type (FW based):		None 0 to 3	No	SIL is enabled in Elmo standard mode
	0	Elmo standard compilation			
	1	LAUNCHER Connect 2 DC motors to 3 phase drive (only one of them can be enabled)			
	2	ANAN			
5	**DI** compilation type		None 0 to 0xFFFFFFFF	No	Same description as **WS[4]**
6	ECAT AL Status Register:		None 0 to 0x3F	No	
	Bit 0	Initialization state			
	Bit 1	Preoperational state			
	Bit 2	Safe operational state			
	Bit 3	Operational state			
	Bit 4	State change error: 0: State changed without error 1: Error occurred			
	Bit 5	ID loaded indication: 0: ID was not loaded 1: ID loaded			
	Bits 6,7	Reserved, 0			
7	Sin table size		None 4096		Returns value of sin table 4096
8	Hardware board type:		None 0 to 32		
	0	Gold SCORE			
	1	Gold REVA			
	2	Gold REVC			
	3	Gold REVD			
	4	Platinum REVA			
	5	Platinum REVB			
	6	Platinum REVC			
	7-11	Platinum Reserved			

Index	Description		Units Values	Axis Related	Notes
	12	Quartet RevA			
	13-19	Quartet Reserved			
	20	Titanum RevA			
	21-32	Reserved			
9	ADC instant delay				TBD
10	Actual Gantry communication configuration (**GG[1])** after drive reboot:		None	No	
11	Reserved				
20	Max recorder length in bytes.		None 5242880	No	Returns default value
21	Actual number of recorder length per signal in bytes.		None 0 to 5242880	No	Returns recorder length in bytes per one mapped signal
23	**O**vervoltage factory setting (DI)		Volts	No	Depends on board type
24	Actual overvoltage defined with command **XP[1]**		Volts 0 to **WS[23]**	No	Depends on board type. If **XP[1]** > **WS[23]**, then **WS[24]** saturated to value of **WS[23]**
25	**U**nder voltage –factory setting (DI)		Volts	No	Depends on board type
26	Actual under voltage defined with **XP[5]**		Volts 0 to **WS[25]**	No	Depends on board type. If **XP[5]** < **WS[25]**, then **WS[26]** saturated to value of **WS[25]**
27	Drive temperature threshold		Celsius	Yes	T= Max(DI, drive info) MIN(**TF[7]**, T) Derived from DI, derating & **TF[]**.
28	Motor temperature threshold		Celsius	Yes	Factory setting (DI)
30	Bits 0-7	Product Code (See section Product Code)	None 0 to 0xFFFFFFFF	No	
	Bits 8-11	Reserved			
	Bits 12	Unused			
	Bit 13	ECAT PHY 1: No ECAT PHY connected			

Index	Description		Units Values	Axis Related	Notes
	Bit 14	Gold Product Family (always 1 for Platinum drive)			
	Bit 15	Platinum\Titanium Product Family (always 1 for Platinum\Titanium drive)			
	Bit 16	Safety 1: Safety disabled			STO indication still valid in non-safe configuration.
	Bit 17	Dual Use			Based on "Dual Use" declaration limited to 599Hz.
	Bit 18	STO only			Safety diagnostics is active. Only STO is permitted. Only if bit 16 is set to 0.
	Bit 19	SIL-Enabled			0 – SIL not operational 1 – SIL operational if included in version
	Bit 20	EtherCAT OUT is programmed to Ethernet by default (only on drives without additional PHY)			In case EtherCAT drive without additional PHY: 0 – normal reset behavior 1- Reset will set **PP[23]**=0xa55a by default.
	Bit 21	5V Hall enable monitoring			Indication for EAS only: 0- Do not monitor 5V Hall 1- Monitor 5V Hall
	Bit 22	Yaskawa permitted			0 – do not permit Yaskawa sensor 1 – Yaskawa permitted
	Bit 23	VL_DETECT logic enabled			0- VL_LOGIC not used VL_LOGIC enabled
	Bits 24-31	Unused			
31	Safety SRA CRC			No	

Index	Description		Units Values	Axis Related	Notes
32	Drive fault reason (Sample Exception)		None 0x0 to 0x80000000	Yes	Refer to **MF** command description
33	SIL sample time		µSecs	No	
37	User program max thread				
39	EtherCAT slave control register readout		None	No	The register is set via **TW[39]** Bits 0-15: address Bits 16-19: num of bytes
41	Socket used by axes (bit field):		None 0 to0x1E	No	
	Bit 0	Always 0			
	Bit 1	Set to 1 if socket 1 is used			
	Bit 2..8	Set to 1 if socket 2..8 is used			
	Bits 9-15	Reserved, always 0			
42	Socket with latched error (bit field):		None 0 to 0x1E	No	Refer to **W1** to **W8** commands for error details
	Bit 0	Always 0			
	Bit 1	Set to 1 if latched error in socket 1			
	Bit 2..8	Set to 1 if latched error in socket 2..8			
	Bits 9-15	Reserved, always 0			
43	Socket used by axes with latched error (bit field):		None 0 to 0x1E	No	Refer to **W1** to **W8** commands for error details. **Note:** Only used by axes socket are indicated
	Bit 0	Always 0			
	Bit 1	Set to 1 if latched error in socket 1			
	Bit 2..8	Set to 1 if latched error in socket 2..8			
	Bits 9-15	Reserved, always 0			
44	Socket with active error (bit field):		None 0 to 0x1E	No	Refer to **W1** to **W8** commands for error details
	Bit 0	Always 0			
	Bit 1	Set to 1 if active error in socket 1			

Index	Description		Units Values	Axis Related	Notes
	Bit 2..8	Set to 1 if active error in socket 2..8			
	Bits 9-15	Reserved, always 0			
45	Socket used by axes with active error (bit field):		None 0 to 0x1E	No	Refer to **W1** to **W8** commands for error details **Note:** Only used by axes socket are indicated
	Bit 0	Always 0			
	Bit 1	Set to 1 if active error in socket 1			
	Bit 2..8	Set to 1 if active error in socket 2..8			
	Bits 9-15	Reserved, always 0			
46	Socket with warning (bit field):		None 0 to 0x1E	No	Refer to **W1** to **W8** commands for error details
	Bit 0	Always 0			
	Bit 1	Set to 1 if warning in socket 1			
	Bit 2..8	Set to 1 if warning in socket 2..8			
	Bits 9-15	Reserved, always 0			
47	Socket used by axes with warning (bit field):		None 0 to 0x1E	No	Refer to **W1** to **W8** commands for error details **Note:** Only used by axes socket are indicated
	Bit 0	Always 0			
	Bit 1	Set to 1 if warning in socket 1			
	Bit 2..8	Set to 1 if warning in socket 2..8			
	Bits 9-15	Reserved, always 0			
50	Axis is used as Gantry master and in override mode		None	Yes	Reference from host, Gantry master is overridden
51	Wizard state counter used in tuner		None 0 to 0xFFFF	No	
52-53	Safe EnDAT sensor serial number		None	No	64 bits
54-55	Safe EnDAT Encoder ID		None	No	64 bits
56-57	Safe EnDAT3 Encoder ID		None	No	64 bits
58-59	Safe EnDAT3 serial number		None	No	64 bits

