

Inspiring Motion

Since 1988

Command Reference for Gold Line Drives

November 2016 (Ver. 1.505)

www.elmomc.com



Notice

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 the Gold Line technology.
- 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.
- Information in this document is subject to change without notice.



Elmo Motion Control and the Elmo Motion Control logo are registered trademarks of Elmo Motion Control Ltd.



EtherCAT Conformance Tested. EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.



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-G-CR (Ver. 1.505)

Copyright © 2016

Elmo Motion Control Ltd.

All rights reserved.

Revision History

Version	Date	Details
Ver. 1.0		Initial version
Ver. 1.1		Following commands added: AD, AR, EA, GO, GP, GS, GV, GW, PC, SE, SO
Ver. 1.2	Mar 2012	Following commands updated or added: AA, AC, AG, AS, BP, CA, CD, CL, CP, DC, DF, DL, EA, EE[], GI[], GO, IB, IF, IP, JV, KR, LD, MR, MS, OB[N], OC[N], OL, PB, RP, RR, RS, SV, TW, VH_VL, VP, WS, XQ
Ver. 1.3	May 2012	Following commands added: JP, NF, PO, PU Following commands updated: AA, BG, CD, DC, FC, FF, FX, GI, IL, IP, JV, KI, KP, MF, MO/SO, OV, PA, PL, PR, PX, SC, SD, SP, VE, VH[]/VL[], VX, XA, XM
Ver. 1.401	Oct 2012	Following commands added: RM, VU, US, HT Following commands updated: BS, EE, FC, KV, PU, PX, VX, XM[] Following commands updated: AA[], EE[], GO, HF/HM, IB[], IL[], IP, MF, OC, WS, CA, EC, EE, FF, XA, XP
Ver.1.402	Oct 2012	Correction to GO
Ver. 1.403	Nov 2012	Corrections to EE, CA, EA
Ver.1.404	Dec 2012	Addition of ECAM commands DV[], EI, EM, ET, IL, PY, RM, YM Addition to command OV Corrections to BS, EE, EM, IL, KV, MF, RP, XM, YM
Ver. 1.405	Jan 2013	Corrections to EE, GS, KV, NF, NT, TW
Ver. 1.406	Feb 2013	Correction to NF
Ver. 1.407	Mar 2013	Correction to RR
Ver. 1.408	Mar 2013	Correction to PL, NT
Ver. 1.409	Apr 2013	Correction to AA
Ver. 1.410	May 2013	Correction to GS[6]
Ver. 1.411	Jun 2013	Correction to OV[56]
Ver. 1.412	Jul 2013	Correction to XA, OV[], SR, TW . Addition of VS .
Ver. 1.413	Jul 2013	Addition of Sanyo Denki, Nikon, Panasonic, and Tamagawa Status Flag Function Descriptions
Ver. 1.414	Aug 2013	Correction to CA, EE, TW
Ver. 1.415	Oct 2013	Correction to GS, MF, OV

Ver. 1.416	Oct 2013	Corrections to CA, OB, OL, OP, TW, AC, ER, BG, DC, DV, FF, GS, JP, MS, PA, PR, SC, SD, SP, SR, ST, WS, XA
Ver. 1.417	Nov 2013	Correction to EE[6], WS[30] entries
Ver. 1.418	Dec 2013	Updates for MR[], AA[], FS, MF, VH[]/VL[] New Command: HL[]/LL[]
Ver. 1.419	Jan 2014	Addition of Nikon to Sanyo Denki parameters and applications Update to VH[]/VL[] command
Ver. 1.420	Feb 2014	Update to GO
Ver. 1.421	Mar 2014	Correction to MS[N] Velocity modes
Ver. 1.422	July 2014	Corrections to CA, EE, NF, SC, WS, XA
Ver. 1.423	Oct 2014	Corrections to CA, DV, EA, ER, FF, FP, HT, KG, KI, KP, MF, NF, TS, UM, WI
Ver. 1.424	Nov 2014	Corrections to VH, WS, XA, XP, CA, EE, SD, EC
Ver. 1.425	Jan 2015	General changes of Integer functions to Long. Corrections to CA, EE, SR
Ver. 1.426	Feb 2015	Corrections to HT, JP, JV, MF
Ver. 1.427	Mar 2015	Corrections to CA
Ver. 1.428	May 2015	Corrections to OC
Ver. 1.429	June 2015	Correction to CA, GO, EA
Ver. 1.430	July 2015	Correction to EA Index 17
Ver. 1.431	July 2015	Correction to EA, HM, PC, EM
Ver. 1.432	Sep 2015	Correction to OF, AC, DC, GO, IL, MF, OC, MF, SD, JV, JP, XP "Constants" – at the beginning of the document
Ver. 1.433	Nov 2015	Updates and corrections to CA, EE, PC, US (new), XA, VH, EM, GO
Ver. 1.500	Nov 2015	Updates and new commands: EA, SN, XI, XO, XT, MS, SR
Ver. 1.501	Feb 2016	Updates and new commands: TW, CA, IL, OC
Ver. 1.502	Mar 2016	Updates: PP[14] = 8 removed (out of range), CA, XT,
Ver. 1.503	May 2016	Updates to CA, FF,
Ver. 1.504	Jul 2016	Updates to DD, AN, NF
Ver. 1.505	Nov 2016	Updates to AS, EC Error codes, KV, MS, TW, WI, WS, XP

Description of Attributes.....	8
Constants.....	9
Status Flag Functions and Terms.....	10
AA[N] – Communication Server Commands	17
AC[N] – Set Acceleration	23
AD[] – Set Analog Dead Band	25
AG[] – Set Analog Gain	26
AN[] – Get Analog Input	28
AR[] – Set Analog Units.....	30
AS[N] – Analog Input Offset	32
BG[N] – Begin Motion	33
BH – Get Recorder Signal	36
BP[] – Brake Parameters.....	38
BS[] – Bring Recorded Sample	41
BT – Begin Motion On Time (Reserved)	43
BV – Bus Voltage.....	44
CA[N] – Commutation Array.....	45
CC – Compilation Checksum.....	67
CD – CPU Dump	68
CL – Current Limit Parameters	70
CP – Clear Program.....	72
CS – Force Commutation Angle.....	73
CU – CPU Usage	75
CW – Control Word	76
DC – Set Deceleration.....	77
DD – CAN controller status.....	78
DL – (Binary) Download.....	82
DV[] – Desired Value	84
EA[N] – Feedback Emulation Parameters	86
EC – Error Code.....	92
EE[N] – Extended Error.....	110
EI – Initialize External Reference Generator	121
EM[N] – ECAM / Follower Parameters.....	122
EO – Echo Off.....	126
ER[] – Maximum Tracking Error	127
ET[N] – ECAM table	129
FC[] – Scaling Factors.....	130
FF[] – Feed Forward.....	133
FP[] – Feedback Position	136
FS – PTP Final Speed	138
FT[] – Float Trigger	139
FV[] – Feedback Velocity	140
GI[] – Capture Input MUX Selection.....	141
GO[] – Digital Output Source.....	144
GP[N] – Error Mapping Correction Table Editing	148
GS[] – Gain Scheduling	149
GV[N] – Output Compare Editing Table	157
GW[N] – Output Compare Editing Table.....	158
GX[] – Capture Array Value from HM.....	159
GY[N] – Capture Array Value from HF.....	160
HL[] /LL[] – High/Low Feedback Limit	161

HM[N]/HF[N] – Main/Aux Homing.....	163
HP – Halt Program	169
HT[] – Open Loop Torque	170
HX – Hexadecimal Mode	172
IA[] – Index Analog Sensor	173
IB[] – Digital Input Bits.....	175
ID, IQ – Active/Reactive Current	177
IF[] – Digital Input Filter.....	178
IL[] – Digital Input Logic	180
IP – Input Port.....	187
JP[N] – Jog Velocity in Position mode	189
JV – Jog Velocity	191
KG[] – Gain Scheduled Controller.....	192
KI[], KP[] – PI Controllers	194
KL – Kill User Program	196
KR – Kill Motion Repetitive.....	197
KV[] – High-Order Controller Filter Parameters	199
LC – Current Limit Flag.....	202
LD – Load Data.....	203
LP[] – Load Program Info.....	205
MC – Maximum Current.....	207
MF – Drive Fault	208
MI – Mask Interrupts.....	214
MO/SO – Motor On, Servo On	216
MP[] – Motion Parameters (Reserved)	219
MR[N] – Motion Repetitive	220
MS[N] – Motion Status	222
NF[] – Non-Linear Float	226
NT – Non-Linear Table	228
OB[N] – Output Bits.....	229
OC[] – Output Compare.....	232
OF[] – CAN Objects.....	238
OL[] – Output Logic.....	241
OP – Output Port	245
OV[] – Set CANopen Objects	247
PA[N] – Position Absolute	252
PB – PAL Burn	254
PC[N] – Error Mapping	256
PE – Position Error	259
PL[N] – Peak Limit.....	260
PO – Positioning Options.....	262
PP[N] – Protocol	264
PR[N] – Position Relative	266
PS – Get Program Status.....	268
PT[] – Position Table (Reserved).....	269
PU – Main Position in User-Defined Units.....	270
PV – Position Velocity Time setting (Reserved).....	271
PX – Main Position in Counts.....	272
PY – Auxiliary Position in Counts	273
RC – Recorder Variables	274
RG – Recorder GAP	275

RL – Recorder Length	276
RM – Reference Mode.....	278
RP[] – Recorder Parameters	280
RR – Activate Recorder / Recorder Status.....	284
RS – Soft Reset.....	286
RV[N] – Recorder Variables	287
SC[N] – Stepper Commutation	288
SD – Stop Deceleration.....	291
SE[N] – Sine Excitation.....	292
SF – Smooth Factor.....	294
SN[N] – Serial Number Identity	295
SO – Servo Enabled.....	297
SP[N] – PTP Profiler Speed	298
SR – Status Register	299
ST[N] – Stop Profiler	305
SV – Save Parameters.....	306
SW – Status Word.....	307
TC – Torque Command.....	308
TI[] – Temperature Information	309
TM – Internal Time	311
TR[] – Target Radius	312
TS – Sampling Time.....	314
TW – Wizard Internal Identification	316
UF[N] – Float User Interface.....	318
UI[N] – Integer User Interface	319
UM – Unit Mode.....	320
US[] – User Saturation Parameters	322
VB – Software Boot Version	324
VE – Velocity Error.....	325
VH[]/VL[] – High/Low Reference Limit	326
VO – Software OTP version	328
VP – PAL Version.....	329
VR – Software Firmware Version	330
VS – STO Software Module Version	331
VU – Main Feedback Velocity in User-Defined Units	332
VX – Main Feedback Velocity in Counts per Second	333
WI[] – Wizard Integer Parameters	334
WS – Miscellaneous Reports	336
XA[] – Extra Parameters	342
XC – Resume Program	344
XI[] – Extended Digital Input Bits.....	345
XM[] – Position Modulo	347
XO[] – Extended Digital Output Bits	351
XP[] – Extra General Parameters	353
XQ – Execute User Program	355
XT[] – Threshold Parameters setting	357
YM[] – External Reference Modulo	359



Description of Attributes

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 <i>Mapping</i> means that the object can be mapped to a process data object (PDO). <i>User Program</i> means that the variable can be manipulated from an Elmo user program. <i>FoE</i> 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: <i>Not User Program</i> means that the command cannot be used in a User Program. <i>Motor Must Be On</i> 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 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.

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

Constants

MAX_POSITION_RANGE	2,147,483,647
MIN_POSITION_RANGE	-2,147,483,648
MAX_VELOCITY	2147483647
MAX_ACCELERATION	2147483647
MAX_CURRENT	Value of MC



Status Flag Functions and Terms

1.1.1. 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, (1) 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 (1) and (2), if (difference between (1) 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, (1) 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: (1) 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 $95^{\circ}\text{C} \pm 4^{\circ}\text{C}$ or more, an alarm is turned on.	During normal operation	Non-latch	Automatically turned off



1.1.2. Panasonic Status Flag Function Descriptions

Name	Function	Battery Error output when power SW is turned ON (refer to NOTE below)		Remedy
		Connected	Unconnected	
Over Speed OS (Latched)	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 SW 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)	←	Undefined	Reset error.
Full Absolute-Status FS (Non- latched)	In case the main Power SW 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.	←	←	Set Encoder shaft revolution speed less than 100r/min and wait until the error is automatically reset.
Count Error CE	In case one revolution data is shifted due to operation error or failure when the main Power SW is ON, the logic "1" will be output by the following I, II OR.			Immediate stop of servo system.
	I (Non-latch) In case Encoder shaft is revolving at more than 100 r/min, error detection will be conducted at every 45 ⁰ mechanical angle. When shift of one revolution data is reduced to the mechanical angle ±22.5 ⁰ (TYP.), the error will be automatically reset.	← ←	← ←	Error will be automatically reset. Power SW OFF/ON
	II (Latch) 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 mechanical angle ± 0.7 ⁰ (TYP.), the logic "1" will be output.	←	←	Reset error. Power SW OFF/ON



Name	Function	Battery Error output when power SW is turned ON (refer to NOTE below)		Remedy
		Connected	Unconnected	
Multiple Revolution Error ME (Latched)	In case the abnormalities in multiple revolution data occurs when the main Power SW is turned ON, the logic "1" will be output. (This does not operate when the main Power SW 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 SW is turned OFF, external output will be enabled when the main Power SW 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 SW 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 SW is turned OFF, the logic "1" will be generated and external transmission will be enabled when the main Power SW 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/s².

Angular acceleration	Over Speed Error
0 - 4000 rad/s ²	"0"
4000 - 28000 rad/s ² : 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/s² (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/s² is recommended.



1.1.3. Tamagawa Status Flag Function Descriptions

Name	Function		Battery Error output depending on connecting battery during main power-on		Remedy
			Connected	Unconnected	
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, because 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.
Counter Overflow OF (Latched)	Logic "1" 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 0- 65,535. When a Battery error occurs, Counter overflow is normally operated by resetting the multi-tum data.		←	Undefined	Reset error.



Name	Function	Battery Error output depending on connecting battery during main power-on		Remedy
		Connected	Unconnected	
Battery Alarm BA (Non- Latched)	In case external battery power supply voltage becomes less than $3.1 \pm 0.1V$ (Refer to 5.4 Common Electric Specifications), 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 $2.5 \pm 0.2 V$ or less (Refer to Paragraph 5.4 "Common Electrical Specification"). and it can be transmitted after main power-on. When this flag occurs immediately after main power-on, the multi-turn 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.



AA[N] – Communication Server Commands

The **AA[]** command addresses the communication server (ATMEL).

CANopen/CoE

Attributes

Attribute	Description
Type	General use, Long
Source	All, except RS232 and the user program from page 4
Restrictions	According to the entry description
Range	According to the entry description
Index range	1 to 99
Unit modes	All
Non-volatile	According to entry description
Attribute	None

Remarks

Some of the **AA[N]** entries (indices) are for internal use. These entries are basically protected, and modifying these entries may harm the functionality of the drive.

The **AA[]** command is handled in the communication server and should not be addressed to the digital signal processor (DSP), which is the main processor. Thus, limited interpretation is available. For example, unlike other commands, the **AA[]** command is case-sensitive and responds only to its upper-case name. If the command generates an error, the reply is 19?

The number 19 is the error code for a command syntax error.

The **AA[]** command is mainly for internal use and includes very limited interpretation abilities.

When an error is returned (19?;) the EC should not be influenced.



Indices

The following table describes the **AA[N]** entries. Index values which are not indicated are for internal use only.

Index	Description	Type	Possible Values		Restrictions
1	Drive FW status	Bit field	Bit	Description	Read-only
			0	ATMEL in boot	
			1	ATMEL in FW	
			2	DSP in boot	
			3	DSP in FW	
2	MAC address	String	Depends on the drive hardware		Read-only
4	Ethernet/EtherCAT selection. Set by the factory.	String	0	Drive supports Ethernet only	Read-only
			1	Drive supports EtherCAT and Ethernet	
5	ATMEL version string	String	(for example) <u>In the boot state:</u> ATMEL Boot 1.1.1.0 ([compilation date]) <u>In the FW state:</u> ComServer 1.1.1.0 ([compilation date])		Read-only
6	Drive serial number	String	Same value as the SN[4] command		Read-only



8	ATMEL general status bits	Bit field	Bit	Value	Description
			0	0	EtherCAT hardware init OK
				1	EtherCAT hardware failed to initiate.
			1	0	EtherCAT SW init OK
				1	EtherCAT SW failed to initiate.
			2	0	ATMEL flash memory size is 512.
				1	ATMEL flash memory size is 256.
			3	0	Communication between the DSP & ATMEL is synchronized.
				1	Communication between the DSP & ATMEL is not synchronized. Messages are not transferred between the CPUs.
			4	0	Drive is configured for Ethernet communication.
				1	Drive is configured to EtherCAT communication.
			5	0	EtherCAT chip is ready for use.
				1	EtherCAT chip failed to initiate.
			6 to 31		Reserved
10	IP address set	String	IP address set by the host. The IP address has the following format: xx.xx.xx.xx The actually used IP address can be retrieved by calling the command with Index 20. If DHCP or EoE is used, the actual IP address might differ from this address. The Default is 192.168.1.49		Read/Write Saved to flash memory



11	Subnet mask set	String	Net mask set by the host. The Net mask has the following format: xx.xx.xx.xx The actually used Net mask address can be retrieved by calling the command with Index 21. The Default is 255.255.255.0		Read/Write Saved to flash memory
12	Gateway set Gateway set by the host	String	The gateway has the following format: xx.xx.xx.xx The Default is 192.168.1.1		Read/Write Saved to flash memory The actual gateway can be retrieved by calling the command with Index 22.
13	EtherCAT/Ethernet switching	String	42330	Ethernet	Read/Write Saved to flash memory Note: The change takes effect after power-up. All values, except 1 and 42330, are reserved, but will be treated as 1.
			1	EtherCAT (only if AA[4] supports EtherCAT)	
14	DHCP enable/disable If Ethernet is selected, this command enables or disables the use of DHCP to configure the Ethernet parameters.	String	0	DHCP disabled (default)	Read/Write Saved to flash memory
			1	DHCP enabled	
			Note: The change takes effect after power-up.		



15	ATMEL boot version	String	Format: AtmelBoot 1.1.1.0 ([compilation date][CPU flash memory size])	Read-only
20	Actual IP address	String	Format: xx.xx.xx.xx	Read-only
21	Actual Net mask	String	Format: xx.xx.xx.xx	Read-only
22	Actual gateway	String	Format: xx.xx.xx.xx	Read-only
23	Actual MAC address	String	Format: XX:XX:XX:XX:XX:XX	Read-only Note: In some cases in addition to ':', '.' and '-' are used.
25	ATMEL mini-boot version	String	Format: MiniBoot 1.0.0.1: ([compilation date])	Read-only
27	EtherCAT Logical ID switches	String	Hexadecimal format: 0xYZ Y – MSB Switch 4-bit hexadecimal value. Z – LSB Switch 4-bit hexadecimal value. Note: Evaluated at power-up only.	Read-only
29	Burnt in PAL description	String		Read-only
30	User defined EtherCAT Switch-ID	String	Decimal format: 0 - 65535	Read/Write
31	Product information (same as WS[30])	String	Hexadecimal format: To write lead with password: "Elmo". Example: Elmo0x1400A.	Read-only Saved on flash memory
			Bit	Description
			0 to 7	Product Name
			8 to 11	Reserved for product name
			12 to 13	Always 0
			14	Project (always 1 for NG (Gold Line))



			15	Always 0	
			16	CAN integrated	
			17	0: EtherCAT 1: TCP/IP	
			18 to 20	Feedback type: 0: E type (Encoder + Encoder, Analog sensor) 1: A type (Absolute + Encoder, Analog sensor) 2: R type (Encoder, Analog sensor + Resolver) 3: M type (Absolute + Resolver)	
			21	Define R type drive: Current saturation stays on PL	
			22	EtherCAT ID switches	
			23	Reserved	
			24	Ethernet hardware: 0: Absent 1: Present	
			25 to 31	Reserved	

Indices 2, 3 and 7 are for internal use.

All indices that are not listed include the following logic:

- A Read operation will return 0.
- A Write operation will return an error, i.e., 19?; - "Command syntax error"



AC[N] – Set Acceleration

AC[N] specifies the maximum allowed profiler acceleration

CANopen/CoE

AC[1] Profile_acceleration: **0x6083**Elmo_parameters_objects : **0x3002**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	Effective on the next call to BG[N]
Range	1...(2 ³¹)-1
Index range	1 to 4
Default	1,000,000,000
Unit modes	UM = 5, UM = 2
Non-volatile	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 fed to object **0x6083** during power-up and when the Begin Motion (**BG[N]**) command is used.

The **AC[N]** command does not affect time-dependent motion, such as Interpolated Position or Cyclic Synchronous Mode (See the MAN-G-DS402 Manual).



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[] – Set Analog Dead Band

AD[] specifies the dead band set for analog input 1.

CANopen/CoE

Elmo_parameters_objects: **0x3003**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Range	±10V
Index range	1 , 2
Default	AD[1] = 0 AD[2] = 0
Unit modes	All
Non-volatile	Yes

Remarks

The **AD[]** command sets the dead band of the analog input in the range from **AD[1]** to **AD[2]**.

While the analog input is in the dead band, the reading is ignored.

The sensor reading output slope will start from 0 at the dead band. For example, if the input is greater than **AD[2]**, then, $Reading = (input - AD[2]) * AG[1]$.

AD[] does not affect analog input 2.

Indices

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

Index	Description	Type	Units	Restrictions
1	Negative dead band	Float	Volts	
2	Positive dead band	Float	Volts	

References

[AS\[\]](#), [AR\[\]](#), [AG\[\]](#)



AG[] – Set Analog Gain

AG[] specifies the analog gain set for analog input 1.

CANopen/CoE

Elmo_parameters_objects: **0x3006**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Range	The total command ($AN[1] * AG[1]$) cannot exceed the applicable reference limit: Current ($AR[1] = 1$): $\pm MC$ Velocity ($AR[1] = 2$): $\pm VH[2]$ Position ($AR[1] = 3$): $VL[3]$ to $VH[3]$ Please refer to the applicable commands for more details.
Index range	[1]
Default	AG[1] = 0.1000000
Unit modes	All
Non-volatile	Yes

Remarks

The **AG[]** command sets the conversion gain for converting the input voltage to the specific units that are used (according to **AR[]**).

The conversion gain can be set to a current, velocity or position reference.

For example, if analog input 1 is used for velocity ($AR[1] = 2$), the value set by **AG[1]** is the unit conversion factor for converting from voltage to velocity in counts/sec. In this example if the value set by **AG[1]** is 1,000, this means that every 1 volt in the input is applied as a velocity command of 1,000 counts/sec.

The analog input is read by the **AN[]** command every real time as a general-purpose input. In order to map the analog reading as a reference signal, a socket must be used. Please refer to **CA[41]...CA[44]** for the functional mapping of the sockets. The analog input identification number is 16 (see example below).

After the socket is selected, **AR[]** is used to determine the usage of the analog input. Please refer to **AR[]** for more details.

For example, to set analog input 1 as a current reference, the following should be done:



1. Select a free socket (e.g., socket 4).

2. Disable the motor.

MO=0

3. Set the drive to current loop.

UM=1

4. Map socket 4 to an analog input reference.

CA[44]=16

5. Specify that the units of analog input 1 are for a current (torque) reference.

AR[1]=1

6. Specify that every 1 volt reading of the analog input will produce a 0.1 ampere reference command.

AG[1]=0.1

The direction of the reference can be modified by changing the sign of **AG[]** from minus (-) to plus (+) and vice versa.

In order to use analog input 1 as a reference, the selected socket should be set to 16.

When used as a reference, the analog entry can be filtered using **KV[71]** to **KV[75]**.

Indices

The following table describes the operation options for **AG[]**:

Index	Description	Type	Units	Restrictions
1	Gain of analog input 1	Float	Defined units/volt	

References

[AD\[\]](#), [AR\[\]](#), [AS\[\]](#), [CA\[\]](#), [KV\[\]](#)



AN[] – Get Analog Input

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

CANopen/CoE

Elmo_parameters_objects: **0x300D**

Attributes

Attribute	Description
Type	Float, Read-only
Source	All
Restrictions	None
Range	See the table below.
Index range	1 to 6
Default	N/A
Unit modes	All
Non-volatile	No

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 table below.

Indices

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

Index	Description	Units	Values	Restrictions
1	Reads analog input 1. When RM = 1, the value is used as an auxiliary reference for a motion command (refer to AG[]).	Volts	+/- 10 0	
2	Analog input 1 readout	A2D ticks	0 - 4095	
3	Reads the instantaneous current feedback from motor phase A	Amperes	N/A	The range of values depends on the MC of the drive.



4	Reads the instantaneous current feedback from motor phase B	Amperes	N/A	The range of values depends on MC of the drive
5	Reads the instantaneous current feedback from motor phase C	Amperes	N/A	The range of values depends on the MC of the drive
6	Reads the bus voltage	Volts	N/A	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	N/A	The range of values depends on MC of the drive
8	Reads the 5V digital voltage supply to the CPU	Volts	5.0	

References

[AG\[\]](#), [AS\[\]](#), [BV](#), [MC](#)



AR[] – Set Analog Units

AR[] specifies the function of the analog input when used as a reference command.

CANopen/CoE

Elmo_parameters_objects: **0x3011**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Range	1 to 3
Index range	[1]
Default	AR[1] = 3
Unit modes	All
Non-volatile	Yes

Remarks

AR[M] determines the mapping of the analog input N to either Current, Velocity or Position reference. This is used when the analog input serves as an auxiliary command or feedback for one of the mentioned references.

After the socket number (refer to the **CA[41]** to **CA[44]** commands) is determined, **AR[]** is used to select the reference mode. Please refer to the **AG[]** command for an example.

The analog input can be served as a feedback for Tachometer and Potentiometer readings. In this case the selected socket should be defined by **CA[45]** or **CA[46]**, respectively.

Indices

The units which are used by the reference generator are according to the value of **AR[]** as follows:

Index	Description	Type	Units		Restrictions
1	Units of analog input #1	Long	1	Ampere (for current)	
			2	Counts/sec (for velocity)	
			3	Counts (for position)	
2	Reserved				

References

AD[], AG[], AS[],CA[]





AS[N] – Analog Input Offset

AS[] compensates the offset of analog input *N*.

CANopen/CoE

Elmo_parameters_objects: **0x3012**

Attributes

Attribute	Description
Type	Float
Source	All
Restrictions	None
Range	According to the table
Index range	[1]
Default	0

Remarks

The analog input of the drive is offset during production of the drive. However, in some cases 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*, a measurement will also output a zero voltage.

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

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

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

The Gold drives include a single analog reference.

Indices

The following table describes the analog input 1 entry:

Index	Description	Type	Values	Restrictions
1	Offset value of input 1	Float	+/- 10.0 volts	
2	Offset value of input 2	Long	± 4095	

References

AD[], AG[], AN[], AR[]



BG[N] – Begin Motion

BG[N] starts the next profiled motion.

CANopen/CoE

Elmo_parameters_objects: **0x3020**

Attributes

Attribute	Description
Type	Command
Source	All
Restrictions	N/A
Range	N/A
Index range	1 to 4
Default	N/A
Unit modes	UM = 5, UM = 3, UM = 2
Non-volatile	N/A

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** (it can also be retrieved using **OV[2]**).



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

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, UM = 5	JV	Profile Velocity, 3	AC, DC, SF
JP	UM = 5	JP	Profile Position, 1	AC, DC, SF

Note: **UM** is the minimum unit mode for the relevant motion. Refer to the **UM** command.

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

Elmo Command	Converts to CANopen Object	Action Performed	Note
AC	0x6083	Saturation to maximum acceleration	
DC	0x6084	Saturation to maximum acceleration	
JV	0x60FF	- Motion mode to: Profile Velocity - Saturation to maximum speed	
JP	0x6081 0x6082	- Motion mode to: Profile Position - Saturation to maximum speed	
-	0x607E	Set to 0 indicating: do not convert polarity	
SP	0x6081	Saturation to maximum speed	
FS	0x6082	Final speed at target position	
PA/PR	0x607A	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

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



BH – Get Recorder Signal

BH uploads the values recorded by the recorder to a host in hexadecimal format.

CANopen / CoE

Recorder_data: **0x2030**

Attributes

Attribute	Description
Type	Command, read-only
Source	All, except FoE
Restrictions	- Valid recorded data is ready (i.e., RR == 0). - No other uploading sequence is performed (UL).
Range	1 to 2^{16} , bit field format
Default	No
Unit modes	All
Non-volatile	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 was successfully finished and the **RR** command, which launches and retrieves the status of the recorder, is equals to 0.

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 Hex-Binary text format. The format transfers each number 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:

- A data header
- A data value

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

Byte	Description
0 to 1	Variable type (0, Long; 1, float)
2 to 3	Variable size: (2, short; 4, Long; 8, double)
4 to 7	Transmitted data length



8 to 11	Data sampling time in TS multiplier (1, every TS; 2, every TSx2). The sampling intervals are equal to RG * sampling multiplier.
12 to 19	Floating factor to convert the data from internal units to physical units (not user units). Factors are used to convert the following from internal units: Current: (e.g., "Active Current") to amperes Velocity: (e.g., "Velocity Command") to counts/sec Voltage: (e.g., "Bus Voltage") to volts The user must multiply the data by the factor to obtain the actual value.
20 to transmit data length	Data of the specific uploaded signal

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.