Index	Description	Units Values	Axis Related	Notes
69	MPU frequency	[Hz]	No	From **DI**
71	Shunt Hysteresis	mSec	No	INTERNAL From DI
73	Output Compare Last Set Mode Bits 0…7 : Last set **OC[1]** write value Bits 8…15 Last set **OC[21]** write value	None	Yes	
80	Command **WS[80]** returns 8 configuration bits where: Bits 0-7 | Configuration Bits (see table Configuration Bits below) Bits 8-31 | 0	None	No	
81	Command **WS[81]** returns 8 bits of ECAT address, on a condition that EtherCAT switches are used. If switches are not used, **WS[81]** returns 0	None Bit field	No	
90 - 97	Capture IN 1 – Capture IN 8 resource	None	No	See section **Details** below
98 - 101	Capture Time 1 – Capture Time 4 resource	None	No	See section **Details** below
102	Output Compare Outputs resource used per axis	None	Yes	See section **Details** below
103	Output Compare new resources	None	No	See section **Details** below
104	Output Compare active drivers	None	No	See section **Details** below
105	Output Compare used resources	None	No	See section **Details** below
106	Capture manager status	None	No	See section **Details** below
109	Output Compare Outputs resource used	None	No	See section **Details** below
110	EnDAT3 Read error register	None	No	Result of **T1[7]**=3
111	Temperature readout in A2D	None	No	See section **Details** below
112	Axis related	None	No	See section **Details** below

Index	Description	Units Values	Axis Related	Notes
	Platinum Quartet drive Output Compare Port C resources.			
113	Non axis related **Platinum Quartet drive** Output Compare Port C resources	None	No	See section **Details** below
133	Titanium drive X1 Ports A, B, C used per axis by **OC[1]** or **OC[21]**	None	Yes	See section **Details** below
134	Titanium drive X2 Ports A, B, C used per axis by **OC[1]** or **OC[21]**	None	Yes	See section **Details** below
135	Titanium X1 Ports A, B, C used non-axis by **OC** or **GA**	None	No	See section **Details** below
136	Titanium X2 Ports A, B, C used non-axis by **OC** or **GA**	None	No	See section **Details** below

Details

<table>
<tr><th>Index</th><th>Description</th><th>Details</th></tr>
<tr><td>90 - 97</td><td>Capture IN 1 – Capture IN 8 resource</td><td>The commands presents currently active Capture position HW resources. The command is bit field described in following table:
<table>
<tr><th>Bits 31 - 16</th><th>Bits 15 - 8</th><th>Bits 7 - 0</th></tr>
<tr><td>0 - Reserved</td><td>Axis number:
1 – axis 1
…
6 – Axis 6</td><td>The Service currently occupied the HW Capture:
1 - DS402 Homing
2 - TP1
3 - TP2
4 - HM
5 - HF</td></tr>
</table></td></tr>
<tr><td>98 - 101</td><td>Capture Time 1 – Capture Time 4 resource</td><td>Reserved for Capture Time 1 – Capture Time 4</td></tr>
<tr><td>102</td><td>Output Compare Outputs resource used axis related.

Note:
For Quartet drive and for Titanium drive bits 0 – 15 only are relevant.</td><td>Command is bit field that differently presents Digital Outputs and Port C resources used (per axis) by Output Compare (if enabled)
<table>
<tr><th>Bits 31-22</th><th>Bits 21-19</th><th>Bits 18-16</th><th>Bits 15-8</th><th>Bits 7-0</th></tr>
<tr><td>Reserved, set to 0</td><td>PortC_A, PortC_B, PortC_Index , used by OC[21] command.
Bit 19=1 – PortC_A is used by</td><td>PortC_A, PortC_B, PortC_Index, used by OC[1] command.
Bit 16=1 – PortC_A is used by</td><td>Outputs 8 – 1, used by OC[21] command Bit 8=1 – Output 1 is used by Output Compare (OC[21]) …</td><td>Outputs 8 – 1, used by OC[1] command.
Bit 0=1 – Output 1 is used by Output</td></tr>
</table></td></tr>
</table>

Index	Description	Details				
			OC[21] command Bit 21=1 – PortC_Index is used by **OC[21]** command	Output Compare (**OC[1]** or **OC[21]**) … Bit 18=1 – PortC_Index is used by Output Compare (**OC[1]** or **OC[21]**)	Bit 15=1 – Output 8 is used by Output Compare (**OC[21]**)	Compare (**OC[1]**) … Bit 7=1 – Output 8 is used by Output Compare (**OC[1]**)
103	Output Compare new resources	The commands presents HW resources occupied by new received **OC[1]/OC[21]** command. The command is bit field described in following table:				

Bits 31 - 14	Bits 13 - 8	Bits 7 - 4	Bits 3 - 0
0 - Reserved	Axis number: 0 – Axis 1 5 – Axis 6	Advanced Output Compare HW module number: 0 – HW AOC1 3 – HW AOC4	Output Compare HW module number: 0 – HW OC1 3 – HW OC4

Index	Description	Details
104	Output Compare active drivers	The commands presents currently active output compare drivers in every HW channel. The command is bit field described in following table:

Bits	31 - 28	27 - 24	23 - 20	19 - 16	15 - 12	11 - 8	7 - 4	3 - 0
Platinum Drive	0	0	0	0	0	Slave Advanced HW Drive	HW2 Drive	HW1 Drive
Platinum Quartet Drive	0	0	Slave Advanced HW3HW4 Drive	Slave Advanced HW1HW2 Drive	HW4 Drive	HW3 Drive	HW2 Drive	HW1 Drive
Titanium Drive					HW4 Drive (X2 Eqep2)	HW3 Drive (X1 Eqep0)	HW2 Drive (X2 Quad)	HW1 Drive (X1 Quad)

Possible values of the drivers:

The driver name	The driver explanation	Value
Empty	Empty function, no active drive	0
Start driver	The driver function is active when the first position is in the distance more than -32768 in negative direction or 32767 in positive. When the condition is no longer met the drive function is changed to another driver in accordance with the activated Output Compare mode	1

Wait driver	The driver function is active when the first position is already reached but the next position is in the distance more than -32768 in negative direction or 32767 in positive. When the condition is no longer met the drive function is changed to another driver in accordance with the activated Output Compare mode	2
Non table mode driver	The driver function is active when Absolute Start Position with Fixed Delta Position Mode **OC[1]**=1 is activated and the condition of -32768,32767 is met. When the condition is no longer met the drive function is changed to Wait driver.	3
One direction table mode driver	The driver function is active when Table Mode **OC[1]**=3 or 4 and direction parameter is set to "one direction" **OC[10]**=1 or 2, is activated and the condition of -32768,32767 is met. When the condition is no longer met the drive function is changed to Wait driver.	4
Both direction table mode driver	The driver function is active when Table Mode **OC[1]**=3 or 4 and direction parameter is set to "both direction" **OC[10]**=0, is activated and the condition of -32768,32767 is met. When the condition is no longer met the drive function is changed to Wait driver.	5
HW auto increment mode driver	The driver function is active when Absolute Start Position with Fixed Delta Position Mode with HW auto increment **OC[1]**=6 is activated and the condition of -32768,32767 is met	6
Slave advanced	The driver function is active in Advanced mode for Slave channel	7
Wait for STOP	The driver function is active when all positions are already reached but Output Compare process is not completed yet due to not all repetition pulses are generated by HW	8

<table>
<tr><th>Index</th><th>Description</th><th colspan="4">Details</th></tr>
<tr><td rowspan="3">105</td><td rowspan="3">Output Compare used resources</td><td colspan="4">The command presents HW resources occupied by all active Output Compare services. The command is bit field described in following table:</td></tr>
<tr><td>Bits 31 - 14</td><td>Bits 13 - 8</td><td>Bits 7 - 4</td><td>Bits 3 - 0</td></tr>
<tr><td>0 - Reserved</td><td>Axis which has an active AOC or OC:
Bit 8 = 1; Axis 1
…
Bit 13 = 1; Axis 6</td><td>Advanced Output Compare HW module active:
Bit 4 = 1; HW AOC1 is active
…
Bit 7 = 1; HW AOC4 is active</td><td>Output Compare HW module active:
Bit 0 = 1; HW OC1 is active
…
Bit 3 = 1; HW OC4 is active</td></tr>
</table>

<table>
<tr><th>Index</th><th>Description</th><th>Details</th></tr>
<tr><td>106</td><td>Capture manager status</td><td>The command presents Capture Manager status. The command is bit field described in following table:
<table>
<tr><th>Bits 31 - 24</th><th>Bits 23 - 16</th><th>Bits 15 - 8</th><th>Bits 7 - 0</th></tr>
<tr><td>0 - Reserved</td><td>The socket which has an active Capture position module:
Bit 16 = 1; Socket 1
…
Bit 23 = 1; Socket 8</td><td>Capture position HW module active:
Bit 8 = 1; Capture position 1 is active
…
Bit 15 = 1; Capture position 8 is active</td><td>RT Capture manager active status:
Value = 0; not active
Value = 1; active</td></tr>
</table></td></tr>
<tr><td>109</td><td>Output Compare Outputs resource used non axis related

Note:
For Quartet drive and for Titanium drive bits 0 – 15 only are relevant</td><td>Command is bit field that presents Digital Outputs and Port C resources used differently by Output Compare (if enabled), GO and GC commands
<table>
<tr><th>Bits 31-22</th><th>Bits 21-19</th><th>Bits 18-16</th><th>Bits 15-8</th><th>Bits 7-0</th></tr>
<tr><td>Reserved, set to 0</td><td>PortCA, PortCB, PortCIndex, used by GC command.
Bit 19=1 – PortCA is used by GC[1] command
…
Bit 21=1 – PortCIndex is used by GC[3] command</td><td>PortCA, PortCB, PortCIndex, used by OC command.
Bit 16=1 – PortCA is used by Output Compare (OC[1] or OC[21]) …
Bit 18=1 – PortCIndex is used by Output Compare (OC[1] or OC[21])</td><td>Outputs 8 – 1, used by GO command.
Bit 8=1 – Output 1 is used by GO[1] command…
Bit 15=1 – Output 8 is used by GO[8]</td><td>Outputs 8 – 1, used by OC command.
Bit 0=1 – Output 1 is used by Output Compare (OC[1] or OC[21]) …
Bit 7=1 – Output 8 is used by Output Compare (OC[1] or OC[21])</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Index</th><th>Description</th><th>Details</th></tr>
<tr><td>111</td><td>Endat3 temperature sensors supported for SENSOR _TEMP_MAX (FID 0x20).</td><td>
The command output is divided to 3 groups, each group is bit-field of 8 bits that represents the supported temperature external sensors per motor phase.
The bit index for each motor phase represents specific temperature sensor type as follows:
Bit 0 – None
Bit 1 – KTY 84-130
Bit 2 – PT-1000
Bit 3 – PT-100
Bit 4 – KTY 83-110
Other – Reserved.
The groups per motor phase are denoted via these bit sequences:
<table>
<tr><th>Bits 31-24</th><th>Bits 23-16</th><th>Bits 15-8</th><th>Bits 7-0</th></tr>
<tr><td>Reserved, set to 0</td><td>Supported sensors for M3 temperature</td><td>Supported sensors for M2 temperature</td><td>Supported sensors for M1 temperature</td></tr>
</table>
</td></tr>
<tr><td>112</td><td>Axis related
Platinum Quartet drive
Output Compare Port C resources.</td><td>
Command is bit field that differently presents Quartet Port C1 – Port C4 resources used by Output Compare (if enabled) and GD commands
<table>
<tr><th>Port C2 used by GD[2] command</th><th>Port C2 used by OC[1] or OC[21]</th><th>Port C1 used by GD[1] command</th><th>Port C1 used by OC[1] or OC[21]</th></tr>
<tr><td>Bit 12: PC2A_TX
Bit 13: PC2B_TX
Bit 14: Reserved
Bit 15: Reserved</td><td>Bit 8: PC2A_TX
Bit 9: PC2B_TX
Bit 10: Reserved
Bit 11: Reserved</td><td>Bit 4: PC1A_TX
Bit 5: PC1B_TX
Bit 6: Reserved
Bit 7: Reserved</td><td>Bit 0: PC1A_TX
Bit 1: PC1B_TX
Bit 2: Reserved
Bit 3: Reserved</td></tr>
<tr><th>Port C4 used by GD[4] command</th><th>Port C4 used by OC[1] or OC[21]</th><th>Port C3 used by GD[3] command</th><th>Port C3 used by OC[1] or OC[21]</th></tr>
<tr><td>Bit 28: PC4A_TX
Bit 29: PC4B_TX
Bit 30: Reserved
Bit 31: Reserved</td><td>Bit 24: PC4A_TX
Bit 25: PC4B_TX
Bit 26: Reserved
Bit 27: Reserved</td><td>Bit 20: PC3A_TX
Bit 21: PC3B_TX
Bit 22: Reserved
Bit 23: Reserved</td><td>Bit 16: PC3A_TX
Bit 17: PC3B_TX
Bit 18: Reserved
Bit 19: Reserved</td></tr>
</table>
</td></tr>
<tr><td>113</td><td>Non axis related
Platinum Quartet drive
Output Compare Port C resources</td><td>
Command is bit field that differently presents Quartet Port C1 – Port C4 resources used by Output Compare (if enabled) and GD commands
<table>
<tr><th>Port C2 used by GD[2] command</th><th>Port C2 used by OC[1] or OC[21]</th><th>Port C1 used by GD[1] command</th><th>Port C1 used by OC[1] or OC[21]</th></tr>
<tr><td>Bit 12: PC2A_TX
Bit 13: PC2B_TX</td><td>Bit 8: PC2A_TX
Bit 9: PC2B_TX</td><td>Bit 4: PC1A_TX
Bit 5: PC1B_TX</td><td>Bit 0: PC1A_TX
Bit 1: PC1B_TX</td></tr>
</table>
</td></tr>
</table>