CANopen/CoE

Elmo_parameters_objects: **0x3029**

Attributes

Attribute	Description
Type	Long, 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

Remarks

The drive allows the application to use a brake to hold the motor while 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 flows through 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.

Disabling the servo

When the servo is disabled by setting **MO** = 0, the brake is engaged, and the corresponding indication (**SO** = 0) is received only after the time set by **BP[1]**. Please refer to the **SO** command.

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 **SO** (Servo On) indication should be set to 1 only after **BP[2]** msec 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, Overtemperature, 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 250 µs.



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 (a change from MO = 1 to MO = 0) 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 (a change from MO = 0 to MO = 1) 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	

References

OL[], OP



BS[] – Bring Recorded Sample

BS[N] obtains the $(N + 1)$ th sample from a pre-selected recorded data vector.

CANopen/CoE

Elmo_parameters_objects: **0x302C**

Attributes

Attribute	Description
Type	32-Bit (short, long or float), 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

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.

The **BS[]** command always returns a long variable. The variable can be a Long or float type. In the case of a short variable, the upper bits are padded with zeros.

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** command for more details.



Indices

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

Index	Description	Type	Values	Restrictions
0	First sample	Long		
1	Second sample	Long		
...	...	...		
N	$(N + 1)$ th sample	Long		

References

[RR](#), [RL](#), [RP\[11\]](#)



BT – Begin Motion On Time (Reserved)

CANopen/CoE

Elmo_parameters_objects: **0x302D**

Attributes

Attribute	Description
Type	
Source	
Restrictions	
Range	
Default	
Unit modes	
Non-volatile	

Remarks

References



BV – Bus Voltage

BV gets the maximum drive bus voltage in volts.

CANopen/CoE

Elmo_parameters_objects: **0x302F**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	None
Range	BV > 0 (The bus voltage depends on the drive and has no range.)
Default	200 V
Unit modes	All
Non-volatile	Constant

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 undervoltage threshold (**WI[37]**) and the factory overvoltage threshold (**WI[35]**) 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[13]** for the undervoltage

The actual values of the voltage thresholds can be read by calling **WI[36]** for the overvoltage and **WI[38]** for the undervoltage.

References

[AN\[\]](#), [WI\[35\]](#), [WI\[36\]](#), [WI\[37\]](#), [WI\[38\]](#), [XP\[1\]](#), [XP\[13\]](#)



CA[N] – Commutation Array

CA[] sets the commutation and sensor feedback configuration.

CANopen/CoE

Elmo_parameters_objects: **0x3034**

Attributes

Attribute	Description
Type	Long
Source	All
Restrictions	According to array index
Range	According to array index
Index range	According to array index
Default	CA[4] = 1 CA[5] = 2 CA[6] = 3 CA[13] = 32765 CA[17] = 1 CA[18] = 4000 CA[19] = 0 CA[22] = -1 CA[31] = 10 CA[32] = 1 CA[34] = 1 CA[35] = 17 CA[36] = 25000000 CA[40] = 0 CA[41] = 2 CA[45] = 1 CA[46] = 1 CA[47] = 1 CA[48] = 15669722 CA[49] = 18633 CA[50] = 10000000 CA[51] = 10000000 CA[65] = 1



Attribute	Description
	CA[79] = 1 CA[84] = 2 CA[102] = 1
Unit modes	All
Non-volatile	Yes

Remarks

The **CA[]** command sets the socket functionalities, as well as the sensor and commutation parameters. Depending on the command index, the motor should be turned off and/or the commutation should be reset and recalculated on the next motor on.

Indices

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

Note: The default is 0 unless otherwise stated.

Index	Description	Default	Values	Restrictions/Notes
1 to 3	Polarity of Hall sensors A, B and C, respectively		0, 1	The motor must be off to change the setting. Changing the setting resets commutation.
	0 Reverse the Hall sensor polarity			
	1 Do not reverse the Hall sensor polarity			
4 to 6	Correlate between Hall sensors and motor phases A, B and C, respectively	CA[4] = 1 CA[5] = 2 CA[6] = 3	1 to 3 (for Hall sensors A, B and C)	Each Hall sensor must be assigned to a different motor phase. The motor must be off to change command. Changing command resets commutation.
7	Phase offset relative to the absolute commutation sensor (Hall, Absolute Serial, Analog Hall and Resolver) in stepper units		-512 to 512	The motor must be off to change the setting. Changing the setting resets commutation.



Index	Description	Default	Values	Restrictions/Notes
8	Ignore the encoder error. Only valid for General Biss, Panasonic, Kawasaki, Yaskawa, Sanyo, Nikon, EnDat, Tamagawa and SSI encoders. For EnDat 2.2 selected bits of the error are masked.		0 – 0xFFFF	The motor must be off to change the setting. Changing the setting resets commutation.
9	Phase shift compensation in a Sin/Cos sensor. Units are the factor of the sine coefficient/ 2^{15} .			The motor must be off to change the setting. Changing the setting resets commutation.
10	Serial Exclusive #2 – Number of Data Bits		0 to 24	
11	Analog encoder sine signal offset in ADC units		0 to 4500	The motor must be off to change the setting. Changing the setting resets commutation.
12	Analog encoder cosine signal offset in ADC units		0 to 4500	The motor must be off to change the setting. Changing the setting resets commutation.
13	Analog encoder sine gain to fit the cosine signal. Units are $1/(2^{15})$	32765	10000 to 62000	Gain 1 is 32765. The motor must be off to change the setting. Changing the setting resets commutation.
14	Command Reference Multiplier (Position and Velocity) Multiplies the assigned socket command input. Socket reference function is according to CA[68] or CA[69] .		0 to 2147483647	



Index	Description	Default	Values	Restrictions/Notes		
15	Command Reference Divider (Position and Velocity) Divides the assigned socket command input by $2^{CA[15]}$. Socket reference function is according to CA[68] or CA[69] .		0 to 30			
16	Force commutation search for every motor on		0	0 to 7		
	Bit	Description				
	0	Force Commutation search				
		0				Use commutation when known
		1				Force commutation search for every motor on
	1	Ignore illegal Hall				
		0				Output an error if Hall is illegal
		1				Output a warning (in Object 0x2085 bit 10) in case of illegal Hall
	2	Ignore Hall + encode mismatch				
		0				If field angle of Hall and field angle of encoder differ more than CA[111] and CA[111] is not zero, then output an error
		1				If field angle of Hall and field angle of encoder differ more than



Index	Description		Default	Values	Restrictions/Notes
		CA[111] and CA[111] is not zero, then output a warning in bit 11 in Object 0x2085			
17	Commutation method		1	1 to 6	The motor must be off to change the setting. Changing the setting resets commutation.
	1	By Hall sensor			
	2	By stepper, motor will move to a certain stepper position			
	3	By binary search, minimal movement when commutation is not known			
	4	By analog Hall sensor			
	5	By serial absolute encoder			
	6	By virtual absolute Gurley			
	7	Slave including commutation by PAL			



Index	Description		Default	Values	Restrictions/Notes
18	Feedback counts per electrical cycles. The value is used to determine the number of counts per electrical cycles (CA[19]) of the commutation socket. In rotary motion: counts per revolution In linear motion: counts per single electrical cycle In analog feedbacks: the value obtained after multiplication by the factor read from CA[31]		4000	6 to 2147483647	The motor must be off to change the setting. Changing the setting resets commutation.
19	Number of motor pole pairs In rotary: pole pairs per revolution In linear: usually equal to one		0	Positive values	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
20	Use Digital Halls			0 to 3	The motor must be off to change the setting. Changing the setting resets commutation.
	0	Do not read Hall sensors			
	1	Read Digital Hall sensors			
	2	Read Yaskawa Hall			
	3	Read Panasonic incremental Hall			
21	Gurley encoder resolution			10 to 12	The motor must be off to change the setting. Changing the setting resets commutation.
22	Number of the error bit in the SSI protocol bits, or -1 if no error bit exists.		-1	-1 to 29	The motor must be off to change the setting. Changing the setting resets commutation.



Index	Description		Default	Values	Restrictions/Notes
23	SSI error bit logic			0, 1	The motor must be off to change the setting. Changing the setting resets commutation.
	0	High indicates an error			
	1	Low indicates an error			
24	Hiperface's analog and absolute signals direction are:			0, 1	
	0	Same direction			
	1	Opposite directions			
25	Reverse the direction of stepper angle, equivalent to switching the motor cables between M2 and M3			0, 1	The motor must be off to change the setting. Changing the setting resets commutation.
	0	Do not reverse			
	1	Reverse the direction			
26	Socket number used for adding a current reference command (same as CA[70]). Used also for identify Gantry plant.			1 to 4	Notes: - Stop manager functions do not apply for this entry. - In situations where the brake is used (BP[]) the reference command might be active regardless of the brake state.
27	Socket number used for adding a Yaw current reference command			1 to 4	Notes: - Stop manager functions do not apply for this entry. - In situations where the brake is used (BP[]) the reference command might be active regardless of the brake state. - Only with Gantry master (sensor ID=20)



Index	Description		Default	Values	Restrictions/Notes
28	Motor type:			0 to 6	If CA[28] is set to 1 or 3, UM (unit mode) should not be 3 (stepper). The motor must be off to change the setting. Changing the setting resets commutation.
	0	Rotary brushless			
	1	Rotary DC brush			
	2	Linear brushless			
	3	Linear voice coil			
	4	Rotary two-phase			
	5	Reserved			
	6	Linear two-phase			
29	Changing commutation offset on the fly, in stepper units		0	-512 to 512	
30	Serial Exclusive #2/3 Configuration:		0		
	Bits	Description			
	0-3	SSI start delay. Units are in clocks		0 to 15	
	4-7	Clock Frequency		0 – 2.5 MHz 1 – 1.25 MHz 2 – 625 kHz	
	8-14	Input Glitch Filter, see CA[35] for calculation		2 to 63	
	15	Reserved			
	16-20	Protocol Total Bits - 1		0 to 23	
31	Analog signal multiplication factor. Each Sin/Cos cycle is equal to $2^{CA[31]}$ counts		10	2 to 16	The motor must be off to change the setting. Changing the setting resets commutation.
32	Number of resolver pole pairs per revolution		1		The motor must be off to change the setting. Changing the setting resets commutation.
33	Resolver excitation signal offset. The offset units are equal to the excitation signal clock units [150 MHz].			Positive value	The motor must be off to change the setting. Changing the setting resets commutation.



Index	Description	Default	Values	Restrictions/Notes
34	Resolver N cycle interpolation. The value determines the resolver position interpolation and the frequency of the excitation signal $f_E = f_{TS}/(2*N)$, or $f_E = f_{TS}$ for $N = 0$	1	0 to 4	The motor must be off to change the setting Changing the setting resets commutation
35	PAL glitch filter for absolute sensor (and Quad in PAL, ID=36) reading. Glitch filter value is (CA[35]+1)*13.333 nanosecond	17	2 to 255	Need to reset the sensor by CA[41 to 44]
36	Clock frequency to PAL (Hz). PAL divide this value by 10	25000000	3120000, 6250000, 12500000, 25000000, 40000000, 50000000	Need to reset the sensor by CA[41 to 44] Each sensor has its own range of frequency
37	Error Mapping Source Socket		0 to 4	Change position of one sensor(FP[CA[38]]) by the value of different sensor FP[CA[37]] , with factor defined in CA[14] and CA[15]
38	Error Mapping Destination Socket		0 to 4	Dest = Dest+(Src* CA[14])>> CA[15]
39	Serial Exclusive #2 Dead Zone		0 to 8196	
40	Maximum time for interpolating field angle with Hall only sensor. Time is in milliseconds.	0	0-2000	Zero value is the same as 2000 ms



Index	Description	Default	Values	Restrictions/Notes
41–44	ID of sensor connected to socket CA[41]: socket number 1 CA[42]: socket number 2 CA[43]: socket number 3 CA[44]: socket number 4	CA[41] = 2	1 to 36	Sensor ID is unique. The same sensor ID cannot be set for two sockets. If the socket is used for commutation, the motor must be off to change the setting, and changing the setting resets commutation.
	1	Quad Encoder Port B		
	2	Quad Encoder Port A		
	3	Analog Sin/Cos		
	4	Digital Hall only		
	5	Serial Absolute Biss		
	6	Serial Absolute Panasonic		
	7	Serial Absolute Mitutoyo		
	8	Virtual two sine signal		
	9	Serial Absolute EnDat		
	10	Serial Absolute Tamagawa		
	11	Pulse & Direction port B		
	12	Pulse & Direction port A		
	13	Quad encoder port B used for emulated feedback		
	14	Quad Port A used for emulated feedback		
	15	Copy profiler to socket		
	16	Analog Input #1		
	17	Virtual Absolute Gurley		
	18	Absolute SSI		
	19	Serial (absolute or incremental) Yaskawa		
	20	Gantry Master		
	21	Serial Exclusive		
	22	Resolver		
	23	Serial absolute Kawasaki		
	24	Serial Absolute General Biss		



Index	Description	Default	Values	Restrictions/Notes
	25			
	Serial Absolute Sanyo/Nikon			
	26			
	Simple profiler			
	27			
	Copy gantry differential position			
	28			
	Serial Hiperface			
	29			
	Fast Reference 0x2005			
	30			
	Serial Absolute EnDat 2.2			
	31			
	Panasonic incremental			
	32			
	Serial Exclusive #2			
	33			
	Serial Exclusive #3			
	34			
	Stepper Closed Loop (with Feedback)			
	35			
	PWM Duty Cycle			
	36			
	Quad Port A (without capture)			
45	Socket number used for position loop	1	1 to 4	The motor must be off to change the setting.
46	Socket number used for velocity loop	1	1 to 4	
47	Socket number used for commutation	1	1 to 4	The motor must be off to change the setting. Changing the setting resets commutation.
48	Maximum allowed amplitude square for analog Sin/Cos and resolver encoder, in (ADC-offset) ²	1863226	1863225 to 15669722	The motor must be off to change the setting. Changing the setting resets commutation. For sin/cos amplitude square of (0.5Volt) ² the parameter equal to (3085/2) ² For resolver amplitude square of (2.5Volt) ² the parameter equal to (3481/2) ²



Index	Description	Default	Values	Restrictions/Notes
49	Minimum allowed amplitude square for analog Sin/Cos and resolver encoder, in (ADC-offset) ²	18633	18632 to 1863225	The motor must be off to change the setting. Changing the setting resets commutation.
50	Glitch filter of digital input in port B, in counts/sec. The hardware calculates pulse width per each input (A or B) equal to 2/CA[50] for the same value of input.	10000000	120000 to 75000000	
51	Glitch filter of digital input in port A, in counts/sec. The hardware calculates pulse width per each input (A or B) equal to 2/CA[50] for the same value of input.	10000000	120000 to 75000000	
52	Planar: check at motor off that Sin/Cos amplitude square is above this value		non-negative	Set WS[16] and object 0x2085 bit 1 0 – below 1 – above or equal See CA[48] for conversion factor
53	Stepper angle command source in stepper mode (UM =3,6)	0	0 to 5	0 – main profile 1 to 4 – socket number 5 – position reference (defined by CA[68])
54–57	Invert direction of sensor connected to socket CA[54] : socket number 1 CA[55] : socket number 2 CA[56] : socket number 3 CA[57] : socket number 4		0, 1	If the socket is used for commutation, the motor must be off to change the setting, and changing the setting resets commutation. Does not work for Hall Only (ID=4)
	0	Do not invert		
	1	Invert the direction		



Index	Description	Default	Values	Restrictions/Notes	
58	Number of high-resolution bits to mask from a serial encoder. This is used when the total bits are more than what the drive can read. For a linear encoder or a single-turn rotary encoder the number of bits that the user sees is: CA[59] – CA[61] – CA[58] For a rotary encoder with multi-turn the total bits that the user sees is: CA[59] + CA[62] – CA[61] – CA[58]		0 to 8	The motor must be off to change the setting. Changing the setting resets commutation.	
59	Resolution of serial encoder. For a rotary motor this value defines the single-turn resolution. For linear motor this value defines the total bits arriving in the protocol.		1 to 32	The motor must be off to change the setting. Changing the setting resets commutation.	
60	CA[60] Sensor configuration		0 to 255	The motor must be off to change the setting. Changing the setting resets commutation.	
	Bit	Description			
	0	<i>Biss encoder</i>			
		0			Do not use temperature readings.
		1			Use temperature readings.
	1	<i>SSI sensor</i>			
		0			Gray code is not used
		1			Gray code is used
	2	<i>Yaskawa sensor</i>			
		0			Incremental encoder
		1			Absolute encoder
	3	<i>Serial encoder</i>			
		0			Sample serial encoder every 2* TS
1		Sample serial encoder every TS			



Index	Description	Default	Values	Restrictions/Notes
	4			
	0			
	1			
	5			
	0			
	1			
	6			
	0			
	1			
	7			
	0			
	1			
61	Reducing resolution from a serial absolute sensor, reduces the resolution by masking low-resolution bits. For a linear encoder or a single-turn rotary encoder the number of bits that the user sees is: CA[59] – CA[61] – CA[58] For a multi-turn rotary encoder the total bits that the user sees is: CA[59] + CA[62] – CA[61] – CA[58]		0 to 12	The motor must be off to change the setting. Changing the setting resets commutation.
62	Multi-turn bits in a serial absolute encoder		0 to 16	The motor must be off to change the setting. Changing the setting resets commutation.



Index	Description	Default	Values	Restrictions/Notes
63	Adjust the synchronization mask This command moves (back or forward) the synchronization mask for communication with the serial absolute encoder by the time specified in units of 6.667 nanoseconds (clock running at 150 MHz).		-1000 to 1000	
64	Reserved			
65	Socket number used for source of position gain schedule	1	1 to 4	
66	Number of bits that are transmitted in the SSI protocol		0 to 64	The motor must be off to change the setting. Changing the setting resets commutation.
67	Position of the LSB within the SSI protocol bits		0 to 56	The motor must be off to change the setting. Changing the setting resets commutation.
68	Socket number used for adding a position reference command		0 to 4	Notes: - Stop manager functions do not apply for this entry. - In situations where the brake is used (BP[]) command reference might be active regardless of the brake state. The motor must be off to change command.
69	Socket number used for adding a speed reference command		0 to 4	Notes: - Stop manager functions do not apply for this entry. - In situations where the brake is used (BP[]) the reference command might be active regardless of the brake state.



Index	Description	Default	Values	Restrictions/Notes
70	Socket number used for adding a current reference command		0 to 4	Notes: - Stop manager functions do not apply for this entry. - In situations where the brake is used (BP[]) the reference command might be active regardless of the brake state.
71–74	Number of points for FIR filter in socket CA[71]: socket number 1 CA[72]: socket number 2 CA[73]: socket number 3 CA[74]: socket number 4		0 to 8	If the socket is used for commutation, the motor must be off to change the setting, and changing the setting resets commutation.
75–78	Filter type in socket CA[75]: socket number 1 CA[76]: socket number 2 CA[77]: socket number 3 CA[78]: socket number 4		0, 1	If the socket is used for commutation, the motor must be off to change the setting, and changing the setting resets commutation.
	0			
	1			
79	Socket number used for additional sensor converted to DS-402 master		0 to 4	Cannot be change if ECAM is active
80	Socket number used as slave in Gantry master control		0 to 4	Must be set if Gantry master is used
81	Socket number used as master in Gantry master control		0 to 4	Must be set if Gantry master is used
82	Drive designation in Gantry mode		0 to 8	Must be set if Gantry (master or slave) is used 0 – Default. No Gantry 1 – Gantry master 2 – Gantry slave 3 – Y master in planar 4 – Y slave in planar 5, 6 – reserved 7 – Current command and



Index	Description	Default	Values	Restrictions/Notes
				Field angle master 8 – Current command and Field angle slave
83	Communication through Port A or B between Gantry drives		0, 1	
	0 Port A			
	1 Port B			
84	Communication frequency between Gantry drives	2	0, 1, 2	
	0 2.5 MHz			
	1 3.75 MHz			
	2 5 MHz			
85	Gantry motor ON sequence.		0 to 4	Reserved (internal use)
	0 No sequence at motor on			
	1 Reset Gantry position to zero at motor on			
	2 Reset Gantry position to px (average) at motor on			
	3 Use Gantry sensor for PWM synchronization only			Need to set only Serial Exclusive sensor in each drive
	4 Reserved (internal use)			
86	SSI start delay. Units are in clocks		0 to 15	The motor must be off to change the setting. Changing the setting resets commutation.
87	Touch Probe socket selection		0 to 4	Socket change is not allowed while capture is enabled
88	Number of points in Position FIR filter. 0 and 1 disable the position FIR filter		0,1,2,4,8,16	The motor must be off to change the setting. See GS[8] for options.
89	Planar: speed for alignment search in stepper units for Y1 axis			
90	Planar: max stepper angle for alignment search for Y1 axis			



Index	Description	Default	Values	Restrictions/Notes
91-94	Offset of position in socket CA[91] : socket number 1 CA[92] : socket number 2 CA[93] : socket number 3 CA[94] : socket number 4			
95	Communication time with serial sensors. Zero is by default: If communication frequency is 2.5 MHz communication time is 79 microseconds, otherwise communication time is 60 microseconds Communication time should be 5 microseconds less than reading time cycle of the sensor (see TS and GS[60].3)		0 to 125	
96	Resolution for non-binary resolution with General BiSS sensor		0 to 2147483647	
97	Socket number used for adding a position reference command for Gantry Yaw		0 to 4	Notes: - Stop manager functions do not apply for this entry. - In situations where the brake is used (BP[]) command reference might be active regardless of the brake state.
98	Position Command for additional controller (sensor ID=34)	0	0 to 5	0 – main profile 1 to 4 – socket number 5 – position reference (defined by CA[68])
99	Position Feedback for additional controller (sensor ID=34)	0	0 to 4	0 – None 1 to 4 – socket number
100	PWM Duty Cycle Pin Selection		1 – A, 2 – B, 3 – Index	Configuration will be applied when setting the PWM Duty Cycle in CA[41-44]



Index	Description	Default	Values	Restrictions/Notes
101	Units of PWM Duty Cycle		0 – None 1 – Current 2 – Velocity	Configuration will be applied when setting the PWM Duty Cycle in CA[41-44]
102	Minimum Valid PWM frequency in Hz	1	>0	
103	Minimum PWM Duty Cycle in milli-Percent		0 to 100000	Configuration will be applied when setting the PWM Duty Cycle in CA[41-44]
104	Maximum PWM Duty Cycle in milli-Percent		0 to 100000	Configuration will be applied when setting the PWM Duty Cycle in CA[41-44]
105	Minimum PWM Duty Cycle Value in milA or counts per second			Configuration will be applied when setting the PWM Duty Cycle in CA[41-44]
106	Maximum PWM Duty Cycle Value in milA or counts per second			Configuration will be applied when setting the PWM Duty Cycle in CA[41-44]
107	PWM Duty Cycle Dead-Band in milA or counts per second			
108	PWM Duty Cycle Signal Type		0 – PWM only 1 – Reserved	Configuration will be applied when setting the PWM Duty Cycle in CA[41-44]
109	Value for invalid PWM frequency (below minimum set in CA[102]) in milA or counts per seconds			
110	Socket number used for homing. Homing procedure consider this socket for latched position. Zero (0) is the default where the position socket is used. This parameter is used for Gantry where the position is virtual and the latched position is from the master feedback. The user can change the value using OV[54] without saving it to Flash.		0 to 4	Will be applied for the next homing procedure
111	Hall + encoder mismatch field angle, in stepper units		0 to 171 (120 degree)	If mismatch is above this value, drive outputs an error or warning according to



Index	Description	Default	Values	Restrictions/Notes
				CA[60].2. If CA[111] is zero this function is disabled
112	Threshold speed for sin/cos sensor, in [Hz] of sin/cos cycle.			When speed of sin/cos cycle is above this threshold speed drive calculates position and velocity without interpolation and does not update and check sin/cos amplitude. When the speed drops below 0.9 from this value, drive counts as regular. If this value is zero, this function is disabled

Grouped by sensor type

Sensor ID	Sensor	Entries
1	Quad port B	CA[50]
2	Quad port A	CA[51]
3	Sin/Cos	CA[9], CA[11], CA[12], CA[13], CA[31], CA[48], CA[49], CA[50]
4	Hall	CA[1 to 7], CA[20], CA[40]
5	Biss	CA[8], CA[35], CA[36], CA[58 to 62]
6	Panasonic	CA[8], CA[35], CA[36], CA[58], CA[59], CA[61], CA[62]
7	Mitutoyo	CA[35], CA[36], CA[58], CA[59], CA[61], CA[62]
8	Virtual two sine signals	SE[1 to 7], TW[80]
9	EnDat	CA[8], CA[35], CA[36], CA[58], CA[59], CA[61], CA[62]
11	Pulse & Direction port B	CA[50]
12	Pulse & Direction port A	CA[51]
13	Emulation port B	CA[50]
14	Emulation port A	CA[51]
15	Copy profiler to socket	
16	Analog Input #1	AD[1,2], AG[1], AR[1], AS[1]



Sensor ID	Sensor	Entries
17	Gurley	CA[9], CA[11], CA[12], CA[13], CA[21], CA[31], CA[48], CA[49]
18	SSI	CA[8], CA[22], CA[23], CA[35], CA[36], CA[59], CA[61], CA[62], CA[66], CA[67] , CA[86]
19	Yaskawa	CA[8], CA[20], CA[35], CA[36], CA[59], CA[58], CA[60], CA[61], CA[62]
20	Gantry master	CA[80], CA[81], KP[5], KP[4], KI[4], US[4], KP[6], ER[5], KV[81 to 90], CA[60].7
21	Serial Exclusive	CA[82], CA[83], CA[84]
22	Resolver	CA[9], CA[11], CA[12], CA[13], CA[31], CA[32], CA[33], CA[34], CA[48], CA[49], CA[50]
23	Kawasaki	CA[8], CA[35], CA[58], CA[59], CA[61], CA[62]], CA[8]
24	General Biss	CA[8], CA[22], CA[23], CA[35], CA[36], CA[59], CA[61], CA[62], CA[66], CA[67]
25	Sanyo/Nikon	CA[8], CA[35], CA[36], CA[58], CA[59], CA[61], CA[62]
26	Simple profiler	
27	Copy gantry differential position	
28	Serial Hiperface	CA[9], CA[11], CA[12], CA[13], CA[24], CA[31], CA[48], CA[49], CA[50], CA[58], CA[59], CA[62]
30	EnDat 2.2	CA[8], CA[35], CA[36], CA[58], CA[59], CA[61], CA[62]
31	Panasonic Serial Incremental	CA[8], CA[35], CA[36], CA[59], CA[61]
32	Serial Exclusive #2	CA[10], CA[14], CA[30], CA[39]
33	Serial Exclusive #3	CA[8], CA[30], CA[58], CA[59], CA[60], CA[61], CA[62], CA[67]
34	Stepper Closed Loop (With Feedback)	CA[53], CA[98], CA[99], KP[4,5,6], KI[4], KV[81..90], FF[2]
35	PWM Duty Cycle	CA[100], CA[101], CA[102], CA[103], CA[104], CA[105], CA[106], CA[107], CA[108], CA[109]
36	Quad port A (without	CA[35]



Sensor ID	Sensor	Entries
	capture)	

Grouped by commutation method

Method ID	Method	Entries
1	Hall	CA[1 to 7], CA[20]
2	Stepper	SC[1 to 5]
3	Binary search	SC[1], SC[2], SC[3], SC[6], SC[7]
4	Analog Hall	CA[7]
5	Serial absolute	CA[7]
6	Virtual absolute Gurley	CA[7], SC[1], SC[2], SC[5],
7	Absolute commutation by PAL	

References

MO, SC[], SE[], UM



CC – Compilation Checksum

CC signs the user program and allows it to be executed after successfully completing the downloading procedure.

CANopen/CoE

Elmo_parameters_objects: **0x3036**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All, except the user program
Restrictions	• The motor must be off. • The user program must not be running.
Range	None
Default	–1
Unit modes	All
Non-volatile	None

Remarks

In order to be able to run the user program after the downloading procedure, the host sets the **CC** command with the user program checksum. The checksum is calculated by the drive during the downloading procedure and is saved in the non-volatile memory of the drive. When the host calls the **CC** command, this checksum is compared with the given signature, and if the value matches, the program is loaded into the RAM and is ready to be executed by using the **XQ** command.

When a drive is powered up, **CC** is called automatically to allow updating of the user program without requiring the host to perform any action.

Depending on the size of the user program, the **CC** command typically launches a long process, during which background procedures, such as interpreters, should not be run. The existence of the user program will also increase the drive boot-up time after power-up or after drive reset.

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

If a user program exists in the non-volatile memory of the drive (**PS** = –1), **CC** is executed automatically during the drive boot-up (after power-up or drive reset), if **CC** fails, the **PS** command returns –2.