Index	Description	Details			
		Bit 14: Reserved Bit 15: Reserved	Bit 10: Reserved Bit 11: Reserved	Bit 6: Reserved Bit 7: Reserved	Bit 2: Reserved Bit 3: Reserved
		Port C4 used by GD[4] command	**Port C4 used by OC[1] or OC[21]**	**Port C3 used by GD[3] command**	**Port C3 used by OC[1] or OC[21]**
		Bit 28: PC4A_TX Bit 29: PC4B_TX Bit 30: Reserved Bit 31: Reserved	Bit 24: PC4A_TX Bit 25: PC4B_TX Bit 26: Reserved Bit 27: Reserved	Bit 20: PC3A_TX Bit 21: PC3B_TX Bit 22: Reserved Bit 23: Reserved	Bit 16: PC3A_TX Bit 17: PC3B_TX Bit 18: Reserved Bit 19: Reserved

Details: WS[133]/WS[134] Titanium X1/X2 Ports A, B, C used per axis by OC[1] or OC[21]

WS[n]	Bits 32-24	Bits 23 -20	Bits 19 -16	Bits 15 -12	Bits 11-8	Bits 7-4	Bits 3-0
	Reserved	**X1/X2 Port channels used by OC[21]**			**X1/X2 Port channels used by OC[1]**		
		Port C	**Port B**	**Port A**	**Port C**	**Port B**	**Port A**
WS[133] X1		bit 20=1 X1C_A bit 21=1 X1C_B bit 22=1 X1C_I bit 23 Reserved	bit 16=1 X1B_A bit 17=1 X1B_B bit 18=1 X1B_I bit 19 Reserved	bit 12=1 X1A_A bit 13=1 X1A_B bit 14=1 X1A_I bit 15 Reserved	bit 8=1 X1C_A bit 9=1 X1C_B bit 10=1 X1C_I bit 11 Reserved	bit 4=1 X1B_A bit 5=1 X1B_B bit 6=1 X1B_I bit 7 Reserved	bit 0=1 X1A_A bit 1=1 X1A_B bit 2=1 X1A_I bit 3 Reserved
WS[134] X2		bit 20=1 X2C_A bit 21=1 X2C_B bit 22=1 X2C_I bit 23 Reserved	bit 16=1 X2B_A bit 17=1 X2B_B bit 18=1 X2B_I bit 19 Reserved	bit 12=1 X2A_A bit 13=1 X2A_B bit 14=1 X2A_I bit 15 Reserved	bit 8=1 X2C_A bit 9 =1 X2C_B bit 10=1 X2C_I bit 11 Reserved	bit 4=1 X2B_A bit 5=1 X2B_B bit 6=1 X2B_I bit 7 Reserved	bit 0=1 X2A_A bit 1=1 X2A_B bit 2=1 X2A_I bit 3 Reserved

Details: WS[135] Titanium X1 Ports A, B, C used non-axis by OC or GA

WS[n]	Bits 32-24	Bits 23 -20	Bits 19 -16	Bits 15 -12	Bits 11-8	Bits 7-4	Bits 3-0
	Reserved	X1 Port channels used by GA[]			X1 Port channels used by OC[1] or OC[21]		
		Port C	**Port B**	**Port A**	**Port C**	**Port B**	**Port A**
WS[135] X1		Bit 20=1 – X1C_A Bit 21=1 – X1C_B Bit 22=1 – X1C_I Bit 23 - reserved	Bit 16=1 – X1B_A Bit 17=1 – X1B_B Bit 18=1 – X1B_I Bit 19 - reserved	Bit 12=1 – X1A_A Bit 13=1 – X1A_B Bit 14=1 – X1A_I Bit 15 - reserved	Bit 8=1 – X1C_A Bit 9=1 – X1C_B Bit 10=1 –X1C_I Bit 11 - reserved	Bit 4=1 – X1B_A Bit 5=1 – X1B_B Bit 6=1 –X1B_I Bit 7 - reserved	Bit 0=1 – X1A_A Bit 1=1 – X1A_B Bit 2=1 –X1A_I Bit 3 - reserved

Details: WS[136] Titanium X2 Ports A, B, C used non-axis by OC or GA

WS[n]	Bits 32-24	Bits 23 -20	Bits 19 -16	Bits 15 -12	Bits 11-8	Bits 7-4	Bits 3-0
	Reserved	**X2 Port channels used by GA[]**			**X2 Port channels used by OC[1] or OC[21]**		
		Port C	**Port B**	**Port A**	**Port C**	**Port B**	**Port A**
WS[136] X2		Bit 20=1 – X2C_A Bit 21=1 – X2C_B Bit 22=1 – X2C_I Bit 23 - reserved	Bit 16=1 – X2B_A Bit 17=1 – X2B_B Bit 18=1 – X2B_I Bit 19 - reserved	Bit 12=1 – X2A_A Bit 13=1 – X2A_B Bit 14=1 – X2A_I Bit 15 - reserved	Bit 8=1 – X2C_A Bit 9=1 – X2C_B Bit 10=1 –X2C_I Bit 11 - reserved	Bit 4=1 – X2B_A Bit 5=1 – X2B_B Bit 6=1 –X2B_I Bit 7 - reserved	Bit 0=1 – X2A_A Bit 1=1 – X2A_B Bit 2=1 –X2A_I Bit 3 - reserved

Product Code

The following table describes the number of the product, and the value as configured in **WS[30]**. The product name is retrieved via **VR** command and via **Object 0x1008. Not all products are available.**

Drive # (WS[30] bits 0-7)	Drive Name
3	Harmonica
4	Cello
5	Bassoon
6	Trumpet
7	Tuba
8	Reserved
9	Cornet
10	Whistle
11	Didge
12	Twitter
13	Drum
14	Viola
15	Bee
16	Hornet
17	Hawk
18	Falcon
19	Eagle
20	Reserved
21	Guitar
22	Bell
23	Trombone
24	Panther
25	Reserved
26	Tiger
27	Oboe
28	Clarinet
29	Violin
30	Piccolo
31	Flute

Drive # (WS[30] bits 0-7)	Drive Name
32	Baritone
33	Sitar
34	Mandolin
35	Canary
36	Robin
37	Raven
38	Cimbasso
39	Quartet
40	T-Harmonica
41	String
42	Wind
43	Jaguar
44	Jori
45	Leopard
46	Castanets
47	Lizard
48	Harmonica

Configuration Bits

<table>
<tr><th>Bit number</th><th colspan="2">Function</th></tr>
<tr><td>Bit 0-1</td><td colspan="2">Extended RS connection: 01: Out 3,4 & 10: RS232 Tx\Rx - For internal use</td></tr>
<tr><td>Bit 2</td><td colspan="2">Used for phase loss detection: AC fail</td></tr>
<tr><td>Bit 3</td><td colspan="2">Use for Shunt request activation</td></tr>
<tr><td>Bit 4</td><td colspan="2">Sync PWM: For internal use</td></tr>
<tr><td>Bit 5</td><td colspan="2">Sync PWM gantry: For internal use</td></tr>
<tr><td rowspan="3">Bit 6</td><td colspan="2">ECAT Switches:</td></tr>
<tr><td>0</td><td>No switches</td></tr>
<tr><td>1</td><td>Use ECAT Switches</td></tr>
<tr><td>Bit 7</td><td colspan="2">Sync signal: For internal use</td></tr>
</table>

References

TW[*N*]

WT[] command – Warning Threshold

The **WT[]** command indicates when a warning threshold was violated.

The warning threshold can be used to indicate the host a potential for a possible problem before the fault situation that will disable the servo.

CANopen/CoE

Elmo Alias Object: 0x3608

Attributes

Attribute	Description
Type	Bit-field, Unsigned short, Read only
Source	All
Restrictions	NA
Range	0…65535
Index range	1
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

The thresholds that can be indicated are:

- Voltage below threshold level
- Voltage above threshold level
- Temperature exceeded threshold level
- Analog sensor amplitude is below the minimum or above the maximum

A threshold violation is also indicated via:

- DS-402 Status word (Object **0x6041[0]**): bit 7 - Warning
- Elmo Status register (**SR**): bit 15 – Warning Threshold

The variables that sets the thresholds are described in the following table

Indices

The following table describes the available options for **WT[]**.

Index	Description	Type	Values	Restrictions
1		Unsigned short	Bit field	

Index	Description	Type	Values		Restrictions
	Warning Threshold		**Bit**	**Value**	
			0, 1	Reserved	
			2	Bus Voltage is below the voltage threshold indicated via **XT[1].**	
			3	Bus Voltage is above e threshold set via **XT[2]**	
			4	Analog Sensor amplitude lower than threshold set via **S#[13]** #: number of sensor socket	
			5	Drive Temperature exceeded the threshold set via **TF[2]**.	
			6 - 8	Reserved	
			9	Analog Sensor amplitude higher than threshold set via **S#[14]**.	
			10-15	Reserved	

References

MF[], SW[], SR[], AE[], WE[], TF, XT, S#

XA[] – Extra Axis Parameters

XA[] specifies extra (special) axis related parameters.

CANopen/CoE

Elmo Alias Object: 0x3346

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	See below
Range	See below
Index range	1 to 13
Default	See below
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

Indices

The following table describes the **XA[*N*]** entries.

Index	Description	Default	Range	Units	Comments
1	Maximum speed for dynamic brake	0	0 to 10^{11}	UU/sec	
2	Maximum hot plug in speed	5	0 to 10^{11}	Electrical cycles/second	See Notes.
3	Current exceeded detection time. Fault if current exceeds **MC** by 15% during this time.	100	0 to 9000	µsec	
4	Reserved				
5	Maximal current command step in one TS cycle	20	20 to 200	% of **MC**	
6	Low-pass filter constant for pre-filter of current loop	3000	0 to 20000	Hz	If it equals zero, the filter is bypassed

Index	Description	Default	Range	Units	Comments
7	Field Weakening threshold velocity	100000	0 to 10^{11}	Velocity UU	
8	Field Weakening phase slope	0	0 to 10	Electrical deg /Velocity UU	
9	Electrical velocity low-pass filter	500	0 to 3000	Hz	Electrical velocity (commutation sensor velocity) is used in voltage feed forward and hot plugging algorithms
10					
11	Hot plug in threshold	10^{19}	0 to 10^{19}	Velocity UU	Active if **AB[11]**=1. See Notes.
12	Velocity integral initial value at hot plug in, positive direction	0	**-MC[1]** to **MC[1]**	Ampere	Active if **AB[11]**=1. Defines velocity integral initial value, while rotating in positive direction
13	Velocity integral initial value at hot plug in, negative direction	0	**-MC[1]** to **MC[1]**	Ampere	Active if **AB[11]**=1. Defines velocity integral initial value, while rotating in negative direction

Notes

Dynamic Brake:

XA[1] is the maximum speed that the drive will operate the dynamic brake when the motor is disabled. When this parameter is zero, the dynamic brake function is disabled, otherwise, during motor off (due to command or fault) the drive checks the speed and if below value of **XA[1]**, sets the dynamic brake.

However, if the speed is above value of **XA[1]**, the dynamic brake is released (if it was set).

The speed is taken as the maximum speed between the commutation sensor and main speed (velocity for controller). This protects against the situation when the drive is in current mode and the velocity socket was not set. In dual loop, the main speed is in position units.

Note that in DC brush motors, at Current mode, a situation is possible where no feedback is present resulting in no speed protection for the dynamic brake release. This situation could cause damage to the drive.

Hot Plug In:

Motor cannot be enabled "on the fly" (hot plug in), if its speed is higher than **XA[2].**

XA[11] defines motor reaction during hot plugging process basing on **AB[11]** as follows:

If **AB[11]**=0 , compatibility mode:

- stop motor immediately by setting velocity or current commands to 0 (depending on the servo control mode).

If **AB[11]**=1:

- Profile Velocity mode:
  - continue rotating, if velocity is higher than **XA[11]**;
  - stop motor with deceleration according to the halt option code **0x605D**, if velocity is lower than **XA[11]**;
- Profile Position mode:
  - stop motor with deceleration according to the halt option code **0x605D**;
- Other profile modes:
  - set velocity or current commands to 0.

Setting of K_e (**MP[3]**) is required for correct hot plugging algorithm functionality. It is recommended to do not exceed the default **XA[2]** value**,** if K_e is unknown.

References

TS, AB[7], AB[11], MP[3]

XC – Resume Program

XC[N] Resumes a halted user program

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Unsigned Char, Read/Write
Source	All
Restrictions	None
Range	0 to 255
Default	0
Index Range	1 to7
Unit modes	All
Non-volatile	Yes
Attribute	No

Remarks

While user program is running, the user can halt program, temporarily, using **HP[]** command.

In order to continue running from the point it was halted, **XC[]** command is used.

Index	Description
1	Resume a halted for thread 0 (main).
2	Resume a halted for thread 1.
3	Resume a halted for thread 2.
4	Resume a halted for thread 3.
5	Resume a halted for thread 4.
6	Resume a halted for thread 5
7	Resume a halted for all threads.

Note

XC[] command cannot release program from a breakpoint. For that purpose, use DB##GO[Thread Index] command.

References

XQ, KL[], HP[]

XF[] - Extended Input Filter

XF[] specifies the time period of the extended digital input filter.

CANopen/CoE

Elmo Alias Object: 0x334B

Attributes

Attribute	Description
Type	Unsigned short, Read/Write
Source	All
Restrictions	None.
Range	0…10000
Index range	1,2
Default	**XF[0]**=2; **XF[1]** = 4 (filter constant for AC fail input filter) **XF[2]** = 0 (filter constant extended inputs 2…32)
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

Filter constant is measured in milliseconds.

0 – means no filter.

Non zero value less or equal to 1000 mean that a low-pass filter is applied to digital input with filter constant **XI[]** (TBD)

To be implemented at a later stage.

Indices

Index	Description	Values	Restrictions
1	Filter for input 1 (FAN AC fail)	0…10000	None
2	Filter for inputs 2-32	0…10000	None

References

XO[], **XI[]**

XI[] - Extended Inputs

XI[] reads the extended digital input bits. The command can also sets the logic or PDO event mask for the extended inputs.

Extended Inputs are hardware depended. A drive can have up to 32 extended inputs. Refer to installation guide of the specific drive.

When the hardware is not supported, the command returns 0.