References

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



CD – CPU Dump

CD reads the CPU database status.

CANopen/CoE

Elmo_parameters_objects: **0x3037**

Attributes

Attribute	Description
Type	String, Read-only
Source	All, except the user program
Restrictions	None
Range	None
Default	Database OK
Unit modes	All
Non-volatile	No

Remarks

In some cases, when database processes return errors, the error should be reported in the **CD** command.

Databases are tested during some processes, such as drive boot-up (after power-up or drive reset), during motor enable (**MO** = 1 or "enable operation" of CANopen DS-402 state) or during the loading of parameters from non-volatile flash memory (after the **LD** command). If a parameter fails to be processed or encounters a conflict with another parameter, the event should be reported in the **CD** command.

Return Values

CD returns Database OK when all is OK.

CD returns Check Sum Error in cases in which there is a mismatch between the parameter structures in the non-volatile flash memory and in the RAM. This is typical for new drives (after production) and in a case of incompatibility in database size due to a new release.

CD returns Autoexec Fails in the case in which an autoexec routine exists in the user program and for some reason the routine could not be executed.

During a motor enable procedure an error might occur. In some cases the error will be "Motor Could Not start – reason in CD" (Error Code 48). In such cases **CD** returns the value which is described in MF. For example, if **CD** returns **MF** = 20480, it means that an overvoltage appeared during the motor enable sequence (before the motor was actually enabled). For the whole list, refer to the **MF** command.



References

LD, MF, SV



CL – Current Limit Parameters

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

CANopen/CoE

Elmo_parameters_objects: **0x303F**

Attributes

Attribute	Description
Type	Float
Source	All
Restrictions	None
Range	0 < CL[1] < MC (maximum current) 0 < CL[2] <= 100 0 < CL[3] <= 32000 0 < CL[4] <= 5000
Index range	1 to 4
Default	CL[4]: 3000
Unit modes	All
Non-volatile	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 Gold Drive Administrative 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]**.

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.

The minimum current limit is MC/128. If **CL[1]** < MC/128, the **CL[1]** value will be accepted, but the actual current value will be limited to MC/128.

Index	Description	Type	Units	Restrictions
1	Continuous current limit	Float	Ampere	
2	Motor stuck current level	Float	Percent (of CL[1])	
3	Motor stuck speed level - velocity	Float	Counts per second	
4	Motor stuck time-out - duration	Float	Millisecs	

References

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



CP – Clear Program

CP erases the user program.

CANopen/CoE

Elmo_parameters_objects: **0x3043**

Attributes

Attribute	Description
Type	Command, Write
Source	All, except the user program
Restrictions	• The motor must be off. • The user program must be stopped (KL or HP). • Wizard mode must not be active.
Range	None
Default	None
Unit modes	All
Non-volatile	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**



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_parameters_objects: **0x3046**

Attributes

Attribute	Description
Type	Long
Source	All
Restrictions	Commutation may be wrong
Range	0 to CA[18]
Index range	
Default	
Unit modes	3
Non-volatile	No
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=384; BG; // 384 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



CU – CPU Usage

CU calculates the present CPU usage.

CANopen/CoE

Elmo_parameters_objects: **0x3048**

Attributes

Attribute	Description
Type	Unsigned Long, Read
Source	All, except CoE
Restrictions	No
Range	0 to 100
Default	None
Unit modes	All
Non-volatile	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 2 msec. This affects the execution time of the user program, connected communication channels and other background tasks.

References



CW – Control Word

CW specifies the control word imitating the DS-402 control word value.

CANopen/CoE

Elmo_parameters_objects: **0x304A**

Attributes

Attribute	Description
Type	Bit Field, Read/Write
Source	USB, TCP
Restrictions	None
Range	N/A
Default	0
Unit modes	All
Non-volatile	No

Remarks

The control word is used in conjunction with the status word in the DS-402 CANopen standard for drives and motion. Typically, the control word is received in the CANopen or EtherCAT communication channel with object **0x6040**. The user can imitate the object behavior or check the present results from a host which uses CANopen or EtherCAT.

For more details about the **CW** bit field, refer to the CANopen DS-402 manual.

References



DC – Set Deceleration

DC specifies the maximum allowed deceleration of the main DS402 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

Profile_deceleration: **0x6084**Elmo_parameters_objects: **0x3050**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	Effective on the next call to BG
Range	1...(2 ³¹)-1
Default	1,000,000,000
Unit modes	UM = 5, UM = 2
Non-volatile	Yes

Remarks

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

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

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

The **DC** command does not affect time-dependent motion, such as Interpolated Position or Cyclic Synchronous Mode (see MAN-G-DS-402 Manual).

The acceleration and deceleration of the main profiler are subject to the limits of the **SD** value. If the **DC** value is higher than the **SD** value, the **SD** value is used, and the **DC** 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 – CAN controller status

DD returns the status of the CAN controller as unsigned long bit field value.

CANopen/CoE

Elmo_parameters_objects: **0x2082**

Attributes

Attribute	Description
Type	Read
Source	All
Restrictions	None
Range	0...0xFFFFFFFF
Index range	0, 1
Default	None
Unit modes	All
Non-volatile	No

Remarks

None

DD value reports

- The following table lists the errors reported by **DD[1]**.

Bits	31...25	24...16	15...8	7...0
Description	Reserved	Network status	CAN transmit error counter	CAN receive error counter

The bit fields are described in following table:

Bit	Meaning	Remarks
0-7	CAN receive error counter (REC)	The Rx error counter is increased when the receive channel detects an error. When the counter reaches 96, the warning bit is set (bit 16). When the counter reaches 127, the drive is in Error Passive state (bit 17) and does not produce any more Error Frames.



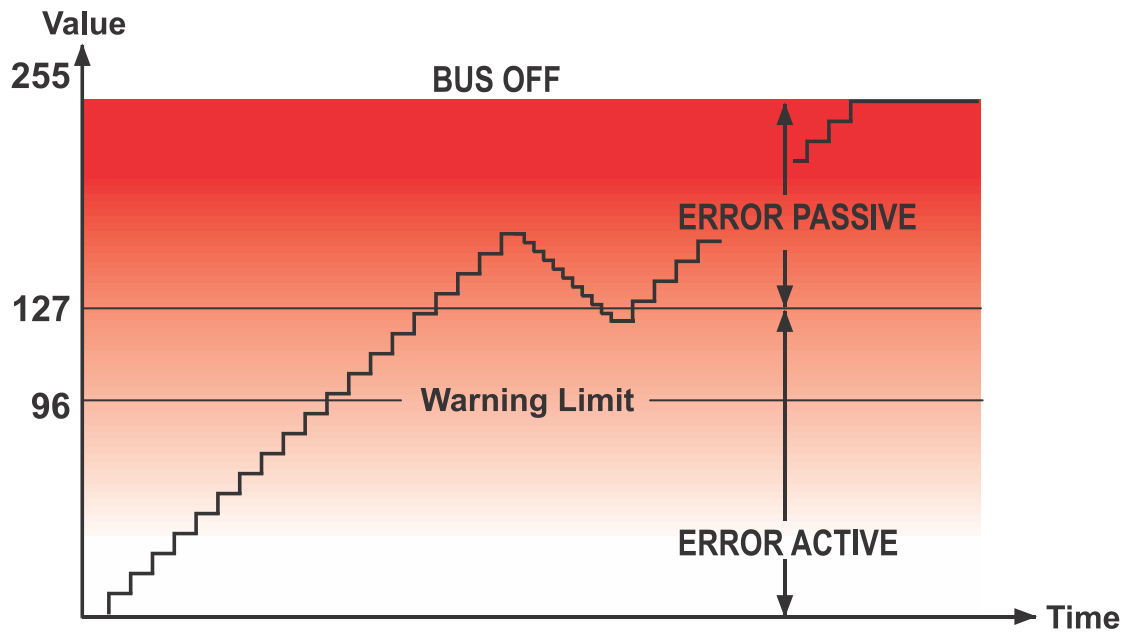
Bit	Meaning	Remarks	
8-15	CAN transmit error counter (TEC)	The Tx error counter is increased when the transmitter channel detects an error frame. When the counter reaches 96, the warning bit is set (bit 16). When the counter reaches 127, the drive is in Error Passive state (bit 17) and does not produce any more Error Frames. When the counter reaches 255, the drive enters Bus Off state	
16	EW, Warning status	0:	Values of both error counters (CANREC and CANTEC) are less than 96.
		1:	One of the two error counters (CANREC or CANTEC) has reached the warning level of 96.
17	EP, Error-passive state	0:	The CAN module is in error-active mode.
		1:	The CAN module is in error-passive mode. CANTEC has reached 128.
18	BO, Bus-off status. The CAN module is in bus-off state.	0:	Normal operation
		1:	There is an abnormal rate of errors on the CAN bus. This condition occurs when the transmit error counter (CANTEC) has reached the limit of 256. During Bus Off, no messages can be received or transmitted. The bus-off state can be exited by clearing the CCR bit in the CANMC register or if the Auto Bus On (ABO) (CANMC.7) bit is set and after 128 * 11 receive bits have been received. After leaving the Bus Off, the error counters are cleared.
19	ACKE, Acknowledge error.	0:	All messages have been correctly acknowledged.
		1:	The CAN module received no acknowledge.
20	SE, Stuff error.	0:	No stuff bit error occurred.
		1:	A stuff bit error occurred.



Bit	Meaning	Remarks	
21	CRCE, CRC error.	0:	The CAN module never received a wrong CRC.
		1:	The CAN module received a wrong CRC.
22	SA1, Stuck at dominant error	The SA1 bit is always at 1 after a hardware reset, a software reset, or a <i>Bus-Off</i> condition. This bit is cleared when a recessive bit is detected on the bus.	
		0:	The CAN module detected a recessive bit.
		1:	The CAN module never detected a recessive bit.
23	BE, Bit error flag	0:	No bit error detected.
		1:	The received bit does not match the transmitted bit outside of the arbitration field or during transmission of the arbitration field, a dominant bit was sent but a recessive bit was received.
24	FE, Form error flag	0:	No form error detected; the CAN module was able to send and receive correctly.
		1:	A form error occurred on the bus. This means that one or more of the fixed-form bit fields had the wrong level on the bus.



The following graph displays the method that the Frame error detection is counted:



G-DS301003A

References



DL – (Binary) Download

DL initiates binary download.

Note: The **DL** command works with binary content, and the data is beyond the scope of this document.

CANopen/CoE

Attributes

Attribute	Description
Type	Command, Write
Source	USB, FoE, TCP (RS-232 & CANopen TBD)
Restrictions	• The motor must be off. • The user program must not be running. • Wizard mode must not be active.
Range	No
Default	No
Unit modes	All
Non-volatile	No

Remarks

DL is used to download data to the drive. It uses a binary format to reduce communication load and time.

The **DL** command consists of three parts:

- Initiation packet
- Body packet
- Termination packet

Please refer to the Software Manual for more details.

The **DL** command is a complicated procedure, which is normally handled by the upper host. Elmo EAS uses this command to download firmware, database image files and user programs.

The **DL** command starts binary interpretation, during which there will be no reply over any command.

In case of an error in the procedure, the drive will automatically to the ASCII interpreter after 3 seconds of no communication.



The downloaded data is loaded to the flash memory. For this reason, the procedure might require a long time for the burning and validation of the data.

References



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 internal units, namely, counts/sec and counts.

CANopen/CoE

Elmo_parameters_objects: **0x3063**

Attributes

Attribute	Description
Type	See the table below.
Source	All
Restrictions	None
Range	See the table below.
Index range	1 to 18
Default	0
Unit modes	All with respect to the relevant control loop
Non-volatile	No

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	Type	Values	Notes
Input number	Command description	Similar to the Type attribute	Range or any other value description as applicable to the command	Similar to the Restriction attribute
1	Current command	Float	-MC to +MC [amperes]	Actual current reference to the current controller, including all



Index	Description	Type	Values	Notes
2	Velocity command	Long	-2e9 to +2e9 [counts/sec]	Actual speed reference to the speed controller, including the output of the position controller
3	Position command	Long	-2e9 to +2e9 [counts]	Actual position reference to the position controller
4	Velocity command only from the socket reference	Long	-2e9 to +2e9 [counts]	The socket reference portion of the speed command
5	Software controlled velocity command	Long	-2e9 to +2e9 [counts/sec]	The software controlled portion of the speed command to the controller
6	Position command only from the socket reference	Long	-2e9 to +2e9 [counts]	The socket reference portion of the position command
7	Software controlled position command	Long	-2e9 to +2e9 [counts]	The software controlled portion of the position command to the controller
8	Stepper angle reference	Long	0 to 511	The calculated angle to the current loop when stepper mode is used
12	ECAM table input	Long	-2^{31} to $(2^{31}-1)$ [counts]	The input to the ECAM table (after ratio)

References



EA[N] – Feedback Emulation Parameters

EA[N] enables the configuration and activation of feedback emulation.

CANopen/CoE

Elmo_parameters_objects: **0x3068**

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
Attribute	None

Remarks

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

1. Quadrature A/B wave format
2. Pulse/Direction wave format
3. Up/Down wave format
4. Halls signals
5. PWM (100% & direction or 50% modes)

Emulation is supported by specific drive GCON-based Rev E (See **WS[8]** command) hardware only.

The emulation mode requires hardware routing to PORT C. This is done via the **GO[]** command. Function 5 is used for the emulation (e.g. **GO[14]=5** for Port C channel A). Refer to the **GO[]** 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 socket ID 13 or 14 (refer to **CA[41]** to



CA[44]). Where **EA[8]** is 0 and internal feedback is used, there is no need to select a socket for this purpose.

If the follower error between the emulated socket and the emulation position is greater than $\pm 1,000,000,000$, 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% & 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. It should be noted that the “Positive” and “Negative” can be determined via **OL[14]** or **OL[15]** logic bits indicating that the PWM direction can be logically inversed regardless of the input 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:

1. If the motor is enabled, the emulation automatically starts. Note that the emulation will start $\sim 100\mu\text{Sec}$ after the servo is enabled (**SO=1**).
2. If the motor is disabled, the emulation is automatically suspended, and resulting in 0 PWM (0 PWM is the minimum value which can be set via **EA[16]**).

The emulation function is supported from PAL version 8 and above. Refer to the **VP** command.

Halls emulation output is supported from PAL version 12 and above. Refer to the **VP** command.

SCORE feedback emulation is not supported.



Indices

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

Index	Description		Type	Default	Restrictions
1	Value	Emulation Output	Long	0	
	134	Read Only PWM mode 50% is suspended due to motor disabled. PWM out is 50%.			
	133 (0x85)	Read Only PWM mode P&D is suspended due to motor disabled. PWM out is 0%			
	-1	Read-only Error cannot follow feedback rate			
	0	Emulation disable			
	1	Quadrature wave signals			
	2	Pulse/Direction wave signals			
	3	Up/Down wave signals			
	4	Hall signals			
	5	PWM mode 100% & Direction (P&D)			
	6	PWM mode 50%			
2	Emulation pulse width N in up/down or pulse/direction waves $N = 2$ to 75 pulse width is $N * 13.3$ [nsec] $N = 76$ to 202 pulse width is $(N - 75 + 1) * 1.04$ [μsec]		Unsigned Short	2	2 to 202
3	Bit	Description	Long	0	
	0	Emulation direction 0: Direction similar to the emulated encoder 1: Direction of emulated output is inversed			
	1	For PWM modes only, the emulation is bound to Servo Enable (SO command) 0: Emulation is not bind to SO 1: Emulation binds to SO . When SO =0 the emulation is set to result 0 output.			When SO =0: In 100% duty cycles is set to 0%. In 50%, duty cycle is set to 50%.



Index	Description	Type	Default	Restrictions
4	Emulated socket number 1 to 4 For PWM emulation modes EA[1] =5/6: 0: The emulated socket for velocity is determined by CA[46] . (VX)	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 ID 13 or 14. Applicable only to EA[8] !=0	Long	0	Read-only
6	Emulation multiplier N = 1 to 2147483647 defines the emulation multiplier. The number of emulated encoder pulses and, as a result, the velocity value will be multiplied by N.	Unsigned Long	1	
7	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 .	Long	0	
8	Not relevant for PWM emulation modes.		0	
	Value			
	0	Use internal emulation feedback		
	1	Use AqB socket for emulation feedback		
9	Emulation Source selector for PWM mode only		1	
	Value	Description		
	1	Velocity Command, The input of the velocity controller		
	2	Velocity according to EA[4]		
	3	Reserved		
	4	Current Command. The input of the current controller		
	5	Current feedback		
	6	Current Socket		
	7	User manual. Duty is set by TW[70] command		
10	Select the PWM frequency. See table below	Long	5	3KHz



Index	Description	Type	Default	Restrictions
11	Reserved			
12	Speed range minimum value. The PWM will be saturated to “Min PWM” (EA[16]) when Speed (or Speed command) is lower than this value.	Long	-VH[2]	Units: User Units / sec
13	Speed range maximum value. The PWM will be saturated to “Max PWM” (EA[17]) when Speed (or Speed Command) is higher from this value.	Long	+VH[2]	Units: User Units / sec
14	Current range minimum value. 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. 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 %

Frequencies

The following table describes the optional PWM frequencies. The resolution of the PWM output depends on the selected frequency.

EA[10]Value	Frequency (Hz)	Resolution	Resolution (bits)
1	1000	37500.00	15.19
2	1500	50000.00	15.61
3	2000	37500.00	15.19
4	2500	30000.00	14.87
5	3000	25000.00	14.61
6	3500	21428.00	14.39
7	4000	18750.00	14.19
8	4500	16666.00	14.02
9	5000	30000.00	14.87
10	5500	27272.00	14.74
11	6000	25000.00	14.61
12	6500	23076.00	14.49



EA[10]Value	Frequency (Hz)	Resolution	Resolution (bits)
13	7000	21428.00	14.39
14	7500	20000.00	14.29
15	8000	18750.00	14.19
16	8500	17647.00	14.11
17	9000	16666.00	14.02
18	9500	15789.00	13.95
19	10,000	15000.00	13.87
20	12,000	12500.00	13.61
21	14,000	10714.00	13.39
22	16,000	9375.00	13.19
23	18,000	8333.00	13.02
24	20,000	7500.00	12.87
25	30,000	5000.00	12.29
26	40,000	3750.00	11.87
27	50,000	3000.00	11.55
28	60,000	2500.00	11.29
29	70,000	2142.00	11.06
30	80,000	1875.00	10.87
31	90,000	1665.00	10.70
32	100,000	1500.00	10.55
33	200,000	750.00	9.55

References

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



EC – Error Code

EC specifies the interpreter and communication error code.

CANopen/CoE

Elmo_parameters_objects: **0x306A**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All, except the user program
Restrictions	None
Range	0 to 255
Default	0
Unit modes	All
Non-volatile	None

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,

WW=10 ;

The command does not exist and will generate the error code **19?**;

Here 3 is the number of the error code. 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** is not updated in cases in which the command is completed with no error.

EC = 0 clears the **EC**.



Error Codes

The following table details the command. In some examples reserved commands are used.

Error Code	Error String	Description / Example / Remedy
1	Do not Update	Reserved
2	Bad Command	Non-existent command This error occurs when an entry that refers to a non-existent command is used. Examples: PG The command PG does not exist. GM[2] The command GM does not exist. 5+14+GT The command GT does not exist.
3	Bad Index	Index of array does not exist, for example: IL[7] generates an error, because the index range is from 1 to 6.
4	PAL does not support this sensor	This PAL version does not support this type of sensor
7	Mode cannot be started - bad initialization data	This error is returned when preset values of a function introduce a conflict. For example, there may be a conflict between the first index in the PVT table (PV) and the available write pointer (MP[6]) when PVT motion begins.
9	CAN message hardware buffer was overrun	CAN message was lost.
10	Cannot be used by PDO	CAN cannot be used by PDO.
11	Cannot write to flash memory	Most probably a hardware problem
13	Cannot reset communication - UART is busy	Cannot reset RS232 communication since the lines are busy. For example: while uploading Recorder data, (while BH command is in process).
16	Array '[]' is expected, or empty expression in array	The command attribute is array. '[]' is expected.



Error Code	Error String	Description / Example / Remedy
17	Format of UL command is not valid - check the command definition	The format of the UL command is not valid. Check the command definition. Example: <code>UL=0x10480000</code> The MSB digit is 1 instead of 0 (it must always be 0). This command should be <code>UL=0x00480000</code> Refer to the UL command.
19	Command syntax error	This error indicates that an incorrect syntax was used or that an incorrect use of legal command was performed. For example: Calling a non-parameter command: <code>SR=100 ;</code> Syntax error: <code>200 = AC ; // expression on the left-hand side</code> Calling a non-indexed command: <code>SD[5]; // SD is a scalar command</code> <code>CA=3 ; // CA[] is a vector command</code>
20	Bad Set Point sending order	Relevant for Interpolated Position Mode, sub mode -1
21	Operand Out of Range	FS=2,000,000,100 returns this error because the required speed is beyond the limits of the drive.
23	Command cannot be assigned	The command cannot be assigned. For example, assigning a digital input as Safety is illegal: <code>IL[1..6]=24</code> or <code>IL[1..6]=25</code>
26	Profiler mode not supported in this unit mode (UM)	This error indicates that the requested profile cannot be performed with the present control loop. For example, in Torque Control (UM = 1) the user requests the position relative command (RM = x).
27	Bad ECAM setting	ECAM setting by EM[] command is not correct, refer to EE[6] for details



Error Code	Error String	Description / Example / Remedy
28	Out Of Limit Range	The value entered in a parameter is outside of the range declared by another related parameter. Example: XM[1] = -20000 XM[2] = 20000 MO = 1 PA = 30000 The value of PA is outside of the range for XM[1]/XM[2] .
33	Bad sensor setting	A bad sensor setting was made during Set Motor Enable (MO = 1). CA[18] and CA[19] are zero when the mode is other than stepping mode (UM = 3).
34	There is a conflict with another command	There is a conflict with another command. Cases: Set the OB command.
35	Max bus voltage (BV) or max current (MC) is not valid	Motor cannot be started because BV or MC defined by DI procedure is not suitable for Drive Info Table
36	Commutation method (CA[17]) or commutation table does not fit to sensor	This error occurs while attempting MO = 1 when the commutation method or the commutation table does not fit the sensor. Example: CA[20] = 0, i.e., no Halls are present. CA[17] = 1, i.e., commutation by Halls.
37	Two Or More Hall sensors are defined to the same place	Hall sensors are defined to the same place. Check CA[4] , CA[5] or CA[6]
38	PORT C setting is incorrect	PORT C mux is with conflict with other GO[] or cannot be assigned to the output
40	In Wizard Experiment!	In Wizard Experiment!
42	No Such Label	Example: XQ##FOO will return this error if neither a label ##FOO nor a function with the name FOO exists in the user program.
45	An Attempt to read a write only command	An Attempt to read a write only command



Error Code	Error String	Description / Example / Remedy
47	Program does not exist or not Compiled	An attempt to run (XQ command) a program when the program has not been compiled (CC command) will generate this error.
48	Motor could not start - fault reason in CD	The motor could not start. The reason for the fault can be retrieved from CD . For example: Trying to set MO = 1 while: there is a motor bus undervoltage, there is a safety input fault, there is a motor short status, there is an over-temperature situation etc.
51	Inhibit OR Abort inputs are active, Cannot start motor	Inhibit or abort inputs are active. The motor cannot be started.
54	Bad Data Base	An attempt to save post processing parameters failed because of an incorrect parameter value. Or User program download failed (wrong symbol table). Try to re-compile the program and download.
57	Motor Must be Off	This error is caused by trying to set parameters which are not allowed to be set under Motor-On, that is, while the motor is on. Example: Set TS (TS = <i>n</i>) while MO = 1.
58	Motor Must be On	PA = 1000 generates an error if MO = 0. The absolute position reference is automatically set to the present position when MO = 1; therefore, when MO = 0, setting PA is pointless.
60	Bad Unit Mode	The reference command is not suitable for the unit mode. For example: If UM = 2 (velocity loop) PA = 1000 (point-2-point motion) generates this error.



Error Code	Error String	Description / Example / Remedy
61	Data Base Reset	This error may occur after upgrading the drive version if the newer version uses a different database structure. Try to set the correct user parameters. Then save (SV command). The SV command will check the new database, save it or return errors.
62	Socket change not allowed	Socket change is not allowed while capture is enabled
67	Recorder Is Busy	This error appears when an attempt is made to change the recorder configuration or to set a new recording request while the recorder is busy. Let the recorder complete its job.
68	Required profiler mode is not supported	Required profiler mode is not supported
69	Recorder Usage Error	Example: RC=2 (record second vector only); RR=2 and later... BH=1 (bring first vector only) is an error, because an attempt is made to bring a vector that was not recorded.
70	Recorder data Invalid	Recorder settings (such as RC=n) have been changed since the last records were made or the recorder has not been operated at all since power-up.
71	Homing is busy	This error is caused by incorrect activation of homing. It also can be that a shared buffer is used, preventing the recorder to be activated. For example: Trying to use data recording while homing is using the recorder buffer (HM[11] = 5 or HF[11] = 5) Or Setting HM[1] while the DS-402 homing mode is activated or the DS-402 touch probe is on.



Error Code	Error String	Description / Example / Remedy
74	Bad profile database, see 0x2081 for object number (EE[2])	During the initiation of the profiler, the database is tested for conflicts in the parameters. If one of the parameters conflicts with another, this error will be presented. The problematic parameter (object) can be retrieved by EE[2] or CANopen object 0x2081. For example: <pre>MO=0 UM=5 XM[1]= -1000000 XM[2]= 1000000 VL[3]= -1500000 VH[3]= 900000 MO=1</pre> EC returns this error. EE[2] returns 24701 (0x607D).
75	Download is in progress	Download is in progress
76	Error mapping is not allowed	Error mapping is not allowed while Homing is in progress (HM[1]\HF[1])
78	Out of Program Range	Error: A virtual machine had to jump into an out-of-code-segment area. The amount of memory allocated for the user program is stated in the drive's User Manual.
79	Sensor setting error	Sensor setting error, possible conflict with other socket settings
81	Download failed see specific error in EE[3]	Download failed. See the specific error in EE[3] .
82	Program Is Running	Wait until the program finishes, or use the HP command or KL command to stop the program.
83	Command is not permitted in a program.	A command which was used inside a user program is not allowed to be used within a program. For example: The next expression XQ##START inside a user program is an error because this command (XQ) is a NON-PROGRAM command.
85	STO is not active	STO is not active or STO diagnostic failed



Error Code	Error String	Description / Example / Remedy
87	Hall sensed with illegal value	Hall sensed with illegal value, may be not connected
94	Not allowed while Error mapping	Not allowed while Error mapping (PC[1]) is in progress
97	RS232 receive buffer overflow	RS232 received a buffer overflow. Characters arrived through RS-232 at too high a rate, causing the internal storage to exceed its capacity. No more space is left to store new characters.
98	Cannot measure current offsets	Current offsets cannot be measured.
99	The sensor does not support this command	The sensor does not support this command
100	The requested PWM value is not supported	The PWM frequency that was requested cannot be used with the drive. This error means that XP[2] cannot be set because MC (the maximum current) and BV (the maximum bus voltage) parameters were not identified (a default was set). Contact Technical Support.
101	Absolute encoder setting problem	When encoder temperature (TI[3]) is read, this error will be generated if the encoder is not configured for one of the supported modes (26 bits or 32 bits). See encoder configurations: CA[59] = 26 or 32 (bits) CA[41] = 5 (BIS mode) CA[60] = 1 (temperature readable)
102	Output Compare is busy	Output Compare (OC[1] \ OC[21]) or Emulation (EA[1]) are busy
103	Output Compare Sensor Is Not QUAD Encoder	Output Compare Sensor Is Not QUAD Encoder
104	Output Compare Table Length OR Data	Output compare table length or data are not valid
105	Speed loop KP out of range	The speed loop KP[2] or KI[2] is out-of-range, i.e., has a negative value. Example: KP[2] = -2.5 Or KI[2] = -44.09



Error Code	Error String	Description / Example / Remedy
107	Encoder emulation parameter is out of range	Encoder emulation parameter (EA[2] to EA[7]) is out of range
108	Encoder emulation in progress	Encoder emulation (EA[1]) is already in progress
110	Too long number	The number is too long. The number of digits before or after the decimal point exceeds the limit of 20 digits.
122	Motion mode is not supported or with initialization conflict	The motion mode is not supported or is in a conflict during the initiation. This can be caused, for example, upon switching from DS-402 motion modes into Elmo's motion modes without adjusting the motion mode parameters.
123	Profiler queue is full	This error indicates that the point-to-point position buffers are full. The user is required to wait until the first position reaches "target reached" or flush the buffer according to the DS-402 Profile Position mode.
125	Personality not loaded	There is a problem in the non-volatile flash memory (firmware image). Try to reload the firmware.
126	User Program failed - variable out of program size	A variable is outside of the program area. Try to reload the program. Try to reduce the number of variables.
128	Bad variable index in database	This error occurs when an attempt is made to map user program (global) variables to the recorder. It is performed using the DB##RV[N] command. <i>N</i> must be in the range from 0 to 7. Using a value of <i>N</i> outside of this range ($N < 0$, $N > 7$) will cause this error. In case the recorder variable is an array, for example, VarArr[M]: DB##RV[N]=VarArr[M] If <i>M</i> is greater than the maximum array index, this error occurs.
129	Variable is not an array	When a variable is mapped into the recorder variables list (the DB##RV[N] command), this error occurs if an attempt is made to map a scalar variable as an array. Example:



Error Code	Error String	Description / Example / Remedy
		DB##RV[1]=MyVar[3] MyVar is not an array.
130	Variable name does not exist	When a variable is mapped into the recorder variables list (the DB##RV[N] command), this error occurs if the name of the variable does not exist (the wrong name is given). Example: DB##RV[1]= MyVar MyVar is an incorrect name.
131	Cannot record local variable	This error occurs when an attempt is made to map a user local variable. Only global variables are allowed. Example: DB##RV[1]=LocVar where LocVar is a local variable.
132	Variable is an array	This error occurs when a user variable is mapped to the recorder and the variable is an array, but the brackets and index are missing. Example: DB##RV[1]=VarName VarName is an array. It should be as below: DB##RV[1]=VarName[2]
133	Number of function input arguments is not as expected	This error occurs when a user function defined with arguments is called through the XQ command with no input arguments or with a number of arguments that does not conform to the function definition. Example: The program defines a function: <pre>function main(int a)</pre> Calling this function as <pre>XQ##main</pre> will return this error.



Error Code	Error String	Description / Example / Remedy
134	Cannot run local label/function with the XQ command	A local label cannot be run with the XQ command. For example: <code>XQ##START</code> When START is defined in the user program inside a user function , it is considered to be a local label, and therefore it is illegal to use it with the XQ command.
135	Frequency identification failed	Frequency identification failed.
136	Not a number	Not a number. Float overflow was detected (number $\geq 1e^{37}$). Example: <code>UF[1]=5e³⁹</code>
138	Position Interpolation buffer underflow	Position Interpolation buffer underflow
139	The number of break points exceeds maximal number	The number of breakpoints exceeds the maximum number. The maximum number of breakpoints in the Gold line is 6. The command to add breakpoint is: <code>DB##BP=ProgramLine</code>
140	An attempt to set/clear break point at the not relevant line	An attempt was made to set/clear a breakpoint in a non-relevant line. For every line of the text program, there is a corresponding line of compiled code. This error appears during an attempt to set a breakpoint at a non-corresponding line of compiled code. Example: <code>DB##BP=653</code> If the actual Drive breakpoint was set at a different code line, this error appears.
142	Checksum of data is not correct	Checksum of data is not correct. An internal error occurred during download of boot identity parameters.
144	Numeric Stack underflow	Numeric stack underflow. An attempt was made to retrieve an entry from an empty stack.



Error Code	Error String	Description / Example / Remedy
145	Numeric stack overflow	Numeric stack overflow. An attempt was made to push a value to the numeric stack when it is full. The user program contains very complex code requiring more stack space than is available. It may also be that there are too many subroutines called. An expression in the command line of the interpreter is too complex; it calls too many functions, causing the numeric stack to overflow.
147	Executable command within math expression	An executable command was inserted within a mathematical expression. An attempt was made to assign an executable command.
148	Nothing in the expression	There is nothing in the expression. An attempt was made to evaluate an empty expression. Example: AC=; This is wrong, because the assigned value is missing.
151	Parentheses mismatch	Parentheses mismatch There is a mismatch between opening and closing parentheses. This error pertains to both parentheses and brackets. Example: sin(2; This is wrong because a closing parenthesis is absent.
152	Bad operand type	Bad operand type There is a mismatch between the actual value type and the expected value type. An internal compiler error occurred due to a mismatch between an operand type and its addressing mode. Contact Technical Support.



Error Code	Error String	Description / Example / Remedy
153	Overflow in a numeric operator	Overflow in a numeric operator: Example: <pre>UI[1]= 200000000</pre> <pre>UI[1]=UI*2000 //It generates this error.</pre> Division by zero: <pre>UI[1]/0</pre> Remainder calculation by zero. <pre>UI[1]%0</pre>
154	Address is out of data memory segment	Address is out of data memory segment. The address of a variable in the data segment exceeds the data segment size. This internal compiler error is caused by corrupted compiled code. In such a case, email Technical Support for assistance. Attach the Composer date and version (in the Help menu) and the program you attempted to compile.
155	Beyond stack range	Compiled code contains a pointer to a stack entry that exceeds the actual stack range (STACK_IMMEDIATELY addressing method). This internal compiler error is caused by corrupted compiled code. In such a case, email Technical Support for assistance. Attach the Composer date and version (in the Help menu) and the program you tried to compile.
156	Bad op-code	Compiled code contains mismatched addressing mode. This internal compiler error is caused by corrupted compiled code. In this case, email Technical Support for assistance. Attach Composer date and version (in Help menu) and the program you tried to compile.
158	Out of flash memory range	Failure in download program procedure For example: The size of the program function table is greater than the maximum allowed size. Try to reduce number of functions in the program.



Error Code	Error String	Description / Example / Remedy
159	Flash memory verification error	Failure in download process: Checksum does not match. Possible hardware problem. Contact Technical Support.
161	Program is not halted	An attempt was made to execute a command that requires the user program to be halted. For example: Activation of the XC command while the virtual machine is not in the halted state.
163	Not enough space in program data segment	There is not enough space in the program data segment. Try to reduce variable usage in the user program.
165	An attempt to access flash memory while busy	An attempt was made to access serial flash memory while busy. Failure occurred on reading serial flash memory. This might be due to a hardware problem. Contact Technical Support.
166	Out Of Modulo Range	This error occurs when the main encoder (the PX command) is set to an out-of-range value: XM[1] is the lower limit XM[2] is the upper limit XM[1] < PX < XM[2] (See CANOpen objects 0x607B.2 and 0x607B.1).
168	Speed too large to start motor	MO = 1 or the motor was started with the Enable switch while the motor was rotating too rapidly.
169	Time out using peripheral.(overflow or busy)	This error occurs in cases in which CPU peripherals are used from two resources. For example: The UL or PK uploading command failed because the transmitter buffer is full. The OP command is not accepted because requests to update the output port arrived from two channels (CANopen & USB) simultaneously.
170	Cannot erase sector in flash memory	The serial flash memory cannot be erased. This problem may occur during downloading of the firmware or parameters. Contact Technical Support.



Error Code	Error String	Description / Example / Remedy
171	Cannot read from flash memory	The serial flash memory (SFlash) cannot be read because an attempt was made to read an illegal area in the SFlash or an internal board communication failure occurred. Contact Technical Support.
172	Cannot write to flash memory	Writing to the serial flash memory (SFlash) failed because an attempt was made to write to an illegal area in the SFlash or an internal board communication or SFlash failure occurred. Contact Technical Support.
173	Executable area of program is too large	This error is detected during the CC command. The user program (the entire code image, excluding text) is too large. Try to reduce the size of the user program code.
174	Program has not been loaded	This error is detected during the CC command. The user program image is illegal or does not exist. Try to download program again.
175	Cannot write program checksum - clear program (CP)	An incorrect checksum value was set in the CC command after the user program was downloaded. The checksum is calculated by the drive, and a failure occurred during the downloading session. Try to query the CC value and set it to the value calculated by the drive.
176	User code, variables and functions are too large	The user program non-text area (code, variables and function table) is oversized. Try to reduce program code.
177	Capture/Compare conversion error in analog encoder. Require known quad location	Capture/Compare conversion error in analog encoder. Require known quad location
178	CAN bus off	CAN bus is off
179	Consumer HB event	Consumer HB event. Consumer HB time is exceeded
180	DF is not supported in this communication type	DF is not supported in this communication type
181	Writing to Flash program area, failed	Writing to the serial flash program area during the Set CC command failed. Contact technical support.



Error Code	Error String	Description / Example / Remedy
182	PAL Burn Is In Process or no PAL is burned	PAL burn is in process during: An attempt to set motor on (MO = 1). This action failed. An attempt to burn PAL. The Read PB command. If PB > 0, try again your action. If PB == 0, wait. If PB < 0, try to burn PAL again or contact technical support.
184	Capture option already used by other operation	Only one of the following Capture options can be used at the same time: ELMO Heritage Home Touch Probe (DS-402) DS-402 Home They all use the same capture module. An attempt to use another option without closing the active one will generate this error.
185	This element may be modified only when interpolation is not active	The following elements may be modified only when interpolation is not active: OF[22] (CANopen object 60C0) – Interpolation sub mode select. OV[23 to 24] (CANopen object 60C2) – Interpolated Data Period. OV[25 to 27] (CANopen object 60C4) – Interpolated Buffer Org.
186	Interpolation queue is full	The interpolation queue is full. This might appear while setting object 60C1 – Interpolated data record.
187	Incorrect Interpolation sub-mode	The interpolation sub-mode is not supported.
188	Gantry slave is disabled	Gantry master cannot enabled if gantry slave is not enabled at current mode
189	CAN message was lost, software buffer overflow	CAN message was lost, software buffer overflow
190	Profile Acceleration is out of range	Profile Acceleration is out of range or zero
191	Motor over temperature	Motor over temperature



Error Code	Error String	Description / Example / Remedy
200	Main Feedback error refer to: EE[1]	Main Feedback error refer to: EE[1]
201	Commutation sequence failed	Commutation sequence failed
202	Encoder - Hall sensor mismatch refer to: XP[7]	Encoder - Hall sensor mismatch refer to: XP[7]
203	Current limit was exceeded	Current limit was exceeded
204	External inhibit input detected	External inhibit input detected
205	AC Fail: loss of phase	AC Fail: loss of phase
206	Digital Hall run too fast or disconnected	Digital Hall run too fast or disconnected
207	Speed error limit exceeded refer to: ER[2]	Speed error limit exceeded refer to: ER[2]
208	Position error limit exceeded value of ER[3]	Position error limit exceeded value of ER[3]
209	Cannot start motor because of bad database refer to: CD	Cannot start motor because of bad database refer to: CD
210	Bad ECAM table	Bad ECAM table
216	Cannot find zero position without DHalls	Cannot find zero position without DHalls
217	Over Speed violation refer to: HL[2]	Over Speed violation refer to: HL[2]
221	Motor stuck	Motor stuck
222	Out of position limits refer to: HL[3] or LL[3]	Out of position limits refer to: HL[3] or LL[3]
223	Numerical overflow	Numerical overflow
224	Gantry slave is not enabled	Gantry slave is not enabled
229	Cannot start motor because of internal problem	Cannot start motor because of internal problem
233	Under voltage protection	Under voltage protection
235	Overvoltage protection	Overvoltage protection
237	Safety switch	Safety switch
241	Short protection	Short protection



Error Code	Error String	Description / Example / Remedy
243	Over temperature protection	Over temperature protection
245	Additional Inhibit input	Additional Inhibit input

References

[EE\[\]](#), [MF](#), [SR](#)



EE[N] – Extended Error

EE[N] reports detailed error codes according to a specific feature described below. Object **0x2081** is an alias to the **EE[N]** command

CANopen/CoE

Elmo_parameters_objects: **0x306C**

Attributes

Attribute	Description
Type	Long
Source	RS232, USB, TCP, EoE
Restrictions	Read-only
Range	None
Index range	1 to 6
Default	0
Unit modes	All
Non-volatile	No
Attribute	None

Remarks

Extended error information is used by the following features:

- Feedback Error
- Profiler initialization
- ECAM initialization
- Download procedure
- SDO communication error

In the above cases, if an error occurs, the **EC** or **MF** value, or abort SDO, will indicate an error, and more specific and detailed error information will be available in the relevant **EE[]** command.

An extended error code is valid only after the following scenarios:

1. *Feedback error.* In some encoder types, a feedback error can be detected and reported to the host. The following feedbacks are inspected for a feedback error:
 - a. Absolute serial sensor (BiSS, EnDAT, etc.). Refer to the **EE[1]** table data.



- b. Virtual absolute sensor (Gurley). Refer to the **EE[1]** table data. The Gurley error is identified by the sensor fault **MF=1**, the reason for error can read by **EE[1]** command:



There are actually two types of possible error:

EE[1]=20 (hex)

Amplitude is out of range. Range is define in **CA[48]** (maximum amplitude squared) and **CA[49]** (minimum amplitude squared). Both parameters are in ADC units.

EE[1]=1x (hex)

Data not valid, where x are the four LSB bits above updated in the Gurley read out algorithm

- c. Analog sensor (sine/cosine sensor). Refer to the **EE[1]** table data.

EE[1]=20 (hex)

Amplitude is out of range. Range is define in **CA[48]** (maximum amplitude squared) and **CA[49]** (minimum amplitude squared). Both parameters are in ADC units.

If any of these feedbacks fail, the motor is disabled automatically, and **MF** reports an error (**MF = 1**). In cases where there is more details regarding the reason for the failure, the **EE[1]** command supports these details .

Note that the errors depend on the specific details determined by the encoder manufacturer.

2. *Profiler Initialization*. The **EC** command value is 74, which indicates an error during profiler initialization. Check **EE[2]** for the detailed error.
3. *Download FW procedure fails*. The **EC** command value is 81, which indicates an error during the download procedure. Check **EE[3]** for the detailed error.
4. *SDO communication*. If SDO returns an abort message, which indicates an error while processing the SDO command, **EE[4]** gives more details about the error.
5. *ECAM Initialization*. The **EC** command value is 27, which indicates an error during ECAM initialization. Check **EE[6]** for the detailed error.



Indices

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

Index	Description	Type	Values
1	Feedback error, relevant according to the above description	Long	See the table for EE[1] below.
2	Profiler initialization error, relevant in the case of EC=74	Long	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	Long	Returns a value that indicates the cause of the error. See the table for EE[3] below.
4	When SDO (used in CANopen) returns an abort message, EE[4] provides further details about the error	Long	Returns ELMO error value according to the EC command error list. If the value is -1, no further information will be provided as there was no information available.
5	Faults which are detected by the Motor On procedure (MO=1) are detailed in EE[5] . Relevant in case of EC=48	Long	See the table EE[5] fault below.
6	ECAM initialization error, relevant in the case of EC=27	Long	Returns a value that indicates the cause of the error. See the table for EE[6] below.

Errors

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

Value	Description
Bits 0 to 7	Bits 0 to 7: Capture of the status field (the errors are encoder manufacturer-dependent) in the case of a serial/absolute encoder error. See Table 4.
Bit 8	CRC error
	0 No error
	1 CRC error
Bit 9	Encoder ID OK (Panasonic and Tamagawa only)
	0 ID is OK



	1	ID of the sensor does not fit the sensor
Bit 10	Flag that indicates if no data arrived from the encoder	
	0	OK
	1	No data
Bit 11	Sync or CRC error between Gantry drives (excluding the serial encoder)	
	0	OK
	1	Communication error
Bits 16 to 22	EnDat error message (See Table 4)	

Table 3: EE[1] - Serial Encoder Errors



Encoder	Description		
Biss	Bit 0: Warning	If it equals 0, the encoder scale should be cleaned.	
	Bit 1: Error	A value equal to 0 indicates that the absolute position data may not be valid or that the temperature is above the maximum operating temperature of the encoder.	
General Biss	Bit 0: Error	A value equal to 0 indicates that the absolute position data may not be valid or that the temperature is above the maximum operating temperature of the encoder.	
EnDat	Bit 0: Error	If it equals 1, the internal data check failed.	
	Bits 16 to 22: Error Message A bit equal to 1 signals an error message.	16	Light source
		17	Signal amplitude
		18	Position value
		19	Overvoltage
		20	Undervoltage
		21	Overcurrent
		22	Battery
Panasonic Refer to the Error codes in section 1.1.2 Panasonic Status Flag Function Descriptions To reset error, use TW[20] =socket #	Single- turn		
	Bits 0 to 7 Note: For more information refer to the manufacturer's specification.	0	0
		1	System down
		2	0
		3	0
		4	Counter overflow
		5	Count error
		6	Full Abs status
		7	Overspeed
	Multi-turn		
	Bits 2 and 3	2	One of the following: Multi-turn error, Battery alarm, Full absolute status, counter error, counter overflow, overspeed
		3	System down



Encoder	Description		
Tamagawa To reset error, use TW[20]=socket #	Single-turn		
	Bit 5	5	Count error
	Multi-turn		
	Bits 0 to 7	0	Battery alarm
		1	System down
		2	Multiple revolution error
		3	0
		4	Counter overflow
		5	Count error
		6	Full Abs status
		7	Overspeed
SSI	Bit 0	0	Error bit
Mitutoyo	Bits 0 to 7	0	0
		1	0
		2	0
		3	1
		4	0
		5	Communication error
		6	Encoder alarm
		7	System error
Kawasaki	Bits 0 to 2	0	Busy
		1	Absolute track error
		2	Interpolation error
Yaskawa To reset error use TW[20]=socket # Note: Resetting an error also sets the Multi-turn counter to zero	Bits 0 to 7	0	0
		1	Reset Complete
		2	Overheat
		3	Overspeed
		4	Position Error
		5	Battery Warning
		6	Encoder Error
		7	Backup Alarm



Encoder	Description		
Sanyo/Nikon Refer to the Error codes in section 1.1.1 Sanyo Denki/Nikon Status Flag Function Descriptions To reset error use TW[20]=socket # To reset BA error and multi-turn counter, use TW[19]=socket #	Bits 0 to 3: Encoder Status	0	STErr + PSErr + BA
		1	OvSpd + MemErr + OvTemp
		2	BW
		3	Busy + MemBusy
	Bits 16 to 25: Specific alarm status codes	16	Over Temperature
		17	Memory Busy
		18	Busy
		19	PS Error (Multi turn)
		20	ST Error
		21	MemErr
		22	Overspeed
		23	0
		24	Battery Alarm
		25	Battery Warning
EnDat 2.2 Refer to specific encoder datasheet for detailed description To reset error use TW[20]=socket #	Bits 16 to 31: Operating status error sources	16	Light Source
		17	Signal Amplitude
	For masking purposes when setting CA[8], consider bits 16 – 31 as bits 0 – 15.	18	S Pos1
		19	Over voltage
		20	Under voltage
		21	Over current
		22	Temperature exceeded
		23	S Pos2
		24	S System
		25	S ALL Power down
		26	M Pos1
		27	M Pos2
		28	M System
		29	M ALL Power down
		30	M Overflow/Underflow
		31	M Battery



Encoder	Description		
Gurley	Bit 1: Sequence Error	1	Refer to the Gurley distributor of the encoder for details
	Bit 2: Quadrature Error	2	Refer to the Gurley distributor of the encoder for details
	Bit 3: Reset Out	3	Refer to the Gurley distributor of the encoder for details
	Bit 4: Data Not Valid	4	For details see bits 1 to 3
	Bit 5: Wrong Amplitude	5	Analog signal amplitude is out of range (see CA[48] and CA[49])
Sin/Cos, Resolver, or Hiperface	Bit 5	5	Wrong Amplitude
Panasonic Incremental To reset error use TW[20] =socket #			0 – No Error 1 – Encoder internal initialization error 2 – Count error 1 3 – Count error 2 4 – Z signal error 5 – CS signal error
Serial Exclusive #3	Bit 0	0	Error bit

Table 4: Status Bits of Different Encoders (Bits 0 to 7)

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

EE[2] Value (object)	Description
0x6091	Position ratio is out of range.
0x607C	Homing offset is out of position limits.
0x607B	Minimum position range limit is greater than the maximum position range limit.
0x607D	Software position limits have one of the following errors: • Upper and lower software position limits are equal. • Position software limits and position range limits are ambiguous. Range boundaries must not overlap.

Table 5: EE[2] - Profiler Initialization Errors



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

EE[3] Value (object)	Description
1	Header packet is missing DL
2	Header packet termination code is not 0x1234
3	Received header packet number does not match the expected packet number
4	Header packet size is incorrect
5	Header packet checksum error
6	Header packet type is not legal
7	Body packet termination code is not 0x1234
8	Received body packet number does not match the 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 database post process fail
14	Time-out while waiting for a message (1.5 seconds)
15	Downloaded data is larger than the user parameters reported in DL type 19
16	Error while writing downloaded data to FLASH memory
17	General download aborting error, with no precise error information

Table 6: EE[3] - Download Procedure Errors

The following table lists the faults which occurs during the motor on procedure and reported by **EE[5]**.

EE[5] Value (Object)	Description
1	Main Feedback error. Check feedback connection or settings
16	External Inhibit is triggered. Check digital input which is define in "Inhibit" state
12288 (0x3000)	Drive is in under voltage state. Check power supply .Voltage should be at least WI[38] volts.
20480 (0x5000)	Drive is in over voltage state. Check power supply. Voltage should not exceed WI[36] volts.



28672 (0x7000)	Safety switch is in safety state. Must set the safety switch to active.
45056 (0xB000)	Drive power stage is in "short protection" state.
53248 (0xD000)	Drive power state is over temperature. Check TI[1] for temperature read out in Centigrade.
61440 (0xF000)	Additional External Inhibit is triggered. Check digital input which is define in "Additional Inhibit" state

Table 7: EE[5] – Motor On Fault Errors

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

EE[6] Value (Object)	Description	Details
1	Ratio denominator, EM[10] is out of range.	EM[10] must be > 1.
2	First or last ECAM table index, denoted by EM[5] & EM[2] , is out of range.	Error due to: EM[5] < 1 or EM[5] > 2047 EM[2] < 2 or EM[2] > 2048
3	ECAM table gaps, denoted by EM[4] and EM[7] , is out of range.	Error due to: EM[4] < 1 EM[7] < 0
4	The Velocity or Acceleration FIR filter length used by the ECAM output is out of range.	Error due to: EM[12] < 0 or EM[12] > 8 EM[13] < 0 or EM[13] > 8
5	If Non-constant master gap table (EM[11]) is used, the total length of Master and Slave tables (EM[2] - EM[5] +1), must be even.	(EM[2] - EM[5] +1) must be even number
6	First position entry of the master table (ET[EM[5]]) must be 0.	ET[EM[5]] must be 0
7	If Non-constant master gap table is used (EM[11]), the position in the master table must be monotonic rising.	Error due to: ET[n] >= ET[n+1] where n is the all index range of the Master table.



8	Illegal ECAM mode was selected	Error due to: EM[1] < 0 EM[1] > 2
9	Illegal last segment shortening (EM[7]) value	Error due to: EM[7] >= EM[4]
10	First index of ECAM table is not smaller than last index	Error due to: EM[5] >= EM[2]
11	Attempt to change active segment of ECAM table	When changing ET[N] , table index N must be outside the range EM[8]+/-2 if RM=1 and MO=1

Table 8: EE[6] - ECAM Initialization Errors

References

EC, MF, SR



EI – Initialize External Reference Generator

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

CANopen/CoE

Elmo_parameters_objects: **0x3070**

Attributes

Attribute	Description
Type	N/A
Source	All
Restrictions	N/A
Range	N/A
Default	N/A
Unit modes	UM=2, UM=5
Non-volatile	No

Remarks

EM[N] settings, except for **EM[9]**, go into effect only after applying **EI** command.

References

EM[N]



EM[N] – ECAM / Follower Parameters

EM[N] determines the behavior of the ECAM (Electronic CAM) motions.

CANopen/CoE

Elmo_parameters_objects: **0x3074**

Attributes

Attribute	Description
Type	Long
Source	All
Restrictions	According to array index
Range	According to array index
Index range	1 to 13
Default	According to array index
Unit modes	UM=5 (Position - ECAM and Follower) UM=2 (Velocity – Follower)
Non-volatile	Yes
Attribute	None

Indices

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

Index	Description	Type	Default	Restrictions
1	ECAM mode			
	Value	Long	0	RM=0
	Mode			
	0			
	1			
	Non-periodical (linear) ECAM			
	2			
	Periodical (cyclic) ECAM			
2	Last valid index of ECAM table.	Long	10	2 to 2048
3	Starting position value of the input to the ECAM function, where the ECAM function output is ET[EM[5]] If EM[3] is out of range for PY , the modulo PY is employed.	Long	0	-2^{31} to $(2^{31} - 1)$



Index	Description		Type	Default	Restrictions
4	Master reference (ΔPY) distance (gap) between consecutive points in the ECAM table ET[N] . Effective when the constant master gap table is selected (EM[11]:bit2 = 0).		Long	1	1 to ($2^{31} - 1$)
5	First valid index of the ECAM table.		Long	1	1 to 2047
6	Reserved		Long	0	
7	Last master segment shortening. Last master distance (gap) $\Delta PY = EM[4] - EM[7]$. Effective when constant master gap table is selected (EM[11]:bit2 = 0)		Long	0	0 to ($2^{31} - 1$) EM[7] < EM[4]
8	Reports the present index in the ECAM table. When the ECAM motion is not active, EM[8] reports 0.		Long	0	1 to 2048 Read-only
9	Ratio numerator. Set EM[9] to synchronously activate the new ECAM/Follower ratio = EM[9]/EM[10] .		Long	1	-2^{31} to ($2^{31} - 1$)
10	Ratio denominator. The EM[10] setting becomes effect only after setting EM[9] .		Long	1	1 to ($2^{31} - 1$)
11	ECAM options		Long	0	
	Bit	Description			
	0	Reserved			
	1	Reserved			
	2	ECAM table type			
	0	One table, , fixed (constant) master gap: ET[n] , n=1...2048			
	1	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;			
	3	ECAM input source			
	0	Additional encoder (defined in socket architecture)			
	1	Main Profiler or CAN Encoder			



Index	Description		Type	Default	Restrictions
	4	Interpolation type			
		0	Quadratic interpolation		
		1	Linear interpolation		
	5	Table index search mode			
		0	Incremental		
		1	Direct		
12	ECAM velocity FIR filter length		Long	1	1 to 8
13	ECAM acceleration FIR filter length		Long	1	1 to 8

Notes

- Modifying of ECAM input source (**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** 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 ($(\text{EM}[2]-\text{EM}[5]+1)/2$) 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.



References

[ET\[\]](#), [RM](#), [EI](#)



EO – Echo Off

EO specifies the communication echo mode.

CANopen/CoE

Elmo_parameters_objects: **0x3076**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	USB, RS232
Restrictions	None
Range	0, 1
Default	0
Unit modes	All
Non-volatile	Yes

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 follower tracking error of the relevant control loop.

CANopen/CoE

Elmo_parameters_objects: **0x3079**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	Refer to the note below.
Range	See the table below.
Index range	2 to 3
Default	ER[2] = 1000000000 ER[3] = 1000000000 ER[5] = 1000000000
Unit modes	ER[2] for the velocity loop under UM = 2 and UM = 5 ER[3] for the position loop under UM = 5 ER[5] for position loop in gantry setting under UM =5
Non-volatile	Yes
Activation	Immediate

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
0	Reserved			
1	Reserved			
2	The maximum allowed velocity error Its value is abs(DV[2] - VX)		0 to 2000000000 [counts/sec]	100000000 [counts/sec]



Index	Description	Type	Values	Default
	in counts/second			
3	The maximum allowed position error in counts Its value is $\text{abs}(\text{DV}[3]-\text{PX})$ in counts		0 to 1000000000 [counts]	1000000000 [counts]
5	The maximum allowed position error in counts, of the Yaw position in gantry or in stepper closed loop. Yaw position is the difference between master and slave position.		0 to 1000000000 [counts]	1000000000 [counts]

References

MF, **MO**, **SR**, **DV[]**, **VX**, **PX**



ET[N] – ECAM table

ET[N] specifies the Long array which can be used as general-purpose non-volatile memory. Currently it is used by the following drive algorithms: ECAM, error mapping, and output compare.

CANopen/CoE

Elmo_parameters_objects: **0x307B**

Attributes

Attribute	Description
Type	Long
Source	All
Restriction	None
Range	-2^{31} to $(2^{31} - 1)$
Default	None
Index range	1 to 2048
Unit modes	All
Non-volatile	Yes

Remarks

- When the motor is enabled (**MO**=1) and ECAM table is running (**EM**[1]=1 or **EM**[1]=2), the ECAM table entries can be changed "on the fly", except for **ET**[**EM**[8]-2]...**ET**[**EM**[8]+2] entries

References

EM[N],PC[N]



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_parameters_objects: **0x3084**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	• $FC[1]*FC[6]*FC[8] < 2^{63}$ • $FC[2]*FC[5]*FC[7] < 2^{63}$
Range	1 to $(2^{31} - 1)$
Index range	1 to 8
Default	1 for all settings
Unit modes	All
Non-volatile	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.

$$\frac{\text{UserUnits}}{\text{EncoderCounts}} = \frac{\text{FeedConstant}}{\text{GearRatio} * \text{PositionEncoderResolution}}$$

Here the *feed constant* is the positional movement for any motor movement and is calculated using the formula

$$\text{FeedConstant} = \frac{\text{Feed}}{\text{DrivingShaftRevolutions}} = \frac{FC[7]}{FC[8]} = \frac{0x6092.1}{0x6092.2}$$

The *gear ratio* is the ratio that defines what a gear adds to the movement and is calculated using the formula



$$\text{GearRatio} = \frac{\text{MotorShaftRevolutions}}{\text{DrivingShaftRevolutions}} = \frac{\text{FC}[5]}{\text{FC}[6]} = \frac{0x6091.1}{0x6091.2}$$

The *position encoder resolution* is the ratio between the motor shaft and the encoder counts:

$$\text{PositionEncoderResolution} = \frac{\text{EncoderCounts}}{\text{MotorShaftRevolution}} = \frac{\text{FC}[1]}{\text{FC}[2]} = \frac{0x608F.1}{0x608F.2}$$

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., **0x6063**).