CANopen/CoE

Elmo extended inputs: 0x2202

Elmo Alias Object: 0x334E

Attributes

Attribute	Description
Type	Unsigned long Index 1: Read\Write (write has no effect and does not return error) Index 2: Read\Write
Source	All
Restrictions	None
Range	XI[0]=2; **XI[1]**, **XI[2]** – the range is 0…0xFFFFFFFF
Index range	1 to 2
Default	Index 1 – None Index 2 – 0
Unit modes	All
Non-Volatile	Index 1 – No, Index 2 – Yes
Axis Related	No

Remarks

Extended digital inputs are updated every TS*HS µSec. They are general purpose inputs and are not subjected to any internal function.

The extended inputs can be mapped to a PDO via CANopen and EtherCAT.

Values

Extended digital inputs 17…24 are indicated in **XI[1]16**…**XI[1].23**.

Values for **XI[]** are dependent on whether the "Use ECAT switch" (**WS[80]** bit 6) is set to 0 or 1 whereby **XI[1]**.16…**XI[1]**.23 are set to 0. See table below.

WS[80] bit 6 "Use ECAT switch"	XI[1] Bits 31…24	XI[1] Bits 23…16	XI[1] Bits 15…0
0	Always 0	Extended digital inputs 17…24, where **XI[1]** bit 16 is input 17	Extended digital inputs 1…16, where **XI[1]** bit 0 is input 1
1	Always 0	Always 0	Extended digital inputs 1…16, where **XI[1]** bit 0 is input 1

Indices

Index	Description	Values	Restrictions
1	Reads the extended digital input value (write has no effect and does not return error) **XI[1]** value is dependent on status of drive extended inputs and XI[2]: ReadValue XI[1] = Read Inputs from CPLD xor XI[2]	0...0xFFFFFFFF	See value definitions
2	Read\Write the logic of the input.	0	

References

XO[], **XF[]**

XM[] – Position Modulo

XM[] specifies the counting range for the main feedback, which is [XM[1]…XM[2] - 1].

The main feedback can be retrieved by object 0x6064 or the **PU** command.

CANopen/CoE

XM[1] – 0x607B.1

XM[2] – 0x607B.2Elmo Alias Object: 0x3352

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	• The motor must be off. • Homing must not be active (**HM[1]** = 0)
Range	-2^{52} to $(2^{52} - 1)$
Index range	1, 2
Default	**XM[1]** = 0 **XM[2]** = 0
Unit modes	All
Non-Volatile	Yes
Axis Related	Yes

Remarks

A profiler block can be used for generating:

- **Non-modulo motion**
  The motion is SW limited to within a finite position range (Object 0x607D or **VH[3], VL[3]**). The non-modulo motion is enabled, if one of the following conditions holds:
  - **XM[2]** = **XM[1]** = 0;
  - **XM[1]** <= **VL[3]** and **XM[2]** >= **VH[3]**;
  - If **XM[2] = XM[1] = VH[3] = VL[3] = 0**, neither the software position limit nor the position range limit affects the motion.

- Modulo motion
  The position counts in a position range limit (Object 0x607B|tag=Object_0x607B_Position_range_limit or XM[1], XM[2]). The modulo motion is enabled, if

  XM[2] > **XM[1]** and **XM[1]** > **VL[3]** and **XM[2]** < **VH[3]**.

In the modulo mode, the feedback (Object 0x6064 or **PU**) is always counted cyclically. This means that after the position is counted to its maximum value, the next position count will reset the position counter back to its minimum value.

The speed reading is not affected by the position jump.

Example

If **XM[1]** = -5 and **XM[2]** = 5, the main position is counted in a cycle with a length equal to **XM[2] - XM[1]** = 10.

The main position will always be in the range [-5…4]. If the main feedback rotates in the positive direction, the main position count will proceed from 0, 1, 2, 3, 4 to -5, -4, -3, -2, -1, 0, 1 . . . and so on.

Optional modulo movements

The following modulo modes are optional. In all modes positioning can be relative or absolute.

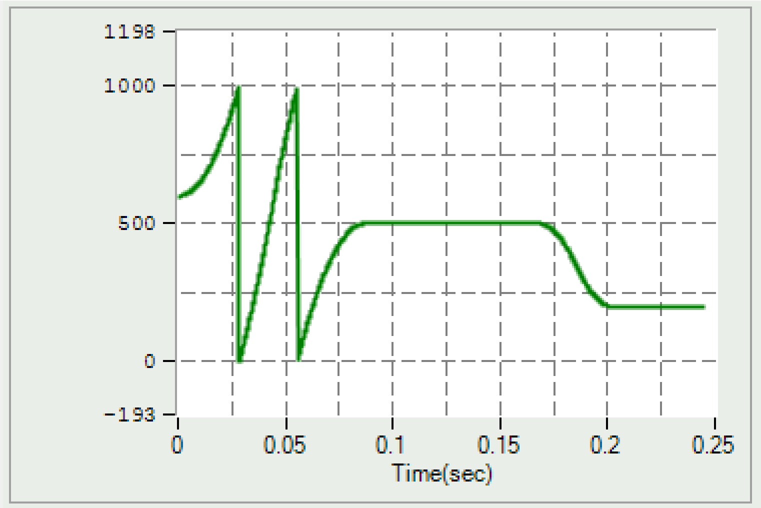
Using object 0x60F2 bits 7:6 (or **PO[1]** bits 7:6):

Mode	Bits 7:6 Value	Remarks
Normal positioning	0x0 (default)	Normal rotary positioning is similar to linear axis positioning. If the position range limits (Object 0x607B or **XM[1]**, **XM[2]**) are reached or exceeded, the input value wraps automatically to the other end of the range. Movement greater than a modulo is possible only with this bit combination.
Negative movement	0x1	Positioning only in the negative direction. If the target position is higher than the actual position, the axis moves over the minimum position range limit (**XM[1]** or Object 0x607B.1) to the target position.
Positive movement	0x2	Positioning only in the positive direction. If the target position is lower than the actual position, the axis moves over the maximum position range limit (**XM[2]** or Object 0x607B.2) to the target position.
Positioning with shortest path	0x3	Positioning with the shortest path to the target position. **Note:** If the difference between actual value and target position in a 360° system is 180°, the axis moves in positive direction.

Example of normal modulo positioning

The profile generator does not remember the modulo rolling that the position demand value passed. Every new movement is based on the location of the current demand value within the modulo range.

In the figure below, **XM[1]** = 0, **XM[2]** = 1000, the initial position is **PX**=600, and two absolute motions are applied: **PA** = 2500 and, then, **PA** = 200. In the first motion, the PTP trajectory travels through 600…999, 0, 1…999, 0, 1…499, 500 over a total distance of 1900 counts. In the second motion, it travels through 500, 499…201, 200 over a total distance of (-300) counts.



Example of positioning with shortest path:
If **XM[1]** = -512, **XM[2]** = 512, the initial position is **PX** = -500, and the target absolute position is **PA** = 500, the PTP trajectory will travel through -500…-512, 511…500 over a total distance of 23 counts.

Default modulo behavior versus Elmo legacy modulo behavior

The drive default modulo, as well as the DS-402 default modulo, is the *normal positioning* mode.