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*:

$$\text{VelocityValue} = \frac{\text{PositionValue}}{\text{sec}} * \text{VelocityFactor}$$

$$\text{VelocityFactor} = \frac{\text{FC}[9]}{\text{FC}[10]} = \frac{0x6096.1}{0x6096.2}$$

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*:

$$\text{AccelerationValue} = \frac{\text{VelocityValue}}{\text{sec}} * \text{AccelerationFactor}$$

$$\text{AccelerationFactor} = \frac{\text{FC}[11]}{\text{FC}[12]} = \frac{0x6097.1}{0x6097.2}$$

The velocity and acceleration factors must be used when the required acceleration or velocity (**AC**, **DC**, **SP** etc.) exceeds the maximum value (>2e9), as in the following examples:

1. The required velocity is 5e9, and the acceleration is 10e9: **FC[9]=1;FC[10]=10;**
FC[11]=1;FC[12]=1; SP=5e8; AC=DC=1e9;
2. The required velocity is 1e9, and the acceleration is 10e9: **FC[9]=1;FC[10]=1;**
FC[11]=1;FC[12]=10; SP=1e9; AC=DC=1e9.

Indices

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

Index	Description	Type	Values	Note
0	Reserved			
1	Position encoder resolution numerator	Long	1 to (2 ³¹ – 1)	0x608F.1
2	Position encoder resolution denominator	Long	1 to (2 ³¹ – 1)	0x608F.2
3	Velocity encoder resolution	Long	1 to (2 ³¹ – 1)	0x6090.1



Index	Description	Type	Values	Note
	numerator			(reserved)
4	Velocity encoder resolution denominator	Long	1 to $(2^{31} - 1)$	0x6090.2 (reserved)
5	Gear ratio numerator	Long	1 to $(2^{31} - 1)$	0x6091.1
6	Gear ratio denominator	Long	1 to $(2^{31} - 1)$	0x6091.2
7	Feed constant numerator	Long	1 to $(2^{31} - 1)$	0x6092.1
8	Feed constant denominator	Long	1 to $(2^{31} - 1)$	0x6092.2
9	Velocity factor numerator	Long	1 to $(2^{31} - 1)$	0x6096.1
10	Velocity factor denominator	Long	1 to $(2^{31} - 1)$	0x6096.2
11	Acceleration factor numerator	Long	1 to $(2^{31} - 1)$	0x6097.1
12	Acceleration factor denominator	Long	1 to $(2^{31} - 1)$	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_parameters_objects: **0x3087**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	See the table below.
Index Range	1 to 6
Default	See the table below.
Unit modes	See the table below.
Non-volatile	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
0	Reserved			
1	Specifies how much of the second derivative of the position reference (or the first derivative of the velocity reference) is fed as a reference to the current controller. For open and closed loop stepper mode (UM=6) specifies the factor of additional gain proportional to acceleration. In amperes per acceleration (counts per second²).	0	0 to 2000	In position and velocity modes (See OV[2]) and also in stepper open and closed loop
2	Specifies the factor of velocity feed forward added to the position controller output in the velocity command.	1	0 to 1	In position modes (See OV[2])
3	Specifies the ratio between the velocity sensor resolution and the position sensor resolution in dual loop. The socket number for the velocity sensor is the value of CA[46]. The socket number for the position sensor is the value of CA[45].	1	≥ 0	In position modes (See OV[2])
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 outputted) 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) specifies additional current proportional to speed.	0	0 to 200	In stepper open and closed loop mode
6	Motor constant for Hot Plugging, in Volt (peak) of one phase per Krpm	0	0 to 2000	When this value is zero this function is



				disabled. Set XA[2] according to required speed

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; $\text{FF}[3] = \frac{1000 * 5}{2000} = 2.5$.

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/sec². Then you need to set the

following: $\text{FF}[1] = \frac{1}{1,000,000} = 1.0e-6$

Example 3

To obtain current command proportional to velocity reference in stepper open and closed loop mode (**UM**=6), reset **HT[2]** and **HT[3]** and only set **FF[5]**.

Note: Hot plugging is when the user needs to set the motor ON when the motor is moving. **XA[2]** is the electrical speed [Hz] to enable the motor. To increase this speed above the default value (5 Hz) it is recommended to also set the back EMF of the motor **FF[6]**, so that the first voltage command is suitable for this speed.

References

UM, **CA[]**, **XA[]**



FP[] – Feedback Position

FP[N] specifies the position of the feedback which is associated with socket *N*.

CANopen/CoE

Elmo_parameters_objects: **0x3091**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	The motor must be off.
Range	None
Index Range	1 to 4
Unit modes	All
Non-volatile	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 of the position socket as defined in **CA[45]**.

The main position feedback must be set when the motor is disabled (by the **PX** or **FP** command). In this case it is recommended to use the **PX** command and not the **FP[N]** command, because the **PX** command also synchronizes the profiler and the position actual value (object **0x6064**).

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.



Indices

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

Index	Description	Notes
1	Position of the sensor in socket number 1	Counts
2	Position of the sensor in socket number 2	Counts
3	Position of the sensor in socket number 3	Counts
4	Position of the sensor in socket number 4	Counts

Note: For sensor ID=34 (stepper closed loop) the command FP[N] return position plus output of the controller, together multiplied by KP[6], provides the feedback value.

References

[CA\[\]](#), [PX](#), [FV\[\]](#)



FS – PTP Final Speed

FS specifies the configured velocity, which the drive will have on reaching the target position.

CANopen/CoE

Elmo_parameters_objects: **0x3094**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	Effective on the next call to BG
Range	0 to 2e9
Default	0
Unit modes	UM = 5 & Profile Position mode (0x6041 = 1)
Non-volatile	No

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 bland 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[] – Float Trigger

FT[] specifies the floating point trigger for the recorder.

CANopen/CoE

Elmo_parameters_objects: **0x3095**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	Recorder inactive (RR = 0 or RR = -1)
Range	As in the table below
Index range	1, 2
Default	0
Unit modes	All
Non-volatile	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 slope	Float	Float range	
2	Set for window	Float	Float Range	

References

[RP\[5\]](#), [RP\[6\]](#)



FV[] – Feedback Velocity

FV[N] reads the velocity of the feedback which is associated with socket *N*.

CANopen/CoE

Elmo_parameters_objects: **0x3097**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	For non-active sockets return zero
Range	$-2^{31} \dots 2^{(31-1)}$
Index Range	1 to 4
Unit modes	All
Non-volatile	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 (not user units).

VX is the same as **FV[N]**, where *N* is the socket of the velocity feedback or position feedback, depending on the DS-402 polarity object (object **0x607E** or **OV[14]**).

Indices

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

Index	Description	Notes
1	Velocity of socket number 1	Counts/sec
2	Velocity of socket number 2	Counts/sec
3	Velocity of socket number 3	Counts/sec
4	Velocity of socket number 4	Counts/sec

References

CA[], VX, FP[]



GI[] – Capture Input MUX Selection

GI[] routes digital inputs into the Strobe and Index inputs of quadrature module 0 or 1 of the drive.

CANopen/CoE

Elmo_parameters_objects: **0x30A4**

Attributes

Attribute	Description
Type	Unsigned Long, Read/Write
Source	RS232, USB, TCP, EoE
Restrictions	Socket number (bits 8:15) 1 to 4
Range	See the restrictions entry below.
Index range	1 to 4
Default	Refer to the GI[M] command format table below.
Unit modes	All
Non-volatile	Yes
Attribute	None

Remarks

The Gold drive includes two quadrature encoder modules Quad 0 (Port B) and Quad 1 (Port A). 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 quadrature modules include 2 hardware signals: Index and Strobe. These signals are basically used for homing and position capturing abilities.

The **GI[M]** command enables routing of general purpose digital inputs and sensors (motor index signal) into the Index and Strobe inputs of the drive's quad modules. This allows any position capturing from various types of inputs.

The Strobe of Quad module 0 or/and Quad module 1 can be connected to one of the following for position capturing and homing purposes:

- First motor Index output
- Second motor Index output
- Gold drive general purpose digital Inputs 1 to 6

Note:

Different drives may have different number of available digital inputs. Consult the specific drive's User 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 Quad 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.

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.

Bits 8 to 15, which were previously used to select the socket number, are ignored for compatibility reasons.

Indices

The following table details the **GI[]** options (for the actual values, see the second table below):

Index	Description	Default	Value/Options	
0	Reserved	None	None	
1	Quad Module 0 (Port B) Index routing	0: Port B index is used for capture	For GCON based drives (GWHI, GTRO, GDRU):	
			Value	Description
			0	Quad module 0. Index is routed for the Index entry (GI[1] and GI[3]) or Strobe entry (GI[2] and GI[4])
2	Quad Module 0. Strobe signal routing	7: Input 6 selected as strobe	1	Quad module 1. Index is routed for the Index entry (GI[1] and GI[3]) or Strobe entry (GI[2] and GI[4])
3	Quad Module 1 (Port A). Index routing	1: Port A index is used for capture	2 - 7	Digital inputs 1 to 6 are routed for the Index entry (GI[1] and GI[3]) or Strobe entry (GI[2] and GI[4])
4	Quad Module 1 Configured for: Strobe signal input	6 : Input 5 selected as strobe	For *SCORE based drives (GGUI): GI[1] and GI[3] cannot be modified. Any value will result in 0 without error.	

* Note for SCORE:

Only a strobe entry in **GI[2]** & **GI[4]** can be modified. The default of **GI[2]** & **GI[4]** is 0 where that Strobe is used as capture for the homing mode. This allows normal operation without modifying the **GI[]** value.



GI[M] command format:

Bits	Value Description
0 to 7	Mux input: 0 – Quad-0 (Port B) + index 1 – Quad-1 (Port A)+ index 2 – Input-1 3 – Input-2 4 – Input-3 5 – Input-4 6 – Input-5 7 – Input-6 8 – Input-7 – Not supported in this version 9 – Input-8 – Not supported in this version 10 – Input-9 – Not supported in this version 11 – Input-10 – Not supported in this version 12 – Input-11 – Not supported in this version 13 – Input-12 – Not supported in this version 14 – Input-13 – Not supported in this version 15 – Input-14 – Not supported in this version 16 – Input-15 – Not supported in this version 17 – Input-16 – Not supported in this version
8 to 15	Ignored

References

[HM\[\]](#), [HF\[\]](#), DS-402 Homing Mode, Touch-Probe



GO[] – Digital Output Source

GO[] routes digital output into Quad feedback module output or CPLD output. The routing is determined according to the desired function of the digital output.

Note that when **GO[]** = 0 , the output function is determined via the **OL[]** command.

CANopen/CoE

Elmo_parameters_objects: **0x30AA**

Attributes

Attribute	Description
Type	Unsigned Long, Read/Write
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	GO[1] to GO[4] : 0 to 2, or 7 GO[14] to GO[15] : 0 to 8
Index range	1 to 4 14 to 16 when port C is used as output
Default	Refer to the table below.
Unit modes	All
Non-volatile	Yes
Attribute	None

Remarks

The drive includes two quadrature encoder modules, Port B is alias to Quad 0 and Port A is alias to Quad 1. These modules are used, among other things, to generate pulses according to position reading.

The drive includes port C output that is used (among other things) for feedback emulation, daisy chain and gantry functions.

Each quadrature module includes a Strobe hardware signal. This signal can be used to generate pulses according to a given position. The Strobe is also used for capture & homing purposes.

The **GO[N]** command enables routing between the digital output (denoted by *N*) and the quad module's strobe output. Depending on the requested function, the routing is also performed between the output and the CPLD.



This command allows the user to select the following for each of the supported outputs:

- General-purpose output. Here the output is controlled by the functionality defined in the **OL[N]** command.
- Port B Strobe. Here the Output Compare 1 or Output Compare 2 are activated on Port B (Quad 0).
- Port A Strobe. Here the Output Compare 1 or Output Compare 2 are activated on Port A (Quad 1), and the output is routed to the Strobe pin.
- Daisy chain Port B (Quad 0). Here the output is connected to Quad module 0.
- Daisy chain Port A (Quad 1). Here the output is connected to Quad module 1.
- Emulation. Here the output is connected to the encoder emulation output allowing either of the following emulation signals which are generated from port C according to the emulated feedback (socket):
 - AqB pulses
 - Pulse and Direction
 - Up Down format
 - Digital Halls format

GO[] Indices and Values

All **defaults** values of **GO[]** are 0.

GO[0] is Reserved

GO[1] to GO[4] denotes the functionality of the digital output 1 to 4.

GO[5] to GO[13] are reserved

GO[14] to GO[16] routes the function of Port C to general purpose, emulation, daisy chain, Gantry, STO output or buffering of the absolute sensor.

The following tables describe the values:

GO[1...4] values	Description
0	General-purpose Output: The function of the output is defined according to OL[x] functionality, where x is the output number.
1	Port B route to Output compare Output is used as an output signal when position compare function (OC[]) is active on Port B.
2	Port A is route to Output compare Output is used as an output signal when position compare function (OC[]) is active on Port A
7	STO output status Output is active in case one of the STO inputs (STO1_IN or STO2_IN) are not active and the drive is in safe active state.



GO[14...16] values	Description
0	General-purpose Output: The function of the output is defined according to OL[x] functionality, where x is the output number.
1	Port B route to Output compare Output is used as an output signal when position compare function (OC[]) is active on Port B.
2	Port A is route to Output compare Output is used as an output signal when position compare function (OC[]) is active on Port A
3	Daisy chain 1 Output is used to buffer AQB signals from Port B (Quad 0)
4	Daisy chain 2 Output is used to buffer AQB signals from Port A (Quad 1)
5	Emulation Output is used to emulate feedback according to EA[]
6	Reserved for backwards compatibility
7	Relevant for GO[1] to GO[4]
8	Buffer absolute digital sensor signals Output 14 to 16 (GO[14] to GO[16]) are buffered to PORT C (daisy chain) as follows: • Port C 'A': Buffer clock out of the sensor • Port C 'B': Buffer Data input of the sensor • Port C 'Index': Buffer Data output of the sensor

Notes

- Consult the Administrative Guide for the actual abilities and restrictions of the outputs
- When Output Compare is used the following sensors are valid for compare:
 - Encoder Quad Port A sensor ID 2
 - Encoder Quad Port B sensor ID 1
 - Analog Sine\Cosine on Port B sensor ID 3
 - Serial Absolute Hiperface on Port B sensor ID 31
 - Port A emulation sensor ID 14
 - Port B emulation sensor ID 13



- If the function denoted by **GO[]** is higher than the value 2 (i.e., Daisy Chain 0/1 or Emulation), all three relevant outputs of Port C (14, 15, 16) must be configured for the same function.
- If the function of **GO[14]** to **GO[16]** is not configured to Daisy Chain or Emulation, the drive function will fire blanks and no pulses will be generated.

References

GV[], GW[], OC[]



GP[N] – Error Mapping Correction Table Editing

GP[N] edits entries in the error mapping correction table.

CANopen/CoE

Elmo_parameters_objects: **0x30AB**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	RS232, USB, TCP, EoE
Restrictions	An error mapping correction table must be defined first with the PC[3] command.
Range	None
Index range	Depends on the selected table, see the PC[3] command.
Default	None
Unit modes	All
Non-volatile	Yes

Remarks

GP[N] enables editing of the error mapping correction table.

After setting **PC[3]** to the requested array, which will be used as the error mapping correction table, the table can be filled with correction values as defined in Error Mapping manual.

The index range of the **GP[]** command depends on the table selected in **PC[3]** (for optional tables and sizes, see the **PC[3]** command).

It is recommended not to edit the selected array directly, but to use the **GP[]** command.

The **GP[]** command protects against deviations of the table index.

The **GP[]** command does not protect against editing of the table while error mapping is enabled. Ambiguity may occur when it is incorrectly used during feature activation (**PC[1]** differs from 0).

The **GP[]** values are in user units.

References

PC[]



GS[] – Gain Scheduling

GS[] defines the gain scheduling process of the position and velocity controller.

CANopen/CoE

Elmo_parameters_objects: **0x30AE**

Attributes

Attribute	Description
Type	Long
Source	All
Restrictions	
Range	According to the table
Index range	1 to 20
Default	According to the array index, elsewhere is zero
Non-volatile	Yes
Activation	Immediate

Remarks

The Gold 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".

1. 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.
2. Gain scheduling by position may be necessary due to mechanical changes with position dependence.
3. 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 **KG[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. Manipulate it only if you are sure of what you are doing.

Indices

The following table lists the gain scheduling parameters. Unused indices are reserved for compatibility with older drives.

Index	Description	Default	Values	Restrictions
0	Reserved			
1	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.	100	>0	
2	Use scheduled controller according to the following:		0 to 68	
0	No gain scheduling (by KP[2] , KI[2] , KP[3])			
1 to 63	Specific controller from table			
64	Gain scheduling by speed (use GS[7] to set the speed source and position error, which are multiplied by GS[10])			
65	Gain scheduling by position. Use CA[65] to set the position socket and GS[19] , GS[20] to set the position limits.			
66	Best Settling scheduling.			



Index	Description	Default	Values	Restrictions
	- During profiler – index 63 KI=KG[63] KPvel=KG[126] KPpos=KG[189] - After the profiler stops and the time is less than GS[11] (ms) – index 62. KI=KG[62] KPvel=KG[125] KPpos=KG[188] - 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 0x2E00 low 8 bits			
	68 Controller index is manually scheduled: Index is by Object 0x2E00 high 8 bits			
3	With Quad encoders, select the minimum speed for calculating speed as 1/T. Above this speed the calculation is by 1/T. Units are counts/sec. With Hall sensors only, any value differing from zero will not calculate speed by 1/T.		>=0	
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, 1	
	0 By command			
	1 By feedback			



Index	Description	Default	Values	Restrictions
8	Position FIR filter (CA[88]) enable mode	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).
	0 Always enabled			
	1 Enabled after delay of GS[11] ms after the profiler stops			
	2 Enabled after the profiler stops			
9	Non-linear factor for position controller. This value limits the position controller output to a specific acceleration to close the position error. $Abs(KP[3]*Error) \leq \sqrt{2*GS[9]*abs(Error)}$. Units are counts/sec ² .	2×10^9	0 to 2×10^9	
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 1,200	



Index	Description	Default	Values	Restrictions
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.	0	0 to 8000	
12	Define functionality of object 0x2005	0	0,1	
	0 Can be used as position, and be connected to socket			
	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 continuous current.			
13	Define the maximum speed for gain schedule by speed (one of GS[2,16,17,18] is equal to 64)	6200	>=1	Should be higher than GS[1]
14 to 15	Reserved			
16	Use scheduled gains in velocity advanced filter #10, to disable zero KV[25]	1	1 to 68	
	1 to 63 Specific controller from table			
	64 Gain scheduling by speed (use GS[7] to set the speed source)			
	65 Gain scheduling by position (use CA[65] to set the position socket)			
	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			



Index	Description	Default	Values	Restrictions
	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 0x2E00 low 8 bits			
68	Filter schedule index is manually selected by Object 0x2E00 high 8 bits			
17	Use scheduled gains in velocity advanced filter #2, to disable zero KV[30] : 1 to 63 Specific controller from table 64 Gain scheduling by speed (use GS[7] to set the speed source) 65 Gain scheduling by position (use CA[65] to set the position socket) 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.	1	1 to 68	
	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 0x2E00 low 8 bits			
68	Filter schedule index is manually selected by Object 0x2E00 high 8 bits			
18	Use scheduled position advanced filter, to	1	1 to 68	



Index	Description	Default	Values	Restrictions
	disable zero KV[45] :			
	1 to 63 Specific controller from table			
	64 Gain scheduling by speed (use GS[7] to set the speed source)			
	65 Gain scheduling by position (use CA[65] to set the position socket)			
	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[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 0x2E00 low 8 bits			
	68 Filter schedule index is manually selected by Object 0x2E00 high 8 bits			
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.			
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.			



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]**.

The PIP controllers table(**KG[]**) can be changed when servo is enabled (**MO** = 1).

References

CA[], **KG[]**, **KV[]**



GV[N] – Output Compare Editing Table

GV[N] edits entries in the position table for output compare in module 0.

CANopen/CoE

Elmo_parameters_objects: **0x30B1**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	RS232, USB, TCP, EoE
Restrictions	The output compare table must be defined first in OC[7] .
Range	None
Index range	Depends on the selected table, see the OC[7] command.
Default	None
Unit modes	All
Non-volatile	No

Remarks

The **GV[]** command enables editing of the compare positions table for output compare table based modes regardless of the selected table.

After setting **OC[7]** to the requested array to be used as the position compare table, the table can then be filled with compare positions as defined in the output compare manual.

The index range of the **GV[]** command depends on the table selected in **OC[7]** (for the optional tables and sizes, see the **OC[7]** command).

It is recommended not to edit the selected array directly, but to use the **GV[]** command.

The **GV[]** command protects against deviation of the table index.

The **GV[]** command does **NOT** protect against editing the table while output compare – 0 is operational, i.e., **OC[1]** can report 1 or 2.

The **GV[]** values are in user units.

References

OC[], **GO[]**, **GW[]**



GW[N] – Output Compare Editing Table

GW[N] edits entries in the position table for output compare in module 1.

CANopen/CoE

Elmo_parameters_objects: **0x30B2**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	RS232, USB, TCP, EoE
Restrictions	The output compare table must be defined first in OC[27] .
Range	None
Index range	Depends on the selected table, see the OC[27] command.
Default	None
Unit modes	All
Non-volatile	No

Remarks

The **GW[]** command enables editing of the compare positions table for output compare table based modes.

After setting **OC[27]** to the requested array to be used as the position compare table, the table can then be filled with compare positions as defined in the output compare manual.

The index range of the **GW[]** command depends on the table selected in **OC[27]** (for the optional tables and sizes, see the **OC[27]** command).

It is recommended not to edit the selected array directly, but to use the **GW[]** command.

The **GW[]** command protects against deviation of the table index.

The **GW[]** command does **NOT** protect against editing the table while output compare – 1 is operational, i.e., **OC[21]** can report 1 or 2.

The **GW[]** values are in user units.

References

OC[], **GO[]**, **GV[]**



GX[] – Capture Array Value from HM

GX[] retrieves captured values from the capture array defined by **HM**.

CANopen/CoE

Elmo_parameters_objects: **0x30B3**

Attributes

Attribute	Description
Type	Long, Read-only
Source	RS232, USB, TCP, EoE
Restrictions	The captured buffer must be defined by the HM[11] command.
Range	None
Index range	HM[12] to HM[13]
Default	None
Unit modes	All
Non-volatile	No

Remarks

During position capture, the set of captured positions can be defined by **HM[11]** 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.

Note: If capture to array is selected ($1 \leq \mathbf{HM[11]} \leq 5$), the user can read the captured values from the array via the **GX[]** command.

If **HM[]** is not configured, **GX[]** will return an error. Otherwise, it will return the value that is currently in the selected array, according to the specified index.

The next index to be filled is indicated in **HM[9]**.

References

[HM\[\]](#)



GY[N] – Capture Array Value from HF

GY[N] retrieves captured values from the capture array defined by **HF**.

CANopen/CoE

Elmo_parameters_objects: **0x30B4**

Attributes

Attribute	Description
Type	Long, Read-only
Source	RS232, USB, TCP, EoE
Restrictions	The captured buffer must be defined by the HF[11] command.
Range	None
Index range	HF[12] to HF[13]
Default	None
Unit modes	All
Non-volatile	No

Remarks

During position capture, the set of captured positions can be defined by **HF[11]** in an array. In that case the **GY[]** 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.

Note: If capture to array is selected ($1 \leq \text{HF}[11] \leq 5$), the user can read the captured values from the array via the **GY[]** command.

If **HF[]** is not configured, **GY[]** will return an error. Otherwise, it will return the value that is currently in the selected array, according to the specified index.

The next index to be filled is indicated in **HF[9]**.

References

[HF\[\]](#)



HL[] /LL[] – High/Low Feedback Limit

HL[] and **LL[]** specify the minimum and maximum allowed motor speed and position.

CANopen/CoE

HL - Elmo_parameters_objects: **0x30C1**

LL - Elmo_parameters_objects: **0x3129**

Attributes

Attribute	Description
Type	Integer32, Read/Write
Source	All
Restrictions	No
Range	LL[3] : -2^{31} to $(2^{31} - 1)$ HL[2] : 0 to $(2^{31} - 1)$ HL[3] : -2^{31} to $(2^{31} - 1)$
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 = 2, 5 VL[3], VH[3] : UM = 5
Non-volatile	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 ($-\text{HL[2]}$), 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 $(2^{31} - 1)$
3	The motor position is limited to the range [LL[3]...HL[3]] .	User-defined	-2^{31} to $(2^{31} - 1)$

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[N]/HF[N] – Main/Aux Homing

HM[N] /HF[N] enables the ability to capture input events and to execute a predefined operation when an event occurs.

CANopen/CoE

HM - Elmo_parameters_objects: **0x30C2**

HF – Elmo_parameters_objects: **0x30BB**

Attributes

Attribute	Description
Type	Long
Source	USB, RS232, TCP, EoE
Restrictions	According to array index
Range	According to array index
Index range	1 to 13
Default	None
Unit modes	All
Non-volatile	No
Attribute	None

Remarks

The **HM[N]** command defines the main homing parameters, always work on the main feedback sensor.

The **HF[N]** command defines the auxiliary (Aux) homing parameters. The sensor is selected by the user (see **HF[10]**).

Note: The following text refers to the **HM[N]** command. It also applies to the **HF[N]** command unless stated otherwise.

The command sets and gets parameters of the Main/Aux 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:

1. Modify the main feedback position counter
2. Log the event position counter
3. Flag a digital output

An event is a change in a digital input signal. The polarity of the change is defined by the **IL[]** command.



Currently a drive supports two types of capture accuracy:

1. Index, Home. Fast accurate hardware capture. Using these inputs must be configured by the **GI** command
2. All other inputs. SW capture with a delay configured in the **IF** command

Values defined in **HM[3]** are duplicated for compatibility reasons.

A drive currently supports six inputs. If **HM[3]** ≥ 21 , the behavior will be unpredictable.

All **HM[N]** entries that are not read-only, except **HM[1]**, can be changed during the homing procedure. The activation of these values will be performed at the next homing activation, that is, when **HM[1]** $\neq 0$.

If **HM[2]** is set to a value beyond **XM[1]** and **XM[2]**, the actual main position will not be updated when homing is complete. (Feature not available at this time)

Each homing event is attached to a predefined functionality (FLS, RLS, general-purpose, home, and so on). If the corresponding input is not defined first, the homing procedure may never end. Refer to the **IL[N]** command.

IN5 and IN6 can be captured by hardware. In order to use this feature, use the **GI[N]** command to direct IN5 or IN6 into home capture, and configure the homing parameters to a home event. Refer to the **GI[N]** command.

The homing and capture procedures can be carried out in any unit mode (**UM** = 1, 2, 3, 4, 5).

- In external reference mode (**RM** = 1), when **HM[4]** = 0, the software portion of the reference is stopped, while the external portion is not. In such cases, the motor continues to move according to the analog reference. (Feature not available at this time)
- When capturing more than one event is configured, i.e., when **HM[1]** > 1 , the delta between each event must be $(4 \cdot \text{TS})$ for digital inputs, and $(2 \cdot \text{TS})$ for home input or Index.
- When both **HM[N]** and **HF[N]** are run, and using the main feedback sensor is selected (**HF[10]** = 1), the results are unpredictable.
- When **HM[11]** = 5, capture is added to the data recording array, and data recording should not be used.
- If **HM[11]** $\neq 0$ and **HM[1]/HF[1]** is finite and exceeds the number of possible entries in the array, the value of **HM[1]/HF[1]** value will be saturated to the following value: $(\text{HM}[13] - \text{HM}[12] + 1)$.
- The capture delay is input type-dependent. For Index or Home inputs the delay is $2 \cdot \text{TS}$. For other inputs it is dependent on the configured input filter (see the **IF** command).
- DS-402 homing and DS-402 Touch Probe cannot work while **HM** or **HF** are running.



Indices

The following table describes the **HM[N]/HF[N]** entries.

Index	Purpose	Value	Description
0	Reserved	None	Reserved
1	Activation mode	0	Stop the homing process. HM[1] is automatically reset to 0 when homing is complete.
		≥ 1	Number of events. HM[4] is performed at last event.
		32000	Infinite number of event captures. HM[4] is ignored. If Capture to array is selected (see HM[11]), the array will be filled between HM[12] and HM[13] in an infinite loop.
2	Absolute /relative value		Value to load, 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 .
		1/2	Event according to main home switch capture. The first event is always high transition.(High transition - level change from low to high (rising edge)). The home switch is selected according to the GI command.
		3	High transition (High transition - level change from low to high (rising edge)) of index pulse (capture).
		4	Low transition (Low transition, level change from high to low (falling edge)) of index pulse (capture).
		5/6	Event according to the FLS switch.
		7/8	Event according to the RLS switch.
		9/10	Event according to the DIN1 switch.
		11/12	Event according to the DIN2 switch.
		13/14	Event according to the DIN3 switch.
		15/16	Event according to the DIN4 switch.
		17/18	Event according to the DIN5 switch.
		19/20	Event according to the DIN6 switch.
		21/22	Event according to the DIN7 switch.
		23/24	Event according to the DIN8 switch.
		25/26	Event according to the DIN9 switch.
		27/28	Event according to the DIN10 switch.