In order to maintain the Elmo legacy mode, the modulo mode needs to be set to *positioning with shortest path* by a **PO** command.

In the case where Object 0x607A is set using the control word (Object 0x6040), the setting of object 0x60F2 will be used.

Note that by default the drive is not in a modulo state, because the Software Position limit (Object 0x607D or **VH[3]**, **VL[3]**) is lower than the Range limit (Object 0x607B or **XM[1]**, **XM[2]**).

Indices

The following table describes the **XM[]** entries.

Index	Description	Type	Values
1	Minimum position range limit	User-defined	-2^{52} to $(2^{52} - 1)$
2	Maximum position range limit	User-defined	-2^{52} to $(2^{52} - 1)$

Notes

- If **XM[1]** or **XM[2]** is set so that the main feedback (**PU**) is outside of the range [**XM[1]…XM[2]**], **PU** should be set to the range by taking the modulo:

PU = (**PU** - **XM[1]**) mod (**XM[2]** - **XM[1]**) + XM[1], if **PU** > **XM[2]**, or
PU = -(-**PU** + **XM[2]**) mod (**XM[2]** - **XM[1]**) + **XM[2]**, if **PU** < **XM[1]**

This is done at power-up and at motor ON (**MO** = 1).

- A new **XM[1]**/**XM[2]** setting is activated after the setting of **XM[2]**, at power-up or upon issuing the **MO**=1 command.
- If **XM[2]** = **XM[1]** = 0, the position modulo functionality is disabled. **VH[3]** and **VL[3]** are used for positioning limiting.
- If the **XM[*N*]** value is selected low and the main speed is too high, more than one full revolution of the main counter may occur within a single sampling time. This will cause the main position counter to behave unpredictably. This happens when the modulo cycle completes in less than **HS*TS** µsec. The drive will not indicate this as a failure, and it is up to the user to remedy the situation.
- The **XM[*N*]** values should be given in user-defined position units specified by the **FC** command.

References

VH[]/VL[], **PO**, **FC**

XO[] - Extended Outputs

XO[] sets and reads the extended digital output bits. The command can also sets the logic or output bit mask.

Extended outputs are hardware depended. A drive can have up to 32 extended outputs. Refer to installation guide of the specific drive.

When the hardware is not supported, the command returns 0.

CANopen/CoE

Elmo extended outputs: 0x22A1

Elmo Alias Object: 0x3354

Attributes

Attribute	Description
Type	Unsigned Long Bit field, Read\Write
Source	All
Restrictions	None
Range	Digital outputs 0…0xFFFFFFFF
Index range	1 to 3
Default	XO[0]= 3; XO[1] = 0 **XO[2]** = 0, means not invert input values (active high) **XO[3]** = 0xFFFFFFFF , means all enabled
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

The extended digital outputs are updated every TS µSec.

The extended inputs can be mapped to a PDO via CANopen and EtherCAT.

Values

Extended output 1 is related to **XO[]** bit 0, Extended output 32 is related to **XO[]** bit 31.

Indices

Index	Description	Values	Restrictions
1	Read/Write the extended digital output logic level:	0..0xFFFFFFFF	none

Index	Description	Values	Restrictions
	0: non-active 1: active Extended Outputs = (**XO[1]** XOR **XO[2]**) & **XO[3]**)		
2	Logic level of the output: 0: do not invert **XO[1]** values 1: invert **XO[1]** values	0…0xFFFFFFFF	None
3	Extended output mask: 0: Output is disabled 1: Output is enabled	0…0xFFFFFFFF	none

References

XI[], **XF[]**, **TS**

XP[] – Extra Non-Axis Parameters

XP[] specifies extra (special) non-axis related parameters.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Parameter, Integer
Type	Short, Read/Write
Source	All
Restrictions	See below
Range	See below
Index range	1 to 5
Default	See below
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

Indices

The following table describes the **XP[*N*]** entries.

Index	Description	Default	Range	Units	Comments
1	Overvoltage threshold	0	0 to **BV**	Volt	Can only be reduced from the default value (**BV**)
2	Defines the PWM frequency as a factor of the controller sampling time: **TS**/TPWM = **TS***fpwm = **XP[2]**/2 The current ripple frequency at load is equal to **XP[2]**/**TS**.	2	0 to 6		Motor enable is not allowed after **XP[2]** change until drive reboot
3	Control loops execution order: 0 – Set according to drive configuration: If number of axes (**WS[1]**) ≤2: Sensors→Position→Velocity→Current→PWM	0	0 to 2		Motor must be disabled

Index	Description		Default	Range	Units	Comments
		If number of axes (**WS[1]**) >2: Current→PWM→Sensors→Position →Velocity				
	1	Force Position/Velocity control first: Sensors→Position→Velocity→Curr ent→PWM				
	2	Force current control first: Current→PWM→Sensors→Position →Velocity				
4	Filter constant of bus voltage measurements		1500	100 to 3000	Hz	
5	Undervoltage threshold		WI[38]	0 to **WI[38]**	Volt	

Notes

- The current loop must be tuned after a modification of **XP[2]**.
- When **XP[2]** is used to multiply the PWM frequency from the default, the current saturation (**CL[1]** and **PL[1]**) might be reduced. The actual values of **PL[1]** and **CL[1]** are reported in **WS[33]** and **WS[34]**, respectively.

References

TS,

XQ –Run User Program

XQ## Executes the user program from a specified label or runs a specified function.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	String, Read/Write
Source	All
Restrictions	None
Range	None
Default	0
Index Range	1
Unit modes	All
Non-volatile	No
Attribute	No

Remarks

XQ## executes a valid user program.

This command shall be typically sent after sending a successful **CC** command.

The general format is:

XQ##[function name]

Examples

- XQ## runs from the start of the user program code.
- XQ##MyFunction(a,b,c) runs the function MyFunction () with a,b & c as arguments to the function.
- XQ##LABEL runs from ##LABEL.

XQ command clears the error status of the program, along with run-time error flags.

It does not reset program variables and does not clear the interrupt mask.

Notes

- **XQ** must include ##, w/o it an error shall be returned.
- **XQ##** , beside acknowledge, does not return a value.
- **XQ##** is also enable all the multi threads that define in the user program.
- If the program is running – **XQ** can't be sent.

References

CC, XC[], PS[], KL[], HP[], CP[]

XT[] – Thresholds parameters

XT[] sets levels for a warning threshold for the bus voltage.

CANopen/CoE

Elmo Alias Object: **0x3359**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	USB, TCP, EoE, RS232
Restrictions	None
Range	See the table below
Index range	1,2
Default	See the table below
Unit modes	All
Non-volatile	Yes
Axis Related	No

Remarks

The threshold violation is indicated via **WT[].** See table below.

Threshold violation is also indicated as warning in Elmo Status Register of axis 1 (**SR**[1] bit 15) and in the Statusword (Object **0x6041** bit 7).

Indices

The following table describes the **XT[*N*]** entries.

Index	Description	Default	Range
0	Reserved		
1	Defines the low bus voltage in millivolts. If the bus voltage power drops below this value bit 2 of WT command is set. The bus voltage can be retrieved via **AN[6].**	0	[0… **BV***1000]
2	Defines the highest bus voltage in millivolts. If the bus voltage power increases this value bit 3 of WT command is set. The bus voltage can be retrieved via **AN[6].**	100000	[0… **BV***2000]

References

WT, BV, AN[6]

YG[] – Gantry parameters

YG[] specifies the Gantry related parameters.

CANopen/CoE

Alias Object: 0x3370

Attributes

Attribute	Description
Type	Double, Read/Write
Source	All
Restrictions	**YG[3]** > **YG[2]**
Range	According to array index
Index range	N=1, 2, 3
Default	According to array index
Unit modes	**UM** = 2, 5
Non-Volatile	Yes
Axis Related	Yes

Remarks

YG[1]…**YG[3]** define offset and range of the Gantry bridge upper axis (Y) position. They are used for current compensation $I^*_{Y\ FF}$ calculation:

$$I^*_{Y\ FF} = \frac{(P_Y - YG[1])}{YG[3] - YG[2]} * \frac{d^2 P^*_X}{dt^2} * K_{YX}$$

Where:

- P_Y - upper axis position, the axis is defined via **SA[15]**;
- $\frac{d^2 P^*_X}{dt^2}$ – Gantry profile acceleration command;
- K_{YX} – current compensation (feed forward) factor **FF[11]**.

Indices

The following table describes the **YG[]** entries.

Index	Description	Default	Values	Remarks
1	Gantry bridge upper axis (Y) position offset	0	-2^{52} to $(2^{52} - 1)$	
2	Minimum Gantry bridge upper axis (Y) position	0	-2^{52} to $(2^{52} - 1)$	**YG[3]** > **YG[2]**

Index	Description	Default	Values	Remarks
3	Maximum Gantry bridge upper axis (Y) position	0	-2^{52} to $(2^{52} - 1)$	**YG[3]** > **YG[2]**

Note: No error will be informed for the YG[] restriction described above.

References

SA[*N*], FF[N]

YS – Ethernet Parameters Status

YS retrieves the Ethernet stack actual parameters.

CANopen/CoE

Object 0x337C

Attributes

Attribute	Description
Type	String, read-only
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	None
Index Range	1-8
Default	NA
Unit modes	All
Non-Volatile	No
Axis Related	No

Remarks

The **YS** command reports the Ethernet stack actual parameters in string format.

The string includes:

- MAC Address
- IP Address
- Subnet Mask
- Gateway

The format is as follows: numbers separated by ':'.

Example for IP address:

- In case hx=0: 255:255:255:255
- In case hx=1: FF:FF:FF:FF

Example for MAC address:

- In case hx=0: 10:10:10:10:10:10
- In case hx=1: 0A:0A:0A:0A:0A:0A

Indices

The following table describes the **YS[*N*]** entries.

Index	Description
1	Ethernet channel MAC address
2	Ethernet IP address
3	Ethernet Net-Mask address
4	Ethernet Gateway address
5	EtherCAT EoE channel MAC address
6	EtherCAT EoE IP address
7	EtherCAT EoE Net-Mask address
8	EtherCAT EoE Gateway address

Notes

All indices that are not listed include the following logic:

A Write operation will return an error, i.e., 19?; - "Command syntax error"

MAC address is set by the factory and is not writable. The Ethernet IP address ,Net mask & default Gateway address is set via the PP[24], PP[25] & PP[26] respectively.

The EoE addresses are set by the EtherCAT host.

References

PP[]

ZF[] – Series of Miscellaneous Commands Parameters

ZF[] is used to store internal information by the EAS II application.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Parameter, Integer
Type	double
Source	All
Restrictions	None
Range	See the table below
Index range	According to the table below
Default	0
Unit modes	All
Non-Volatile	No
Axis Related	Yes

Remarks

Indices

The following table describes the **ZF[*N*]** entries.

Index	Description
1	Motor Continuous Current
2	Motor Peak Current
3	Maximal Motor Speed
4	Motor Magnetic Pitch
5	Commutation Sensor Resolution
6	Velocity Sensor Resolution
7	Position Sensor Resolution
8	Transmission Parameter
9	Sine sweep Initial Controller – Velocity Gain (**KP[2]**)
10	Sine sweep Initial Controller – Velocity Gain (**KP[2]**)

Index	Description
11	Sine sweep Initial Controller – Position Gain (**KP[3]**)
12	Gantry Sine sweep Initial Controller – Velocity Gain (**KP[4]**)
13	Gantry Sine sweep Initial Controller – Velocity Integral (**KI[4]**)
14	Gantry Sine sweep Initial Controller – Position Gain (**KP[5]**)
15	Sine sweep low pass filter (**KV[1]**,**KV[2]**=0.6, **KV[5]**=1 if parameter not zero)
16	Gantry Sine sweep low pass filter (**KV[81]**,**KV[82]**=0.6, **KV[85]**=1 if parameter not zero)
17	Gear ratio numerator
18	Gear ratio denominator
19	Position display units
20	Velocity display units
21	Axis and control configuration
22	Axis identity
23	Direct reference injection point
24	Electro-mechanical configuration
25	Feedback (loop) configuration
26	Loop feedback configuration
27	Mode of operation
28	Configuration check-boxes
29	Tuner page checked
30	Application tools page checked
31	Position display unit factor
32	Velocity display unit factor
33	DS-402 Position units
34	Velocity display unit factor
35	DS-402 Acceleration units
36	Bit 0 – Not in use Bit 1 – Allow other socket for control Bits 2-5 – Safety motor socket Bits 6-9 – Safety load Socket

Index	Description
	Bits 10-13 – Safety additional socket Bit 14 – Use safe I/O
37	Safety Sensor Resolution – primary
38	Safety Sensor Resolution – secondary
39	Capture number of events initial
40	Counts to increment – position factor
41	Counts to increment – velocity factor
	Counts to increment – acceleration factor
43 - 50	Reserved

References

ZX[N] – Series of Miscellaneous Command Extras

ZX[*N*] Is used as an identification table.

CANopen/CoE

Emo Alias Object: Not Available

Attributes

Attribute	Description
Type	Short, Read/Write
Source	All
Restriction	None
Range	
Index range	2048
Default	0
Unit modes	All
Non-Volatile	Yes
Axis Related	No

Remarks

Manually set the PWM of the Power Stage allowing reduction of the Tuning time when torque open loop tests are required.

This command must be used with extreme care.

Each bit controls a single FET which can be set to ON or OFF. The following describes the bit and the relevant phase that it controls:

Bit	Fet \ Phase
0	A - Lower
1	B – Lower
2	C – Lower
3	A – Upper
4	B – Upper
5	C - Upper

TW[31]=1,5: Set PWM command:

Brushless 3 phases: phase A and B.

Brushless 2 phases: phase A and C

DC Brush and voice coil: Phase B

For **TW[31]**=5 only use one value

TW[31]=6: Use with a factor of **TW[74]** shifted 16 bits to the right and with offset

TW[66]: Fill **ZX** array with chirp for current

TW[67]: Fill **ZX** array with chirp for velocity

References

TW[N]

Go Safer, Smarter, Smaller, Simpler.
Go Platinum.

Platinum by Elmo

www.elmomc.com