Index	Purpose	Value	Description
4	After event behavior. Executed when HM[1] reaches 0.	0 default	In UM = 1,2,3,4,5: stop immediately using the SD deceleration value.
		1	Set digital output. Equivalent to OP = HM[6] .
		≥2	Do nothing.
5	What to set for PX during event	0 default	Absolute setting of the position counter: PX = HM[2] . Note: This is not allowed in case HF is used and configured on the main position feedback socket.
		1	Not Supported
		==2	Do nothing.
		== 3	HM[5] - Do nothing. HF[5] – Absolute setting of the sensor position counter to HF[2] value. In case the sensor is defined as Gantry, this will adjust all relevant Gantry sockets.
		>3	Do nothing.
6	Output value		Digital output value to set if HM[4] = 1. Only outputs defined as general outputs are affected.
7	Captured value (PX)		The capture value of PX (read-only). The position value is captured before PX/FP is changed according to HM[5] .
8	Reserved	None	None
9	Next capture array index		Read only value, indicating the index in the capture array for inserting the next captured value. The captured value can be read via the GX/GY command.
10	Socket selection	1 to 4	Selects the socket to be used for HF . This command is available only in HF[] .



Index	Purpose	Value	Description
11	Capture array selection	1	The ZX array is used for capture in the range from 1 to 1023.
		2	The NT array is used for capture in the range from 1 to 255.
		3	The ET array is used for capture in the range from 1 to 2048.
		4	The UI array is used for capture in the range from 1 to 24.
		5	The BH array is used for capture in the range from 1 to 16383 (When this array is used, data recording is not available).
		All others	No array is selected.
12	Capture array low index		Low capture array index, to be filled with capture values.
13	Capture array high index		High capture array index, to be filled with capture values.

Examples

Example 1

The following example uses capture on the main home switch when input-5 is routed into encoder-1 strobe input by the **GI[]** command.

HM[1] will be set to 0 in case that the digital input 5 will be logically set to '1'.

If the example is used on **HF**, **HF[10]** must be configured (**HF[10]** = 1).

Command	Description
HM[1]=0	Disable the ongoing homing sequence
HM[2]=1000	Home position is been offset by 1000 counts from input
IL[5]=17	Input-5 is configured as homing switch, active high
GI[4]=6	Route input-5 into Encoder-1 strobe input
HM[3]=2	Wait for the event on home signal (the first rising edge)
HM[1]=1	Start searching for a single event

Example 2

The following example uses capture on the main home switch when input-5 is routed into encoder-1 strobe input by the **GI** command, and event values are added to the **BH** array infinitely.

If the example is used on **HF**, **HF[10]** must be configured (**HF[10]** = 1).



Command	Description
HM[1]=0	Disable the ongoing homing sequence.
HM[2]=0	Home position is offset by 0 from the strobe input.
IL[5]=17	Input 5 is configured as homing switch, active high.
GI[4]=6	Route input-5 into Encoder-1 strobe input.
HM[3]=2	Wait for the event on home signal (first signal is always a rising edge).
HM[11]=5	Select the BH array for event storage.
HM[12]=1	The low BH index is 1.
HM[13]=16000	The high BH index is 16000.
HM[1]=32000	Start capturing events and add them into the BH array. HM[9] will indicate the next BH array index to be filled. The values are read using the GX command.

Reference

GI, GX, GY, IL, IF



HP – Halt Program

HP halts the user program.

CANopen/CoE

Elmo_parameters_objects: **0x30C5**

Attributes

Attribute	Description
Type	Command
Source	All, except the user program
Restrictions	No
Range	None
Default	None
Unit modes	All
Non-volatile	No

Remarks

The command halts execution of the user program.

A subsequent **XC** command resumes the program from the instruction at which the program was halted. A pending Auto-Routine will remain pending.

An **HP** command issued when no program is running does nothing and sets no error.

The **XC** command resumes execution after a halt.

Program status (**PS**) is 0 when the user program is halted.

References

KL, XQ, XC



HT[] – Open Loop Torque

HT[] specifies the open loop torque for stepper 2-phase

CANopen/CoE

Elmo_parameters_objects: **0x30C9**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Range	According to array index
Index range	1 to 3
Default	0
Unit modes	UM=6 (open and closed loop stepper)
Non-volatile	Yes

Remarks

The **HT[]** command sets the open loop torque 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]	



Index	Description	Units	Values	Restrictions
3	Torque when there is acceleration. Receives the maximum value when between HT[1] and HT[3] . If HT[1] > HT[3] the maximum value obtained is HT[1] .	Amperes	0 to PL[1]	

References

[CL\[\]](#), [FF\[\]](#), [PL\[\]](#), [UM](#)



HX – Hexadecimal Mode

HX specifies a parameter that allows hexadecimal numbers to be displayed, set and indicated.

CANopen/CoE

Elmo_parameters_objects: **0x30CD**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	USB, TCP, RS232. Has no influence on other.
Restrictions	No
Range	0, 1
Default	0
Unit modes	All
Non-volatile	No

Remarks

The **HX** parameter allows a hexadecimal reply to the host when USB, RS232 and TCP are used. The format eases the reading and understanding of bit-field variables, such as the digital inputs port (**IP**), servo drive status (**SR**) and more.

When **HX** = 0, Longs are reported as decimal numbers.

When **HX** = 1, Longs are reported as hexadecimal numbers.

The **HX** parameter is not required for setting values. The commands **BH=1024** and **BH=0x400** are equivalent, as 0x400 equals its decimal equivalent 1024.

Floating point numbers cannot be presented in hexadecimal format.



IA[] – Index Analog Sensor

IA[] specifies an interrupt for capturing the index of the Sine/Cosine analog encoder and specifies the encoder position at the index location for fast index capture. This procedure should be performed once, when setting an analog encoder to a drive.

CANopen/CoE

Elmo_parameters_objects: **0x30D0**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	Use for an analog sin/cos encoder with an index. This procedure should be managed by EAS.
Range	According to the table.
Index range	1 to 4
Default	0
Unit modes	Any
Non-volatile	Yes

Indices

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

Index	Description when set		Description when read		Values
1	Enable/disable the index interrupt.		The index interrupt is enabled/disabled.		0, 1
	0	Disable the interrupt	0	The index interrupt is disabled.	
	1	Enable the interrupt	1	The index interrupt is enabled.	
2	Clear the interrupt flag.		The interrupt flag is cleared/ set.		0, 1
	0	No effect	0	The interrupt flag is cleared.	
	1	Clear the interrupt flag.	1	The interrupt flag is set (the position was captured).	



3	Set the angle at index rising edge interrupt.	Captured angle at index rising edge interrupt.	
4	Set the angle at index rising edge interrupt.	Captured angle at index rising edge interrupt.	

References



IB[] – Digital Input Bits

IB[] reads a digital input bit.

CANopen/CoE

Digital_inputs: **0x60FD**Elmo_parameters_objects: **0x30D1**

Attributes

Attribute	Description
Type	Long Indices 17 to 32 R/W, other Read-only
Source	All
Restrictions	None
Range	0, 1
Index range	1 to 32
Default	None
Unit modes	All
Non-volatile	No

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.

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	
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
23..30	Reserved. Always 0	0		
31..32	Digital input 15..16 logical pins state	Long	0..1	Write 1 to clear sticky bit

References

[IP](#), [IL\[M\]](#)



ID, IQ – Active/Reactive Current

ID and **IQ** get the active (**IQ**) and the reactive (**ID**) components of the motor current, in amperes.

CANopen/CoE

ID - Elmo_parameters_objects: **0x30D3**

IQ - Elmo_parameters_objects: **0x30E0**

Attributes

Attribute	Description
Type	Float, Read-only
Source	All
Restrictions	None
Range	N/A
Index range	N/A
Default	N/A
Unit modes	All
Non-volatile	No

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\[M\]](#), [CL\[M\]](#)



IF[] – Digital Input Filter

IF[] defines the time period of the digital input filter.

CANopen/CoE

Elmo_parameters_objects: **0x30D5**

Attributes

Attribute	Description
Type	Parameter, Float
Source	All
Restrictions	None
Range	0.0 to 500.0
Index range	1 to 16
Default	0 (no filter)
Unit modes	All
Non-volatile	Yes

Remarks

The IF[] is defined in milliseconds. Input pulses which are shorter than IF[] will be discarded. Inputs pulses which are longer than IF[] will be sensed.

The digital filter basic time is 250 µsec by default.

Note:

The actual number of digital inputs depends on the drive hardware. Typically, 6 inputs are available. The drive firmware will allow the inputs setting even if the drive's hardware is not available.

The input filter is a deterministic period function in the firmware. The actual filter time will be the time (milliseconds) which is closest, but not shorter, to the requested time. The resolution depends on the filter period, which might differ between versions.

Example

If the IF[x] is set to 1.1, the filter period will be 1.25 msec, which is the filter period closest to the requested value.

In cases in which the filter basic period is 100 µsec, the actual filter time will be 1.1 msec.

In any case, pulses which are shorter than the requested value will be rejected.

In cases in which the digital input is used for position capture (homing on home switch, or touch probe) the Software filter is not respected, since the position is captured by hardware, much faster than any firmware period.



Hardware filtering of inputs is not available at this stage.

For inputs that are not supported by the product, the relevant index will be accepted but ignored.

Indices

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

Index	Description	Type	Values	Restrictions
1 to 6	Digital input filter in msec	Float	0.0 to 500.0	

References

IB[], IL[],IP



IL[] – Digital Input Logic

IL[] specifies the function and logic of the digital inputs.

CANopen/CoE

Elmo_parameters_objects: **0x30DB**

Attributes

Attribute	Description
Type	Long, Read/Write, Bit Filed
Source	All
Restrictions	If hardware revision C (RevC) is used than only input 5 can be used as main Home switch. If drive hardware revision is revision E (RevE), any input can be used for Home and capture purposes. The routing of the desired input for capture purposes is done via GI command. The revision type can be retrieved via WS[8]
Range	N/A
Index range	1 to 16
Default	Input 1 to 6: 7 – General purpose, high logic level Input 8 to 16: 5 – Ignore
Unit modes	All
Non-volatile	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 depends on the drive hardware. The software, however, is designed for 6 inputs regardless of the hardware ability.

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:

Digital inputs 11 to 16 are read from the differential encoder entries of Port A and Port B. In order to prevent triggering of these signals, set the commands **IL[11]** to **IL[16]** to the value "Ignore" (**IL[N]** = 4 or 5). "ignore" is also the default value of these 6 additional inputs.

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.
IL[N] = 1	High	Note that when this function is required when the switch is not active the servo is automatically enabled.
IL[N] = 2	Low	Stop immediately under control: soft and auxiliary stop.
IL[N] = 3	High	Stop immediately under control: soft and auxiliary stop.
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.
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.



Command Value	Logic Level	When Active
IL[M] = 15	High	Stop immediately under control: soft stop only. Activates the ST command.
IL[M] = 16	Low	Enable the Main Home sequence. (N/A for the Gold Whistle)
IL[M] = 17	High	Enable the Main Home sequence. (N/A for the Gold Whistle)
IL[M] = 18	Low	Reserved
IL[M] = 19	High	Reserved
IL[M] = 20	Low	Stop immediately under control: stop both the software profiler and the auxiliary reference.
IL[M] = 21	High	Stop immediately under control: stop both the software profiler and the auxiliary reference
IL[M] = 22	Low	Abort motion. Shut off the servo drive, freewheel
IL[M] = 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[M] = 28	Low	Set RM =1; Engage Follower and ECAM
IL[M] = 29	High	Set RM =1; Engage Follower and ECAM
IL[M] = 30	Low	Set RM =0; Disengage Follower and ECAM
IL[M] = 31	High	Set RM =0; Disengage Follower and ECAM
IL[M] = 262	Low	General-purpose sticky input
IL[M] = 263	High	General-purpose sticky input

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	
1 to 4	Function number and behaviors * See detailed description in the section Function	Long	0	Shut off the servo drive, freewheel.	
			1	Stop immediately under control: soft and auxiliary stop.	



	Description.		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	Enable the Main Home sequence.	
			9	Enable the Auxiliary Home sequence.	
			10	Stop immediately under control: 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				
8	Sticky function	Boolean	0	Non sticky	
			1	Sticky	



9 to 15	Reserved			

Function Description

Function 0: Inhibit (freewheel)

When the inputs is 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 depended 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 depended 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

Not supported.



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[M]** 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[M]** command).

This function activates the #@AUTO_HM routine in the user program, depending on its declaration.

Function 10: Hard and Soft stop

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**



IP – Input Port

IP reports the status of the digital inputs.

CANopen/CoE

Digital_inputs:0x60FDElmo_parameters_objects:0x30DF

Attributes

Attribute	Description	
Type	Bit Field, Read-only	
Source	All	
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
	Bits 11 to 15	Not used. Always zero.
	Bit 16	Digital input 1 logical pin state
	Bit 17	Digital input 2 logical pin state
	Bit 18	Digital input 3 logical pin state
	Bit 19	Digital input 4 logical pin state
	Bit 20	Digital input 5 logical pin state
	Bit 21	Digital input 6 logical pin state
	Bits 22 to 25	Reserved. Always 0
	Bit 26	Digital input 11 logical pin state. Port A, A encoder entry
	Bit 27	Digital input 12 logical pin state. Port A, B encoder entry



Attribute	Description	
	Bit 28	Digital input 13 logical pin state. Port A, INDEX encoder entry
	Bit 29	Digital input 14 logical pin state. Port B, A encoder entry
	Bit 30	Digital input 15 logical pin state
	Bit 31	Digital input 16 logical pin state
Unit modes	All	

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.

Note:

Digital inputs 11 to 16 are read from the differential encoder entries of Port A and Port B. In order to prevent the triggering of these signals, set **IL[11]** to **IL[16]** to "ignore" (**IL[N]** = 4 or 5).

References

[IB\[N\]](#), [IL\[N\]](#)



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**

Elmo_parameters_objects: **0x30F9**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	- Motor must be on (except for PA[4]). - Effective on the next call to BG[N]
Range	$-(2^{31}) \dots (2^{31}) - 1$
Index Range	1 to 4
Default	0
Unit modes	UM = 2; UM = 5
Non-volatile	No

Indices

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

Remarks

On the next **BG** after applying the **JP** 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 reflected in **OV[2]** (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 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 the Profile Velocity mode.

CANopen/CoE

Target_velocity: **0x60FF**Elmo_parameters_objects: **0x30FF**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	• The motor must be on. • Effective on the next call to BG
Range	$-(2^{31}) \dots (2^{31})-1$
Default	0
Unit modes	UM = 2; UM = 5
Non-volatile	No

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 reflected in **OV[2]** (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**



KG[] – Gain Scheduled Controller

KG[] specifies the parameters of the gain scheduled speed, position controller and advanced filters. The **KG[]** parameters apply only if the controller gains are scheduled (see the **GS[]** command).

CANopen/CoE

Elmo_parameters_objects: **0x310A**

Attributes

Attribute	Description
Type	Real
Source	All
Restrictions	None
Range	Refer to the commands KI , KG and KP for the specific dependent ranges
Index range	1 to 945
Default	0
Unit modes	N/A
Non-volatile	N/A
Activation	For KP and KI immediate For an advanced filter, only at activating the filter KV[N] .

Remarks

Velocity advanced filter #1 is configured by **KG[190...441]** and activated by **KV[25]**, and gain schedule mode is activated for it by **GS[16]**.

Velocity advanced filter #2 is configured by **KG[442...693]** and activated by **KV[30]**, and gain schedule mode is activated for it by **GS[17]**.

Position advanced filter is configured by **KG[694...945]** 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 **KG[]** parameters array:

Index	KG[M] Value	Units	Length
1 to 63	KI for inner loop	Hz	63
64 to 126	KP for inner loop	Amperes/ (counts/sec)	63
127 to 189	KP for outer loop	rad/sec	63
190 to 252	Parameter 1 for scheduled velocity advanced filter #1	By filter type KV[25] and gain schedule method GS[16]	63
253 to 315	Parameter 2 for scheduled velocity advanced filter #1		63
316 to 378	Parameter 3 for scheduled velocity advanced filter #1		63
379 to 441	Parameter 4 for scheduled velocity advanced filter #1		63
442 to 504	Parameter 1 for scheduled velocity advanced filter #2	By filter type KV[30] and gain schedule method GS[17]	63
505 to 567	Parameter 2 for scheduled velocity advanced filter #2		63
568 to 630	Parameter 3 for scheduled velocity advanced filter #2		63
631 to 693	Parameter 4 for scheduled velocity advanced filter #2		63
694 to 756	Parameter 1 for scheduled position advanced filter #2	By filter type KV[45] and gain schedule method GS[18]	63
757 to 819	Parameter 2 for scheduled position advanced filter #2		63
820 to 882	Parameter 3 for scheduled position advanced filter #2		63
883 to 945	Parameter 4 for scheduled position advanced filter #2		63

References

GS[], KV[], 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_parameters_objects: **0x310C**

KP - Elmo_parameters_objects: **0x3113**

Attributes

Attribute	Description
Type	Parameter, Real
Source	All
Restrictions	None
Range	KI[N] > 0 KP[N] > 0, KP[6] can also be a negative value
Index range	KP[1 to 6] , KI[1, 2, 4]
Default	0
Unit modes	See below.
Non-volatile	Yes
Activation	Immediate, only when the motor is turned on

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.

KP[6] for gantry, defines the factor of the yaw control to the Y axes in a planar motor. A factor of 1 should transmit the same current command of X yaw to the Y axes. For stepper closed loop, this value is converted from encoder counts to stepper angle counts ($512 \cdot \text{CA}[19] / \text{CA}[18]$)

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	Type	Unit Modes	Units
1	Defines the PI current controller		All	KP : volt/ampere KI : Hz
2	Defines the PI velocity controller		Position and velocity (when GS[2] = 0)	KP : ampere/(counts/sec) KI : Hz
3	Defines the gain of the position controller		Position (when GS[2] = 0)	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	Define the factor of the Yaw control to Y axes in planar motor		Position and current with planar motor	KP : N/A

References

KV[], **GS[M]**, **KG[]**



KL – Kill User Program

KL stops execution of the user program and turns the servo off.

CANopen/CoE

Elmo_parameters_objects: **0x310F**

Attributes

Attribute	Description
Type	Command
Source	All, except the user program
Restrictions	None
Range	None
Default	None
Unit modes	All
Non-volatile	No

Remarks

The **KL** command permanently stops the user program.

The program can run again from the start by using the **XQ** command.

The program status after the **KL** command (refer to **PS** command) is -1.

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.

References

HP, **XQ**, **XC**, **PS**



KR – Kill Motion Repetitive

KR command stops the ongoing special motion.

CANopen/CoE

Elmo_parameters_objects: **0x3115**

Attributes

Attribute	Description
Type	None
Source	USB, RS232, TCP, EoE
Restrictions	PTP mode only
Range	None
Index range (used in vectored commands)	None
Default	0
Unit modes	5
Non-volatile	No
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. Please 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.

The **KL** command disables the servo while stopping the special motion mode.



References

[UM](#), [SR](#), [MR\[N\]](#), [ST](#)



KV[] – High-Order Controller Filter Parameters

KV[] specifies the parameters of the second-order 2x2 filters. This filter has DC gain (gain is equal to one in zero frequency).

Indexes equal to 5n are specify the filter type according to Table 9. When changing filter type drive check the validity of filter parameters. This parameter must be changed at motor disabled.

CANopen/CoE

Elmo_parameters_objects: **0x3119**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	The motor must be off.
Range	See the tables below.
Index range	1 to 100
Default	0
Unit modes	All
Non-volatile	Yes

Remarks

Each filter has five parameters. The first four parameters are the physical 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]	KG[190...441], GS[16]
	Gain schedule filter #2	KV[30]	KG[442...693],



Filter Location	Filters	Parameters	Gain Schedule Parameters
			GS[17]
High-order position controller filters (which filter the proportional output)	Filter #1	KV[31] to KV[35]	
	Filter #2	KV[36] to KV[40]	
	Gain schedule filter #1	KV[45]	KG[694...945], GS[18]
		KV[46] to KV[50]	Reserved
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]	
Analog input #1	Filter #1	KV[71] to KV[75]	
		KV[76] to KV[80]	Reserved
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 9 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 there P1 to P4 parameters are in **KG[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		
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 get the lead filter, in the lead/lag filter, the phase should be positive.
- When setting gain schedule filter (**KV[25]**, **KV[30]**, and **KV[45]**) it checks and set all the filters in table, but output an error only according to relevant indexes in (**GS[16]**, **GS[17]**, and **GS[18]**), so better to set first **GS** and later **KV**

References

[KG\[\]](#), [GS\[\]](#)



LC – Current Limit Flag

LC reports the status of the current limiting process.

CANopen/CoE

Elmo_parameters_objects: **0x3120**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	None
Range	0, 1
Default	0
Unit modes	All
Non-volatile	No

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.

To protect the drive the following condition should be met:

$$MC[A]^2 * 3 [sec] \geq PL[1][A]^2 * PL[2] [sec]$$

LC returns values according to the following table:

Value	Description
0	The motor current is limited by the limit PL[1] , or the motor is off.
1	The motor current is limited by the continuous limit CL[1] .

References

- [MC](#), [PL\[\]](#), [CL\[\]](#)



LD – Load Data

LD retrieves the load parameters from non-volatile memory and resets the volatile memory to their default values.

CANopen/CoE

0x1011 with data bytes 0-3 'l', 'o', 'a', 'd'

Elmo_parameters_objects: **0x3121**

Attributes

Attribute	Description
Type	Command
Source	All, except User Program
Restrictions	• The motor must be off. • The user program must be at rest. • Wizard mode must not be active.
Range	None
Default	None
Unit modes	All
Non-volatile	No

Remarks

The **LD** command is used to restore non-volatile application parameters from the flash memory to the RAM. After successful loading, the parameters are processed to their internal and real-time values.

The **LD** command resets volatile parameters to their default values.

During the **LD** sequence, relevant parameters are processed automatically by the drive in a procedure which is similar to what the interpreter would do if the parameter arrived from the communication channel. This assures that the loaded parameter is ready for any function, it was designed for.

In cases in which an error occurs during this post processing, the drive performs an automatic **RS** command, which forces all parameters to take their default values. The reason for the failure can then be retrieved by **CD** command.

To avoid a loss of communication, the **PP[]** parameters which define the communication attributes should not be processed.

The **SV** command saves the parameters in the flash memory.



The **LD** procedure may take a long time (tens of milliseconds), and during that time no other command can be processed.

References

SV, CD



LP[] – Load Program Info

LP[] gets user program information.

CANopen/CoE

Elmo_parameters_objects: **0x312D**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	None
Range	None
Index range	1 to 4
Default	0
Unit modes	All
Non-volatile	No

Remarks

If a user program is loaded into the drive, the **LP[N]** command provides information about relevant properties of this program.

The **PS** command can assure validation or non-validation of **LP[N]** data. See the **PS** command information.

Indices

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

Index	Description	Type	Values	Restrictions
0	Reserved			
1	User program code segment address . Points to the location of the user program in the non-volatile memory for uploading purposes.	Long	-	See note below.
2	Total length of the code required for program execution (as in LP[4]) and the program text .	Long	-	See note below.



Index	Description	Type	Values	Restrictions
3	Program text segment address from where the program can be uploaded for debugging purposes.	Long	-	See note below.
4	Non-text total length : code, symbols, functions, variables.	Long	-	See note below.

Note: The **LP[N]** command can be used even when there is no program in the drive. In that case, no error is shown.

References

[CC](#), [XQ](#), [PS](#)



MC – Maximum Current

MC reports the maximum phase current (peak current) allowed for the drive, in amperes. This command informs the software about the rate of the servo drive used with the controller.

The **MC** value is burned in during the production of the drive and cannot be modified by the user.

The drive can run at the current specified by **MC** for 3 seconds.

CANopen/CoE

Elmo_parameters_objects: **0x313A**

Attributes

Attribute	Description
Type	Float, Read-only
Source	All
Restrictions	No
Range	N/A
Default	According to the servo drive. It cannot be changed.
Unit modes	All
Non-volatile	N/A
Activation	N/A

Remarks

The current can be limited with the **PL[1]** and **CL[1]** commands.

References

[IQ](#), [ID](#), [CL\[\]](#), [PL\[\]](#)



MF – Drive Fault

MF reports and latches the reason that caused the motor to be disabled (**MO** = 0).

CANopen/CoE

Elmo_parameters_objects: **0x313D**

An EMCY message is transmitted when **MF** occurs. The message contains the fault reason. EMCY messages are valid to any DS-402 channel either by EtherCAT or CANopen.

Note that the CANopen "Fault reset" command does not clear the **MF** value and allows the next motor enable command.

Attributes

Attribute	Description
Type	Bit field, Read-only
Source	All
Restrictions	None
Range	None
Default	0
Unit modes	All
Non-volatile	No

Remarks

MF will not report the reason if the motor was shut off from any of the interpreting channels, such as User Program, Serial Communication, CANopen, EtherCAT or TCP/IP.

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 INH/ENA input.

After the motor is shut down due to a fault, the drive should prevent enabling of the motor for 7.5 milliseconds.

If the fault is caused by an amplifier fault, the red LED will be set, and the **AOK** function will be activated (see the **OL[]** command).

The AUTO_ER routine of the user program should be activated upon an **MF** event.



The following table details the bit-field structure with respect to the fault reason.

MF Value (Hex)	Description	Type CAN EMCY (Hex)	Notes
1 (0x1)	Main feedback error	81 7300	Immediate Abort: Servo Off For analog feedbacks check the threshold level in CA [48] and CA[49] For an absolute encoder check the reason in EE[1] . The fault causes a commutation search on the next motor enable.
2 (0x2)	Commutation process fail during motor on	81 7382	Immediate Abort: Servo Off For looking the phase Planar motor when on alignment process
4 (0x4)	Hall main feedback mismatch	81 7380	Immediate Abort: Servo Off Illegal Halls
8 (0x8)	Current exceeded peak limit	21 8311	Immediate Abort: Servo Off 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.
16 (0x10)	External Inhibit was triggered (INH/ENB)	21 5441	According to 0x605E See IL[] for more details about the Inhibit/Abort functions. Note the Additional Abort function. The Inhibit function is also activated if both limit switches (FLS & RLS) are active simultaneously in Cyclic Synchronous Position motion mode.
32 (0x20)	AC fail: Loss of phase	11 3130	According to 0x605E Requires specific hardware configuration dependent on drive. Loss of phase.



MF Value (Hex)	Description	Type CAN EMCY (Hex)	Notes
64 (0x40)	Halls sensor speed is too high.	81 7381	Immediate Abort: Servo Off
128 (0x80)	Speed tracking error	81 8480	According to 0x605E The difference between the commanded speed to the control loop and the feedback exceeded the value defined in ER[2] . This indication is not related to the Max Slippage Error as defined in the DS-402 Profile Velocity mode.
256 (0x100)	Position tracking error	21 8611	According to 0x605E The difference between the commanded position to the control loop and the feedback position exceeded the value in ER[3] . This indication is not related to the Following Error as defined in the DS-402 Profile Position mode.
1024 (0x400)	Gantry Yaw or Stepper closed loop Position error		The Yaw difference in gantry or position error in stepper closed loop, between profile (defined in CA[98]) and feedback (defined in CA[99]) which exceeded the value in ER[5] . This indication is not related to the Following Error as defined in the DS-402 Profile Position mode.
2048 (0x800)	Heartbeat event or frame loss. (communication)	11 8130	According to 0x605E The motor was shut due to a heartbeat event according to CANopen DS301 object 0x1016 .
4096 to 32768 (0x1000 to 0x8000)	Amplifier problem	(See table below)	Immediate Abort: Servo Off Indicates the problem that the power section of the drive has encountered.
131072 (0x20000)	Overspeed indication	81 8481	According to 0x605E The motor speed has exceeded the value which is defined in HL[2] . The motor main speed is reported in



MF Value (Hex)	Description	Type CAN EMCY (Hex)	Notes
			VU.
2097152 (0x200000)	Motor is stuck	21 7121	According to 0x605E 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 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.
4194304 (0x400000)	Feedback is out of position limits	81 8680	According to 0x605E The main position feedback exceeded the HL[3] or LL[3] limit. The main feedback is reported in PU .
8388608 (0x800000)	Numeric overflow - ambiguity in results	81 FF30	Immediate Abort: Servo Off An internal mathematical problem occurred.
16777216 (0x1000000)	Gantry slave disabled		Immediate Abort: Servo Off Gantry master disable because gantry slave is not enabled at current mode



MF Value (Hex)	Description	Type CAN EMCY (Hex)	Notes
536870912 (0x20000000)	Failed to start motor	81 FF10	Immediate Abort: Servo Off Commutation auto-phasing failed, and the motor could not be started. A request to initiate the motor using a CANopen control word failed. Possible problems may be: Inhibit/abort switches are active. Commutation auto-phasing failed. The PAL is not initiated/burned. Too little time has passed since the last fault (typically 7.5 msec) or the last motor disable. Profiler initiation failed due to conflicts between one of the profiler parameter/objects (reason in EE[2]).

The following table details the Amplifier Status bits indication.

MF Indication 0x1000 to 0x8000 Value (Hex)	Description	Type CAN EMCY (Hex)	Notes
0	All OK		
12288 (0x3000)	Undervoltage: The amplifier is not measuring the minimum required voltage.	5 3120	The minimum allowed value is reported in the WI[37] (burnt) and WI[38] (actual) command. Actual bus voltage is reported AN[6] .
20480 (0x5000)	Overvoltage: The amplifier is measuring a voltage which is higher than the allowed threshold.	5 3310	The maximum allowed voltage is reported in the WI[35] (burnt) and WI[36] (actual) command. The actual bus voltage is reported in AN[6] .
28672 (0x7000)	Safety: One or two of the safety inputs are in safety state.	5 FF20	The safety indications are reported in SR bits 14 and 15.
45056	Short Protection:	3 2340	This instantaneous fault is measured



MF Indication 0x1000 to 0x8000 Value (Hex)	Description	Type CAN EMCY (Hex)	Notes
(0xB000)	The current has exceeded a range which is considered as a phase-to-phase or phase-to-ground short.		by the hardware and typically cannot be recorded or indicated outside of the MF command.
53248 (0xD000)	Over-temperature: The drive is sensing a temperature which exceeds the maximum allowed temperature limit.	9 4310	The actual temperature is reported by the TI[1] (TI[2] in Fahrenheit) command.
61440 (0xF000)	Additional Abort was activated. The drive sensed an input switch that is defined as Additional Abort (refer to IL[] command)	81 5442	The fault is similar to "Abort" function with different value report. This allows user to distinguish between two different faults states such as Inhibit and PTC.

References

[SR](#)



MI – Mask Interrupts

MI masks the execution of specified automatic routines in the user program.

CANopen/CoE

Elmo_parameters_objects: **0x3140**

Attributes

Attribute	Description
Type	Bit field, Read/Write
Source	All
Restrictions	The AUTO_PERR routine is not maskable.
Range	Any bit:
	0 The automatic routine is allowed.
	1 The automatic routine is masked.
	0 to 65535
Default	0 (All allowed)
Unit modes	All
Non-volatile	Yes

Remarks

A user program may include 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.
- It may be required that a certain code sequence is un-interruptible.

Note: **MI** masks the execution but does not prevent it. The routine is executed after **MI** allows it.

The bit field characteristic of the **MI** command allows blocking of several automatic routines in a single command.

MI prevents calling of the routine while the specific bit is set. A blocked routine will be called when the specific bit in **MI** is reset to 0.

If AUTO_PERR is activated, all other interrupts are automatically masked (**MI** = 0x7fff).



When an automatic routine is called, the first executable line is performed under a "critical section," allowing the user to set **MI** in the same instance of the routine called.

The **MI** bits are detailed in the following table. The routines are listed in order of descending priority.

MI 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
512 (0x200)	Digital input 3	AUTO_I3
1024 (0x400)	Digital input 4	AUTO_I4
2048 (0x800)	Digital input 5	AUTO_I5
4096 (0x1000)	Digital input 6	AUTO_I6
8192 (0x2000)	Main Home event	AUTO_HM
32,768 (0x8000)	User program error	AUTO_PERR

References

xQ, xC



MO/SO – Motor On, Servo On

MO enables and disables the motor, and **SO** checks the servo state.

CANopen/CoE

MO - Elmo_parameters_objects: **0x3146**

SO - Elmo_parameters_objects: **0x31E2**

DS-402 state machine using the Control Word (**0x6040**) and Status Word (**0x6041**) objects.

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	0, 1
Default	0
Unit modes	All
Non-volatile	No

Remarks

If **UM** = 3, an automatic torque can be applied while **MO** = 1 by setting **SC**[8].

In auxiliary mode (**RM** = 1) **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.

Disabling the motor by setting **MO** = 0 or by any motor fault (**MF**) will cause a delay for 7.5 milliseconds. Any attempt to set the motor (**MO** = 1) during this time will be delayed until this time has elapsed.

The interpreter motor enable, which uses **MO** = 1, and the CANopen (EtherCAT) motor enable, which uses the DS-402 state machine, can basically live side by side, indicating the correct state regardless of the source of the command. However, mixing the two methods is not recommended.

When the motor is disabled (**MO** = 0) by the interpreter, the CANopen state is "Switch on Disable".

MO = 1 overrides the need for "Fault reset" as required by DS-402 after a fault state.



Enabling the motor

MO = 1 is the operative state of the servo drive, driving the motor and activating and executing the required motion. The software runs a set of tests to ensure that all conditions for running the motor are met.

If **MO** is set to 1 and the motor is already on, nothing happens.

When the motor is enabled, the drive reinitializes the internal parameters and motion drivers.

The drive may fail to start if the setup data is found to be inconsistent (for example, if **XM[2] < XM[1]**). In this case, the **CD** command indicates the reason for the failure.

CD may suggest to look for a more detailed error, such as **EE[]**.

During the sequence the last captured motor fault (**MF**) is reset to 0.

The motor is always started so that it does not jump. In case of Position Mode (**UM=5**), the complete position control command — which consists of the internal position command and the external position command — is set to the actual present position of the motor in order to prevent the motor from jumping.

Note that between two consecutive motor enable and motor disable calls, the motor enabling will be delayed for 7.5 milliseconds.

The SO command on motor 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 example where the motion is prevented for a long time while the motor is being set on is when a brake is defined. Brake is defined by **OL[]** which sets the relevant digital input and **BP[]** command that sets the brake time. 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. In this mode, the servo drive can perform various tasks that are impossible when the motor is on, including the following:

- Resetting the drive to default (RS command)
- Calculating and checking the integrity of the drive profiler database



- 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[M]**) 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 motor 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.

References

MF, **SR**, **CD**, **BP[]**, **EE[]**



MP[] – Motion Parameters (Reserved)

CANopen/CoE

Elmo_parameters_objects: **0x3147**

Attributes

Attribute	Description
Type	
Source	
Restrictions	
Range	
Index range	
Default	
Unit modes	
Non-volatile	

Remarks

Indices

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

Index	Description	Type	Values	Restrictions
0				
1				
2				

References



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_parameters_objects: **0x3149**

Attributes

Attribute	Description
Type	Long
Source	USB, RS232, TCP, EoE
Restrictions	According to array index
Range	According to array index
Index range (used in vectored commands)	1-4
Default	0
Unit modes	5
Non-volatile	No
Attribute	None

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.

Index	Description		Type	Default	Restrictions
1	Value	Mode Description	Long	0	UM=5
	0	Disable			
	1	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.			
	2	Point-to-Point repetitive mode, from MR[3] absolute position to MR[4] absolute position.			
	3	Point-to-Point repetitive mode, from MR[3] absolute position to MR[3]+MR[4] absolute position.			
	4	Point-to-point buffered mode			
	5	Point-to-point buffered blend mode			
2	Delay between motions (in addition to target time) Applicable only if MR[1] ==1/2/3		Long	0	0-16777215 [mSec]
3	First position, depend on MR[1] value Applicable only if MR[1] ==2/3		Long	0	
4	Second position, depend on MR[1] value Applicable only if MR[1] ==2/3		Long	0	

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_parameters_objects: **0x314A**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	None
Range	0 to 3
Index range	1 to 4
Default	2
Unit modes	1, 2, 5, 6
Non-volatile	No

Indices

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

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 (starting from version 1.1.7.14).

- 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 Depended Motion Modes and External Reference.



The motion status values are mode-dependent according to the following description.

Torque modes:

- The CANopen Profile Torque mode
- Elmo's **TC** command

MS Value	Description	Note
0	Actual torque is within the torque window (TR[5] , TR[6]) when there is not active Halt	TR[5] and TR[6] are reflected in objects 0x2E06 and 0x2E07 .
1	1. The torque command (DV[10]) has reached the torque target but actual torque is not within the torque window. 2. Halt is active and Speed command is zero	
2	The torque command differs from the torque target.	
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. 2. Halt is active and Speed command is zero.	TR[3] and TR[4] are reflected in objects 0x606D and 0x606E .
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	The velocity command has not reached the velocity target.	
3	The motor is disabled.	



Position modes:

- The CANopen Profile Position mode
- Elmo's **PA** and **PR** commands

MS Value	Description	Note
0	The actual position has reached the target within the position window (TR[1] , TR[2]) when there are no Quick Stop active and switch stop active and Reference Speed Command is zero	TR[1] and TR[2] are reflected in objects 0x6067 and 0x6068 .
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]) not has 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:

- **MS** reports 0 when the velocity command is 0 or when a software position limit is reached or when limit switches e.g. FLS\RLS are sensed
- **MS** reports 2 when the velocity command is not 0

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

External Reference (Follower, ECAM)

In external reference mode, the drive sets **MS** to 0 when the total velocity command is 0. The total velocity command is the sum of all velocity references derived from the profiler and auxiliary reference.



Notes

In time-dependent motion modes and external references, when a limit is reached (FLS\RLS or software position limit) the MS is set to 0.

References

[TC](#), [TR\[\]](#), [DV](#), [JV](#)



NF[] – Non-Linear Float

NF[] specifies the non-linear float table used for various implementations.

CANopen/CoE

Elmo_parameters_objects: **0x3157**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	See the table below.
Index Range	1 to 10
Default	Zero or see the table below.
Unit modes	See the table below.
Non-volatile	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.

Indices

The following table details the **FF[]** entries:

Index	Description	Default	Values	Restrictions
0	Reserved			
1	Specifies the current that will be used at the beginning of positive motion. In Amperes.	0	0 to PL[1]	
2	Specifies the current that will be used at the beginning of negative motion. In Amperes.	0	-PL[1] to 0	



Index	Description	Default	Values	Restrictions
3	Type of cogging compensation Bit 0 – cogging by sine wave Bit 1 – cogging by NT[N] table Bit 2 – flag to use the cogging compensation current (1 use; 0 disable)	0	0 to 6	
4	Cogging compensation sine amplitude in Ampere	0	0 to CL[1]	
5	Cogging compensation sine harmonics in electrical cycle	0	0 to 12	
6	Cogging compensation sine offset in internal electrical angle	0	-511 to 512	
7	Speed integral drain, in Ampere per seconds	0	≥ 0	Used also by object 0x2005 and GS[12]=1
8	Signal conditioner mode	0	0-2	
	0 Disabled			
	1 ZV (Zero Vibration)			
	2 ZVD (Zero Vibration, Zero Derivative)			
9	Natural (resonance) frequency in Hz	20.0	$\geq 4\text{Hz}$	
10	Damping ratio	0.1	0.001 to 0.25	

References

PL[], **GS[]**



NT – Non-Linear Table

NT specifies the non-linear table used for various implementations.

At the present time it is used for cogging compensation for each phase in one electrical cycle.

CANopen/CoE

Elmo_parameters_objects: **0x3165**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	
Range	N/A
Index range	1 to 256
Default	0
Unit modes	All but stepper mode
Non-volatile	Yes

Remarks

To change from current units (amperes) to the internal units of the **NT[N]** array, you should multiply the number of amperes in **WS[22]** and set **NF[3]** to 6.

Indices

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

Index	Description	Type	Values	Restrictions
1 to 255	Current added to control for cogging compensation	float	Internal units	Can be changed only at motor off

References

CL[N], **PL[N]**, **WS[N]**, **NF[N]**



OB[N] – Output Bits

OB[N] sets and resets a general-purpose output bit.

When **OB[N]** is queried, it reflects the **OP** command bits 0 to 31, respectively.

CANopen/CoE

Elmo_parameters_objects: **0x316D**

Attributes

Attribute	Description
Type	Parameter, 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

Remarks

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 is not defined as General Purpose. Setting a none 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. Physical level depends on the way the output is connected externally.

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 are connected to hardware PORT C differential outputs, where output 14 is the 'A' & '~A', output 15 is 'B' & '~B' and output 16 is 'Index' & '~Index'.

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 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, returns the value of digital output (14 to 16).	Long	0,1	
17	- If at least one of the digital outputs is mapped to the Amplifier OK function: - If the drive is ready to be enabled (no amplifier exception such as under voltage which prevents MO=1), OB[17] returns 1; - If 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: - If the brake is engaged, OB[18] returns 1. - If 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	
19	If at least one of the digital outputs is mapped to the Motor	Long	0, 1	



	enable/disable function: • If SO = 1, OB[19] returns 1 • If SO = 0, OB[19] returns 0 2. However, if none of the digital outputs is mapped to the Motor enable/disable function, OB[19] returns 0.			
20	1. If one of the digital output is mapped to Fault indication : • If MF is set to value different to 0, OB[20] returns 1. • If MF is set to 0, OB[20] returns 0. 2. However, if none of the digital output is mapped to Fault indication , OB[20] returns 0.	Long	0, 1	
21	1. If one of the digital output is mapped to Target Reached indication: • If MS is set to 0, OB[21] returns 1. • If MS is set to 1,2 or 3 OB[21] returns 0. (MS=0 is reflected in bit 10 of object 0x6041 Statusword) 2. 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[]**



OC[] – Output Compare

OC[N] allows the generation of pulses when predefined position is reached.

Note:

Mode is supported in GCON based drives only.

CANopen/CoE

Elmo_parameters_objects: **0x316E**

Attributes

Attribute	Description
Type	Long
Source	USB, RS232, TCP, EoE
Restrictions	According to array index
Range	According to array index
Index range	1 to 12, 21 to 32
Default	See table below
Unit modes	All
Non-volatile	No

Remarks

There are two compare modules that can be operated simultaneously:

OC[1 to 12] – Relates to compare module 1.

OC[21 to 32] – Relates to compare module 2.

Note:

The following text refers to **OC[1 to 12]**, but also applies to the **OC[21 to 32]** command unless stated differently.

The **OC** command generates a train of pulses according to a predefined sensor-position values.

The **OC[1]** enables and disables the mode and operates in three sub-mode options:

- **OC[1]=1**: Absolute 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. **PX=OC[2]+k*OC[3]**, where k is the number of successful compared occurrences (k=1,2,...)).
- **OC[1]=3**: Table position based mode. In this mode the positions are taken from a table. The table includes pairs of absolute positions. Each pair includes a “start position” – the absolute position to start the pulse, and an “end position” – the absolute position to end the pulse.



- **OC[1]=4:** Table time based mode. In this mode the positions are taken from a table. The table includes absolute positions. These positions indicate the absolute positions to start a pulse, and the duration of the pulse is indicated in **OC[4]**.
- **Setting the PX when the table is already active might cause the compare module to stop producing pulses. Make sure to set all absolute position handling (e.g. Homing or PX) prior to activation of the Compare module.**
- The following sensors are valid for compare:
 - Encoder Quad Port A sensor ID 2
 - Encoder Quad Port B sensor ID 1
 - Analog Sine\Cosine on Port B sensor ID 3
 - Serial Absolute Hiperface on Port B sensor ID 31
 - Port A emulation sensor ID 14
 - Port B emulation sensor ID 13

Notes:

“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 occasions.

When going in the direction, opposite to the specified direction with the same **OC[3]** value, the compare will occur every $0xFFFFFFFF - \mathbf{OC[3]}$ counts.

In order to prevent the output of an additional pulse during activation or deactivation of the feature, it is recommended to configure the high logic level for the output (Set **OL[i]=1**) before the **OC[]** is activated.

The various homing and capture modes (i.e. DS-402 Homing mode, DS-402 Touch-Probe, **HM[1]** or **HF[1]**) will not operate when the output comparison operates on the same sensor/feedback source. Exception to this rule is the configuring of **HM/HF** source other than Index (i.e. **HM[3]/HF[3] != 1** and **HM/HF[3] != 2**), in all the other cases output comparison and **HM/HF** can be used on the same sensor at the same time.

In the table single direction mode, the pulses are generated according to the order in the table. If the direction is changed during the comparison mode in the middle of the table, the next generated pulse will be the next position in the table before the change of direction.

In all modes, the first position must be at least **TS** time from the activation position (the drive position when sending the **OC[1]** command). If the table is to be converted, the first pulse must be at least 70[msec] from the activation position (please see Administrative Guide for more information).

In table mode, if both directions are selected, the comparison operates infinitely and will only end with the **OC[1]=0** command.



In table modes compare, the table's data is converted 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 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.

Changing sensor position during an output comparison, will **not** re-evaluate the comparison points in the compare table automatically. After sensor position change the output compare must be reinitialized to prevent offset, and if table mode is selected the compare table positions must be converted.

Output 14-16 are fast outputs allowing accurate pulse length and response time. Outputs 1-4 are slow outputs (relatively) and might distort the pulse timing, adding additional time to the pulse. In many cases this does not matter but user should be aware of the hardware abilities.

In case that the compare feedback is not AQB encoder, the none AQB feedback must be emulated to one of the AQB modules and then the compare function can be activated on this last module.

Indices

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

OC Module-0 Index	OC Module-1 Index	Description	Default Value
1	21	Write Values	0
		0 disable output compare	
		1 Accept last changes in OC[M] and enable output compare beginning at absolute position OC[2]	
		2 Reserved (returns out of range)	
		3 Accept last changes in OC[M] and enable table position based output compare mode	
		4 Accept last changes in OC[M] and enable table time based output compare 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 Sub mode depended: Absolute position sub mode: output compare function has started but absolute position has	



OC Module-0 Index	OC Module-1 Index	Description	Default Value
		not yet been reached; therefore, the train of pulses has not begun. Table position based mode: output compare function has started but the first table position has not yet been reached. Table time based mode: output compare function has started but first table position has not yet been reached	
		2 The train of pulses is being generated now	
2	22	The absolute position of the first pulse (depend on the feedback selected). Applicable only in absolute position compare mode. This value cannot exceed the modulo limit in the same direction of motion i.e. $ \mathbf{OC}[2] - \mathbf{PX} $ must be positive.	0
3	23	The hardware position intervals between subsequent pulses (in counts (PX 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.	1000
4	24	N: Pulse duration calculated in the following algorithm: If $(1 \leq N \leq 127)$ – pulse duration value is $N[\mu\text{Sec}]$. If $(127 < N \leq 253)$ – pulse duration value is: $((N - 127 + 1) * 100) [\mu\text{Sec}]$. Minimum pulse length is $1[\mu\text{Sec}]$. Maximum pulse length is $12,700[\mu\text{Sec}]$.	4
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). Not applicable in table compare modes OC[1]=3 4. Any 32 bits value is applicable	0



OC Module-0 Index	OC Module-1 Index	Description	Default Value
6	26	Output compare source signal: 0: Output compare on Position-Feedback. 1-4: Output compare on socket number (1-4). In both cases the socket must be configured as one of the following: Encoder Quad, Analog Sine/Cosine or Hiperface sensor.	0
7	27	The Output Compare uses one of the following tables to store the compare position. When the array is used for the Output compare repository, the user can edit it via GV[]\GW[] commands and not via the array primitive command. Array selection to be used as position table: 1: ZX array is used for compare, range 1-1022 (The array is used during the EAS Wizard and should be refilled after the Wizard is used) 2: NT array is used for compare, range 1-254 3: ET array is used for compare, range 1-2048 (Used when drive is in ECAM mode via EM[]) 4: UI array is used for compare, range 1-24 5: BH array is used for compare, range 1-16382 (When using this array data recording is not available. Sending start recording during this option will fail). Applicable in table modes OC[1]=3 4 . Notes: Sizes of the tables are smaller than the array's actual sizes. GV[N]/GW[N] command will limit the user. To fill the array with compare positions data, use GV[N] /GW[N] command, and do not fill the array directly with array command. NT, ET & UI are non-volatile arrays. SV command shall store these arrays. When reactivating the OC[] need to note the OC[11]/OC[31] state before.	0
8	28	Tables first position index. Applicable in table modes OC[1]=3 4 . Validated only at mode enable.	0



OC Module-0 Index	OC Module-1 Index	Description	Default Value
9	29	Tables last position index. Applicable in table modes OC[1]=3 4 . Validated only at mode enable.	0
10	30	Axis direction:	0
		0 Both directions. In this case the compare mode is infinite (the train of pulses will end only with the OC[1]=0 command).	
		1 Positive direction only.	
		2 Negative direction only.	
		Applicable in table modes OC[1]=3 4 . For details refer to the Gold Drive Administrative Manual MAN-G-ADMINING.	
11	31	Convert table positions 0 – convert table position from position counts (PX) to Hardware (sensor) units (including error mapping if enabled (PC[])), at OC[1] enable command. 1 – Do not convert the table positions from position units to Hardware units. Applicable for table modes only, i.e. OC[1]=3 4 . Note: The converted values are stored in the same table of the position which the user sets. If 0 is selected, each activation table values shall be converted. It is recommended to set 1 after the first conversion (first OC[1]=3 4). This will reduce enable mode time.	0
12	32	Number of pulses generated since mode enabled. Notes: Applicable in all modes. When mode is enabled (OC[1]!=0 command) the value is cleared (i.e. = 0). The user can change the value to any number even when compare is enabled.	0

References

GO[], **GV[]**, **GW[]**, **EA[N]**



OF[] – CAN Objects

OF[] specifies values in the CANopen object list.

CANopen/CoE

Elmo_parameters_objects: **0x3171**

Attributes

Attribute	Description
Type	Long, RW
Source	All
Restrictions	Set TW[36] =1 to get/set object. Some object can be changed on MO off only
Range	Se table below
Index range	1...51
Default	NA
Unit modes	All
Non-volatile	No

Remarks

The **OF[]** is subject to the CANopen object dictionary list. Thus, the range, type and attribute of the objects are not listed here. Refer to the CANopen manual for more details about the attributes of the objects.

All the objects are volatile objects, which are reset to their default values when the drive is booted up by NMT or power-up.

It is highly recommended not to use the **OF[]** command along with setting of the CANopen object. This will lead to ambiguous behavior.

Setting **OF[]** will update the relevant object according to the table below however when setting the object it does not set the **OF[]** array. In order to see the actual object via the **OF[]** interface, send **TW[36]=1** command to the drive.

Indices

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

Index	Object (Hex)	Description	Notes
1	6007	Abort connection code	
2	605A	Quick stop option code	



Index	Object (Hex)	Description	Notes
3	605B	Shut down option code	
4	605C	Disable operation code	
5	605D	Halt option code	
6	605E	Fault option code	
7	6060	Mode of Operation	
8	606F	Velocity threshold	
9	6070	Velocity threshold time	
10	6072	Max torque	
11	6073	Max current	
12	6075	Rated Current	
13	6076	Rated Torque	
14	607E	Polarity	MO OFF only
15	607F	Max profile velocity	
16	6080	Max motor speed	
17	6087	Torque slope	
18	60E0	Positive torque limit	
19	60E1	Negative torque limit	
20	60C5	Max Acceleration	
21	60C6	Max deceleration	
22	60C0	Interpolated position sub mode	Interpolation not active only
23		Reserved	
24	60C4.2	Interpolated data actual buffer size	
25	60C4.3	Interpolated data buffer organization	
26	6083	Profile Acceleration	
27	6084	Profile deceleration	
28	6085	Quick stop deceleration	
29	6065	Position following error	
30	6066	Position following error time	



Index	Object (Hex)	Description	Notes
31		Reserved	
32		Reserved	
33		Reserved	
34		Reserved	
35		Reserved	
36		Reserved	
37		Reserved	
38		Reserved	
39		Reserved	
40	6067	Position Window	
41	6068	Position Window Time	
42	606D	Velocity Window	
43	606E	Velocity Window Time	
44	607D.1	Low software limit	MO OFF only
45	607D.2	High software limit	MO OFF only
46	607B.1	Low Position range	MO OFF only
47	607B.2	High Position range	MO OFF only
48		Reserved	
49	Reserved (2046)	Reserved, not to the DSP, DC clock threshold	
50	2E06	Torque window	
51	2E07	Torque window time	

References

OV[] command



OL[] – Output Logic

OL[] specifies the digital output function and logic.

CANopen/CoE

Elmo_parameters_objects: **0x3177**

Attributes

Attribute	Description
Type	Parameter, Bit field, Read/Write
Source	All
Restrictions	None
Range	0 to 9
Index	1 to 16
Default	0
Unit modes	All
Non-volatile	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 given 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.



Bit-Field Entries

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 4	Function behavior * See the detailed description in Digital Output Function Description below.	Long	0	General purpose
			1	AOK function
			2	Brake
			3	Servo State (MO)
			4	Motor Fault (MF)
			5	Target Reached (MS)
			6...15	Reserved
5 to 15	Reserved			

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	Motor enable/disable indication
OL[N] = 7	High	Motor 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.



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:

- Short protection
- Overvoltage
- Overtemperature
- Overvoltage
- Safety state

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: Motor Enable

The output will be logically set in cases in which the servo is enabled. Refer to the **MO** 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 previous 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 DS402 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. In cases that follower/ECAM is active (**EM[1]**, **RM**, **EI** commands), this output indicates the *In-Position* state.

Notes

- 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]**) 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.

References

OP, **OB[N]**, **MO**, **OC[N]**, **GO[N]**, **EA[N]**



OP – Output Port

OP specifies the digital output port.

CANopen/CoE

Elmo_parameters_objects: **0x317B**

Attributes

Attribute	Description	
Type	Long, Read/Write	
Source	All	
Range	Bit 0	General-purpose output 1 level
	Bit 1	General-purpose output 2 level
	Bit 2	General-purpose output 3 level
	Bit 3	General-purpose output 4 level
	Bit 4 – 12	Reserved
	Bit 13-15	General-purpose output 13-15 level
	Bit 16	Amplifier OK output indication
	Bit 17	Break output indication
	Bit 18	MO ON output indication
	Bit 19	Motor Fault indication
	Bit 20	Target Reached (In Position) indication
	Bit 21-31	Reserved
Default	0	
Unit modes	All	
Non-volatile	Yes	

Remarks

Refer to **OL[]** command for detailed information about the output optional functions.

Outputs 14, 15 & 16 are connected to hardware PORT C differential outputs, where output 15 is the 'A'&'~A', output 16 is 'B'&'~B' and output 17 is 'Index'&'~Index'. These outputs can be used for General-purpose functions only.

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.

Gold 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 **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]



OV[] – Set CANopen Objects

OV[] specifies alias values in the CANopen object list.

It is mainly used during test procedures.

CANopen/CoE

Elmo_parameters_objects: **0x3181**

This command is mainly used during test procedures.

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	See the table below.
Index range	1 to 65
Default	N/A
Unit modes	All
Non-volatile	No

Remarks

The **OV[]** is subject to the CANopen object dictionary list. Thus, the range, type and attribute of the objects are not listed here. Refer to the CANopen manual for more details about the attributes of the objects.

All the objects are volatile objects, which are reset to their default values when the drive is booted up by NMT or power-up.

It is highly recommended not to use the **OV[]** command along with setting of the CANopen object. This will lead to ambiguous behavior.

Setting **OV[]** will update the relevant object according to the table below (if the object is not read only) , setting the object will update the **OV[]** array.

Indices

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

Reserved and unused entries will return 0 and do nothing.

Index	Object (hex)	Description	Notes
1	603F	Read Error Code DS402	



Index	Object (hex)	Description	Notes
2	6061	Read Mode of operation display	
3	6064	Read actual position in User Units	
4	6069	Read the actual speed in counts/sec	Similar to VX . Value can be read from Speed sensor or Position sensor. See 606A.
5	606A	Velocity Sensor selection code. Selects between Position or Velocity sensor	
6	606B	Speed demand value in counts/sec	The output of the profiler converted to counts/sec.
7	606C	Read the actual velocity in User Units	Value can be read from Speed sensor or Position sensor. See 606A.
8	6081	Read, Write Profile Velocity used in Profile position mode	Object is override by SP value when Elmo's Begin Motion (BG) is set.
9	60B0	Read, Write Position Offset for Cyclic Synchronous Position (CSP) mode	
10	60B1	Read, Write Velocity Offset for CSP and Cyclic Synchronous Velocity (CSV) mode	
11	60B2	Read, Write Torque Offset for CSP, CSV and Cyclic Synchronous Torque (CST) mode.	
12	60B8	Read, Write Touch Probe configuration function	
13	60B9	Read Touch Probe Status	
14	60BA	Read Touch Probe 1 Positive edge capture	In User Units
15	60BB	Read Touch Probe 1 Negative edge capture	In User Units
16	60BC	Read Touch Probe 2 Positive edge capture	In User Units
17	60BD	Read Touch Probe 2 Negative edge capture	



Index	Object (hex)	Description	Notes
18	2E10	Touch Probe Home	
19	60C1.1	Read, Write Interpolated Data Record Sub index 1, used in Interpolated Position mode (IP)	Typically the position set point for IP mode
20	60C1.2	Read, Write Interpolated Data Record Sub index 2, used in Interpolated Position mode (IP)	Typically the velocity set point for IP mode
21	Reserved		
22	Reserved		
23	60C2.1	Read, Write Interpolated Time period for IP mode and cyclic synchronous modes	
24	60C2.2	Read, Write Interpolation Time index for IP mode and cyclic synchronous modes	
25	60C4.4	Read, Write Buffer Position in Interpolation data configuration object. Points where the next set point shall be written to in a ring buffer organization. Used in IP mode	
26	60C4.5	Read, Write Size of data record in bytes. Used IP mode.	
27	60C4.6	Write Clears and allows the set point buffer of the IP mode.	
28	60F2	Read, Write Positioning option code. Used in position control modes to determine the behavior of the relative position, modulo and set point ACK in Profile Position mode.	
29	60FA	Read Control Effort	Returns 0: reserved for this object.



Index	Object (hex)	Description	Notes
30	60FC	Position Demand Value internal units	Value is the actual entry to the control loop including real time Auxiliary Position w/o modulo calculations.
31	60FD/20FD	Digital input/Clear digital inputs	
32	607A	Target Position. Used in Profile Position mode and CSP mode.	
33	Reserved		
34	6082	End Velocity used in Profile Position mode	Object is override by FS when Elmo's Begin Motion (BG) is set
35	6098	Homing Method	
36	6099.1	Homing Speed during search of switch	Typically fast speed when looking for edge or limit
37	6099.2	Homing Speed during search of zero	Typically slow speed when searching for the final homing location
38	609A	Homing Acceleration	Used for Deceleration when required
39	607C	Homing offset	
40	Reserved		
41	Reserved		
42	60F4	Position Following Error in User Unit	
43	60FF	Read, Write Target Velocity. Used in Profile Velocity mode.	
44	6071	Reserved for Target torque	
45	Reserved		
46	6077	Read Torque Actual Value	Torque units are in 1/1000 of rated torque (6076)
47	6078	Read Current Actual Value	Current units are in 1/1000 of rated current (6075)
48	2E00	Reserved for Position Compensation Index	
49	2E0A	Reserved for Torque Compensation	



Index	Object (hex)	Description	Notes
		value	
50	6062	Read Position Demand value by User Unit	
51	6079	Read DC link voltage. Bus voltage in millivolts	
52	2060	The non-volatile parameters checksum is retrieved via this object (OV[52]).	The checksum is updated during power cycle, SV and LD commands.
53	20A0	Read Additional Position	
54	20B0.8	Homing Sensor selection	
55	2020.1	Homing Torque Limit	
56	2020.2	Homing Time Limit	
57	2020.3	Homing Position Limit	
58	60FE.1	Physical outputs	Get function can be used
59	60FE.2	Bit mask	Get function can be used
60	2082	CAN Controller status	
61	2085	Extra Status Register	
62	2086	Safe Torque Off status register	Value includes the detailed STO diagnostics status.
63	2F75	Extrapolation time out (IP & cyclic)	
64	2020.4	Detection Velocity Limit	
65	2020.5	Detection Velocity Time Limit	

References

OF[], **BG**



PA[N] – Position Absolute

PA[N] specifies the absolute target positions for the drive profilers.

CANopen/CoE

Elmo_parameters_objects: **0x3186**

PA[1] Target_position: **0x607A**

Attributes

Attribute	Description
Type	Long, 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^{31} to $(2^{31}-1)$
Index range	1 to 4
Default	0
Unit modes	UM = 3; UM = 5
Non-volatile	No

Indices

The following table describes the profilers defined by the command index (**N**).

Index	Profiler name	Profiler description
1	Main	The main drive profiler (DS402)
2	Superimposed	The profiler 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 profiler is used during follower and ECAM mode to create a smooth phase shift in the master position of a slave axis.
4	Socket profiler	Reference to a socket in real time level, can be used for all functions of the socket.



Remarks

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 motion mode will change to the Point-2-Point (PTP) motion mode according to **AC[1]**, **DC**, **SF**, **SP[1]** and **FS**.

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]**.

When **PA[1]** is set and **BG[1]** is commanded, the motion mode reflected in **OV[2]** (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.

Motion is limited by software limits. Motion will stop at these limits.

If **UM** = 3, **PA[1]** determines the target position in electrical angle units (1 pole pair is 360 electrical degrees, which are denoted by 512 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.

References

JV, **FS**, **SP**, **AC**, **DC**, **UM**, **PR**, **PO**, **FC**



PB – PAL Burn

PB reports the burned PAL number.

Changing the PAL is required in case of special features, such as special encoders implementation.

Note: Some of the drives do not support PAL burn. Refer to the **PB[]** description table in this command.

CANopen/CoE

Elmo_parameters_objects: **0x3187**

Attributes

Attribute	Description
Type	Signed Long
Source	USB, TCP, EoE, RS232
Restrictions	N/A
Range	N/A
Default	1
Unit modes	All
Non-volatile	Yes

Remarks

The **PB** command reports current burned PAL with the user-selected PAL in the drive.

PAL burning sequence:

- Start downloading PAL via EAS software or via EtherCAT-FoE protocol.
- After PAL is burned the PB reports -1, until the drive is rebooted.
- When the process is completed successfully, the **PB** command returns the number of the PAL burned in the drive. If the process fails, the **PB** command returns -1, and the drive will not allow the user to perform the following:
 - The user program auto-exe will not be performed at power-up.
 - The user program cannot be executed.
 - The servo cannot be turned on.

Using a PAL burn with a version that is not supported by the firmware will prevent the motor from being enabled.

Power-up must be performed after the burn is completed.



The following is the description table:

Value	Description	PAL Description		
		Gold Guitar (WS[8]==0)	Gold Whistle/ Gold Trombone Rev-A (WS[8]==1)	Gold Whistle/ Gold Trombone Rev-C (WS[8]==2)
-1	PAL is not burned or after new PAL was downloaded (reboot needed). The servo cannot be set (MO = 1), and the user program is disabled.	N/A	N/A	N/A
1	PAL #1 is burned	Port A: Biss Port B: None absolute	PAL Version 20 (Default)	PAL Ver 42.1 (Default) Base Biss EnDat SSI Panaasonic Tamagawa
2	PAL #2 is burned	Port A: Panasonic, Mitutoyo , Tamagawa Port B: None Absolute OR Port A: None Absolute Port B: None Absolute	N/A	PAL Ver 42.2 Base Sanyo/Nikon
3	PAL#3 is burned	Port A: EnDat Port B: None absolute OR Port A: None Absolute Port B: None Absolute	N/A	N/A
0x5AA5	The PAL cannot be replaced.	The actual PAL version was burned during manufacturing.		

References

VP, WS



PC[N] – Error Mapping

PC[N] enables the configuration and operation of error mapping.

CANopen/CoE

Elmo_parameters_objects: **0x3188**

Attributes

Attribute	Description
Type	Long
Source	USB, RS232, TCP, EoE
Restrictions	According to array index
Range	According to array index
Index range	1 to 8
Default	PC[3] = 2
Unit modes	All
Non-volatile	Yes
Attribute	None

Remarks

Error 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 socket position is taken as an absolute entry in the table. The mode can also be activated as a cyclic mode (modulo), where the socket 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. These values can be treated as values in position counts (i.e. - **FP[x]** or **PX** counts), in which case a conversion procedure is executed, or as values in sensor units, in which case no conversion is calculated.

For more details on different error mapping options, please refer to the “Error mapping user manual” document.

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)
- 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 **GP[]** command.

The overall corrected position (abscissa + 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:

$$\text{Corrected Position} = \text{Actual Position} + \text{ErrorCorrection}$$

This limitation is not checked by the drive before enabling error mapping. It is up to the user to verify this.

Indices

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

Index	Description		Type	Default	Restrictions
1	Value	Operation	Long	0	See notes below.
	0	Disable			
	1	Enable linear mode without converting correction table values from user position counts (PX or FP[x]) to the sensor units (hardware). In this mode it is assumed that the position correction values are in sensor units.			
	2	Enable cyclic (modulo) mode without converting correction table values from position counts (FP[x] or PX) to sensor units (hardware counts). In this mode it is assumed that the position correction values are in sensor units.			
	3	Enable linear mode with conversion of the correction table values from position counts (PX or FP[x]) to sensor units (hardware).			



Index	Description		Type	Default	Restrictions
	4	Enable cyclic (modulo) mode with conversion of the correction table values from position counts (FP[x] or PX) to sensor units (hardware).			
2	Socket number to which error mapping will be applied		Long	1	1 to 4
3	Value	Direction	Long	2	1 to 3
	1	The NT array is selected as the correction table. The range is from 1 to 255.			
	2	The ET array is used as the correction table. The range is from 1 to 2048.			
	3	The UI array is used as the correction table. The range is from 1 to 24.			
4	Low index of the correction table		Long	1	≥ 1
5	High index of the correction table		Long	2	$\leq$ Table size
6	Correction table position grid of size 2^N , where $3 \leq N \leq 31$		Long	3	$3 \leq N \leq 31$
7	Error mapping start position (in sensor units)		Long	0	
8	N-Modulo value, where the modulo is between 0 and N		Long	100	Positive value ≤ 1

References

[GP\[\]](#)



PE – Position Error

PE reports the position error.

CANopen/CoE

Elmo_parameters_objects: **0x318A**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	None
Range	0 to $2^{31} - 1$
Default	0
Unit modes	UM = 5
Non-volatile	No

Remarks

The **PE** command returns the instantaneous position tracking error, in counts.

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] – Peak Limit

PL[] specifies the peak limit current peak limit duration.

CANopen/CoE

Elmo_parameters_objects: **0x3191**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	PL[1]: 0 to MC PL[2]: 0.1 to 30
Index range	PL[1]: Peak Current Limit in Amperes PL[2]: Peak Current Duration in Seconds
Default	PL[1] = 0, PL[2] = 3 (RS)
Unit modes	All
Non-volatile	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 **lcmd=0** to **lcmd=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, the continuous current limit **CL[1]** can be set as follows:
0 <=CL[1]<= PL[1] <= MC.



Indices

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

Index	Description	Units	Range
0	Reserved		
1	Defines the motor maximum peak current, in amperes.	Amperes	0 to MC
2	Defines the motor maximum peak duration, in seconds.	Seconds	0.1 to 30

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 (defined by $I^2 \times T = MC^2 \times 3$):

$$PL[2] \text{ (sec)} \leq 3 \text{ (sec)} \times MC^2 / PL[1]^2$$

In order to protect the motor, the user should calculate the value of $(PL[1])^2 \times PL[2]$ such that it will not exceed the overload value allowed by the motor manufacturer.

Note also, that if, prior to the high current demand, the actual average current was **> 0**, the actual peak limit duration will be shorter than the specified **PL[2]**.

References

CL[], **LC**, **MC**, **TC**



PO – Positioning Options

PO specifies the optional behavior of the **PA** and **PR** commands.

CANopen/CoE

Elmo_parameters_objects: **0x3194**

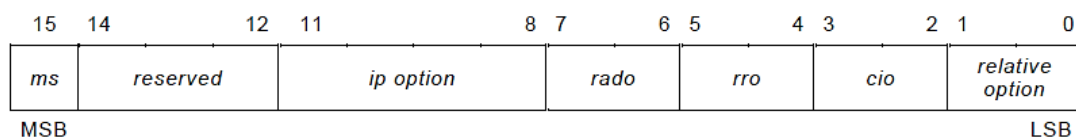
Positioning_Option_Code: **0x60F2**

Attributes

Attribute	Description
Type	Unsigned Long, Read/Write
Source	All
Restrictions	Effective on the next call to BG
Range	According to bit-field definitions
Default	0
Unit modes	UM = 5
Non-volatile	Yes

Remarks

The following diagram shows the bit-field structure of a **PO** 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** 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[N] – Protocol

PP[] programs all communication parameters for the RS232 and CANOpen protocols.

CANopen/CoE

Elmo_parameters_objects: **0x3195**

Attributes

Attribute	Description
Type	Parameter, Long
Source	RS232, CANOpen
Restrictions	MO = 0 for PP[1]
Index range	1 to 15
Default	See the table below.
Unit modes	All
Non-volatile	Yes

Remarks

Indices

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

Index	Description	Type	Values		Restrictions
1	Type of Communication	Long	1	RS232	PP[1] serves as “Enter Communication Parameters” for RS-232. PP[2] and PP[4] come into effect only when PP[1] is written. The response to PP[1]= # is not the same as the response to all other commands, because the communication type switches while processing the command.
			2	RSVD	



2	RS232 Baud rate	Long	0	4,800	This parameter has no immediate effect.
			1	9,600	
			2	19,200	
			3	38,400	
			4	57,600	
			5	115,200 (default)	
3	RSVD				
4	RS232 parity	Long	0	None (default)	
			1	Even	
			2	Odd	
5–12	RSVD				
13	CANOpen device ID	Long	1 to 127		The default is 127.
14	CANOpen Baud rate	Long	0	1,000,000	
			1	500,000 (default)	
			2	250,000	
			3	125,000	
			4	100,000	
			5 to 7	50,000	
15	CANOpen group	Long	1 to 128		The default is 128.

Notes

- The number of RS232 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.
- Unused **PP[N]** parameters are reserved for compatibility with other Elmo drives.

References



PR[N] – Position Relative

PR[N] specifies the relative target positions for the drive profilers.

CANopen/CoE

PR[1] Target_position: **0x607A**Elmo_parameters_objects: **0x3197**

Attributes

Attribute	Description
Type	Long, 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^{31} to $(2^{31}-1)$
Index range	1 to 4
Default	0
Unit modes	UM = 3; UM = 5
Non-volatile	No

Indices

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

Remarks

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 be switched to the position control loop.
- the drive motion mode will be changed to the Point-2-Point (PTP) motion mode according to **AC[1]**, **DC**, **SF**, **SP[1]** and **FS**.

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]**.

When **PR[1]** is set and **BG[1]** is commanded, the motion mode reflected in **OV[2]** (object **0x6061**) is modified to 1 (Profile Position).

Object **0x607A** will be updated according to the **PR[1]** value. Refer to the **BG** command for more details.



Motion is limited by software limits. Motion will stop at these limits.

If **UM** = 3, **PR[1]** determines the target position in electrical angle units (1 pole pair is 360 electrical degrees, which are denoted by 512 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.

References

JV, FS, SP, AC, DC, UM, PA, PO, FC



PS – Get Program Status

PS reports the status of the user program.

CANopen/CoE

Elmo_parameters_objects: **0x3198**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All, except the user program
Range	-2 to 1 (See the table below.)
Unit modes	All

Remarks

A drive can be in one of the following situations with respect to the user program:

1. No program was ever loaded, or no program exists.
2. The user program was downloaded. It is not running, but it is ready to execute a command.
3. The user program was halted by the **HP** command.
4. The user program is currently running.

PS indicates one of the above descriptions. The following table details these indications:

PS value	Description
-2	The program has not been compiled or does not exist. This would also be the indication in cases in which the program exists but could not be loaded from the non-volatile flash memory during boot-up (power-up or drive reset).
-1	The program exists, is at rest, and is ready to be executed.
0	The program is in the halted state. This can be either a break point (in debug mode) or after HP command.
1	The program is running. This case would also be indicated in SR command bit 12.

References

CC



PT[] – Position Table (Reserved)

CANopen/CoE

Elmo_parameters_objects: **0x3199**

Attributes

Attribute	Description
Type	
Source	
Restrictions	
Range	
Index range	
Default	
Unit modes	
Non-volatile	

Remarks

Indices

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

Index	Description	Type	Values	Restrictions
0				
1				
2				

References



PU – Main Position in User-Defined Units

PU reports the present position of the position socket in user-defined units.

CANopen/CoE

Position_actual_value: **0x6064**Elmo_parameters_objects: **0x319A**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	None
Range	-2^{31} to $2^{31} - 1$
Default	According to the value of the sensor. An incremental encoder starts counting from zero.
Unit modes	All
Non-volatile	No

Remarks

PU returns the position of the position socket in user units after applying the position scaling factor (**FC[]**) and the position modulo (**XM[]**).

References

PX, **FC[]**, **XM[]**



PV – Position Velocity Time setting (Reserved)

PV[]

CANopen/CoE

Elmo_parameters_objects: **0x319B**

Attributes

Attribute	Description
Type	
Source	
Restrictions	
Range	
Index range	
Default	
Unit modes	
Non-volatile	

Remarks

Indices

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

Index	Description	Type	Values	Restrictions
0				
1				
2				

References



PX – Main Position in Counts

PX reports the present position of the position socket in internal units (counts).

CANopen/CoE

Elmo_parameters_objects: **0x319D**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	-2^{31} to $2^{31} - 1$
Default	According to the value of the sensor. An incremental encoder starts counting from zero.
Unit modes	All
Non-volatile	No

Remarks

The position sensor must be set by pointing to the relevant sensor (**CA[45]**) and setting the sensor **CA[41-44]**.

References

FP[], OV[], CA[], PU



PY – Auxiliary Position in Counts

PY sets/ reports the present position of the external (auxiliary) reference in counts.

CANopen/CoE

Auxiliary_position_actual_value: **0x20A0**Elmo_parameters_objects: **0x319E**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	-2^{31} to $2^{31} - 1$
Default	According to the value of the connected sensor/profile.
Unit modes	All
Non-volatile	No

Remarks

The auxiliary position **PY** can be referenced by pointing to:

- the relevant sensor (**CA[79]**) set by **CA[41-44]**, if **EM[11]:bit3=0**.
- the main profiler (including CAN Encoder) output, if **EM[11]:bit3=1**.

PY counts cyclically (refer to the **YM[N]** command).

If the **PY** value setting is outside the range [**YM[1]...YM[2]**], the requested value will be ignored and **PY** will not change

References

YM[N], **EM[N]**, **CA[N]**



RC – Recorder Variables

RC specifies which of the mapped signals are to be recorded.

CANopen/CoE

Elmo_parameters_objects: **0x31BC**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All, except CoE
Restrictions	Recorder inactive (RR=0 or RR=-1)
Range	Bit field [bits 0 to 15]
Default	0
Unit modes	All
Non-volatile	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 sixteen variables that are to be recorded.

The host can map many optional variables to any bit of **RC** from bit 0 to bit 15.

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



RG – Recorder GAP

RG specifies the frequency per sampling time that the recorder is activated.

CANopen/CoE

Elmo_parameters_objects: **0x31C0**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All, except CoE
Restrictions	The recorder must be inactive (RR = 0 or RR = -1).
Range	1 to 65535
Default	1
Unit modes	All
Non-volatile	No

Remarks

Because 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** value and the specific unit mode (**UM**).

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**



RL – Recorder Length

RL specifies the total length of the recorded data.

CANopen/CoE

Elmo_parameters_objects: **0x31C5**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All, except CoE
Restrictions	The recorder must be inactive (RR = 0 or RR = -1).
Range	1 to 16384
Default	16384
Unit modes	All
Non-volatile	No

Remarks

If the recorder is set with a length larger than the maximum value (16K), each signal will be set according to the formula **RL**(16K)/number of vectors.

For example, in the case of **RL** = 16384, the maximum recorder length for each signal will be as follows:

Number of recorder signals	Maximum recorder length per signal
1	16,384
2	8,192
...	
8	2048
9	1820
10	1638
...	
15	1092
16	1024



The actual size of the recorded data is returned by the **WI[21]** command.

If the drive has stored a previously recorded data vector, setting **RL** will invalidate this data.
Invalidated data cannot be retrieved

References

RC, RG, RP[N], RR, BH



RM – Reference Mode

RM specifies the use of an external reference generator (ECAM/Follower).

CANopen/CoE

Elmo_parameters_objects: **0x31C6**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All, except CoE
Restrictions	None
Range	0, 1
Default	0
Unit modes	All
Non-volatile	Yes

Remarks

RM also specifies whether the motor can be enabled by using the Inhibit/Enable function of a digital input. Refer to the **IL** command for more details.

The following table describes the possible values of the **RM** command.

Command Value	Description
0	External reference generator is disabled. If a digital input is assigned to the Inhibit /Enable function, the motor will not be enabled by the switch.
1	External reference generator is enabled. If a digital input is assigned to the Inhibit /Enable function, the motor will be enabled by the switch.

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[] enables complete specification of how the recorder is triggered and how the recorded data is transferred to the host.

CANopen/CoE

Elmo_parameters_objects: **0x31C9**

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 15
Default	According to the description table
Unit modes	All
Non-volatile	No

Remarks

Trigger definitions:

The recorder is started by a trigger event, which may be one of the following:

- Immediate:*
The recorder starts immediately after the recording request is issued.
- Triggered by an analog signal:*
The recorder starts upon one of the following events:
 - Positive slope:*
The signal crosses a prescribed level with a positive slope.
 - Negative slope:*
The signal crosses a prescribed level with a negative slope.
 - Window:*
The 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:**
The recorder starts after a requested time (in milliseconds) is reached.
- **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 when the recorder is started. The recorder can be programmed to start before the trigger event, so that the trigger event can be caught “in the middle of the action.” This is possible because the recorder starts to record at the instant when it is launched by the **RR** command, so that when the trigger event occurs, the pre-trigger information is already recorded.

Indices

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

Index	Description	Type	Values		Restrictions
0	Time quantum base	Long	0	Time quantum is 2*TS	
			1	Time quantum is TS	
1	Trigger variable, which is defined similarly to RC , but only 1 bit may be non-zero. The trigger variable does not need to be one of the recorded variables.	Long	1 to 65535		
2	Pre-trigger, the percentage of the recorded signal that is recorded before the trigger event [%]	Long	0 to 100		



3	Trigger type	Long	0	Immediate	
			1	Analog signal	
			2	Positive slope	
			3	Negative slope	
			4	Window	
			5	Reserved	
			6	Reserved	
			7	Digital input	
			8	Motion begin	
4	Level 1 for a positive-slope trigger, or the high side for a window trigger	Long			
5	Level 2 for a negative-slope trigger, or the low side for a window trigger.	Long			
6	Level of digital input when used as trigger for the recorder	Long	1 to 0xFFFFFFFF		
7	Digital input mask, defines which digital inputs trigger the recorder	Long	1 to 0xFFFFFFFF		
8	Lower buffer index for recorded data upload transmission	Long	0 to 16384		When RP[9] = RP[8] = 0, all of the buffer is transmitted.
9	Higher buffer index for recorded data upload transmission.	Long	0 to 16384		
10	Time value for start recording [msec]	Long	Reserved		Used only if RP[4] = 5
11	Selected recorded signal for the BS[] command. Defined similarly to RC , but only 1 bit may be non-zero.	Long	1 to 65535		Recorder ready (RR ==0); Selected signal is recorded.
12	Reserved				
13	Reserved				
14	Reserved				
15	Reserved				



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]**).

References

[RC](#), [RG](#), [RL](#), [RR](#), [BH](#), [BS\[\]](#)



RR – Activate Recorder / Recorder Status

RR launches the recorder, kills an on-going recording process or retrieves the recorder status.

CANopen/CoE

Elmo_parameters_objects: **0x31CB**

Attributes

Attribute	Description
Type	Parameter, Long
Source	All, except CoE and Program
Restrictions	The recorder must be inactive (RR = 0 or RR = -1)
Range	0 to 3
Default	-1
Unit modes	All
Non-volatile	No

Remarks

The **RR** command has the following options:

Value	Description
0	Kill the recorder
1	Start recording upon the next BG command
2	Start recording immediately
3	Arm the recorder with the trigger setting of the RP[3] command
4	Reserved

The **RR** command may report the following values:

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.
1 to 3	The recorder is waiting for completion of the trigger event, respectively.



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 – Soft Reset

RS initializes the drive parameters to their factory defaults and resets all volatile variables to their power-on defaults.

CANopen/CoE

Elmo_parameters_objects: **0x31CC**

Attributes

Attribute	Description
Type	Command
Source	All, except CoE
Restrictions	MO = 0. The user program must be inactive (PS=-1 or -2).
Unit modes	All
Non-volatile	No

Remarks

RS does not change the communication settings; therefore, after executing **RS**, it is still possible to communicate with the drive. The communication parameters, however, are reset.

The **RS** command disables the communication routines for a few milliseconds. If **RS** is executed by a USB command, the EtherCAT message may be lost in the execution interval.

The **RS** command modifies only the RAM contents; it does not affect the flash memory.

If necessary, use the **SV** command to store the default **RS** permanently.

The default parameters are designed so that after **RS** is enabled, the motor should not produce a torque command. Normally, the Motor On command should return an error.

References

LD, **SV**



RV[N] – Recorder Variables

RV[N] maps recorded variables to the recorder through the **RC** command. By setting **RV[N] = X**, the variable *X* is assigned to bit *N* - 1 of **RC** in the variable static table. The default mapping (power on) of **RV[1]** to **RV[16]** 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 *Gold* or *SimplIQ* drive and can be uploaded using the **LS** command.

CANopen/CoE

Elmo_parameters_objects: **0x31CF**

Attributes

Attribute	Description
Type	Long, Bit field
Source	Program, all except CoE
Restrictions	The recorder must be inactive.
Range	According to the index in the static variable table
Index range	1 to 16
Default	1 to 16, respectively
Unit modes	All
Non-volatile	Yes

Remarks

Text

Index	Description	Type	Values	Restrictions
1 to 16	Recorder variable entry	Long	1 to 16, respectively	None

References

RC, **BH**, **RR**



SC[N] – Stepper Commutation

SC[N] gets and sets parameters used for the stepper, binary search and Gurley commutation algorithms.

CANopen/CoE

Elmo_parameters_objects: **0x31D6**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	None
Range	Define in Table
Index Range	1 to 13
Default	See the table below.
Unit Modes	All modes (except for SC[8] , used only in stepper mode)
Non-volatile	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, 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 gets valid, then wait **SC[3]** seconds. At this point the commutation phase is known and saved, the unit mode changes to the original mode that was 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 Gurley commutation
2	Time to increase from 0 to the desired current	Seconds	1.0	0.001 to 12	Stepper, binary search and Gurley 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. Gurley commutation: After the algorithm finds the angle, it waits for SC[3] seconds before completing the commutation.	Seconds	1.0	0.001 to 6	Stepper, binary search and Gurley commutation
4	Reserved for compatibility		1.0	Reserved for compatibility	
5	Stepper commutation: Low speed defined as motor not moving. Gurley commutation: Stepper speed during the commutation	Electrical cycles/seconds	1.0	0.01 to 10	Stepper and Gurley commutation



Index	Description	Unit	Default	Range	In Process
6	Threshold phase for stepper method	Electrical angle/512	5.0	1 to 43	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 10	Stepper commutation
12	Binary search verification		0		Binary search commutation
	0	Disabled			
	1	Enabled			
13	Unbalanced axis commutation		0		Stepper commutation
	0	Disabled			
	1	Enabled			

References

PL[], **CA[]**, **MO**, **UM**



SD – Stop Deceleration

SD[N] specifies the deceleration used during emergency stops.

CANopen/CoE

Elmo_parameters_objects: **0x31D7**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	1.....(2^{31})-1
Default	1e9
Unit modes	All
Non-volatile	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[]**,



SE[N] – Sine Excitation

SE[N] generates an internal signal command as a reference to the controller.

CANopen/CoE

Elmo_parameters_objects: **0x31D8**

Attributes

Attribute	Description
Type	Float
Source	All
Restrictions	None
Range	According to the SE[] table below
Index range	1 to 7
Default	According to the SE[] table below
Unit modes	All
Non-volatile	No

Remarks

Set a virtual sensor. The sine excitation generates a signal that consists of three elements:

- Two frequencies sinus signals denoted as A amplitude and B amplitude,
- An offset **DC** signal denote as **DC** value.

These signals can be used as position, velocity or current for reference, emulation or any other general purpose according to SE[1].

The units are counts, counts/seconds or amperes, respectively.

Setting **TW[80]** = 1 activates the signal (can be changed on the fly). It first ramps to the **DC** value, and then builds the sine waves.

Setting **TW[80]** = 0 stops the signals. It first stops the high and low sine, and then the **DC** offset ramps to zero according to SE[7].

The sine excitation has a function ID of 8, and it must be mapped to a socket. For example, to map socket 4 for sine excitation set **CA[44]** = 8.

The socket then can be used to any other function that can operate on sockets.

When the command is used for sine excitation, it is directed to the relevant control loop (according to **CA[68, 69, 70]**) and is not limited by the Stop Manager limits and no limits of command is checked.

The function starts to work regardless of the presence of a brake in the system.



Indices

Following table describes the **SE[N]** entries:

Index	Description	Units	Default	Values	Notes
1	Units of virtual sensor	1. Current [ampere] 2. Velocity [counts/sec] 3. Position [counts]	1	1 to 3	
2	A amplitude	Depends on input For example: Position: 1000 counts Velocity: 10,000.0 counts/sec Current: 0.25 amperes	0	N/A	Values can be positive or negative. Drive does not limit the entry.
3	A frequency	Hz	100	0 to 5000	
4	B amplitude	Same units as A amplitude	0	N/A	Same as A amplitude
5	B frequency	Hz	0	0 to 5000	
6	DC Value	Same units as A amplitude	0	N/A	Same as A amplitude
7	Slope to DC value	Same as DC value/sec	0	N/A	The time that takes to ramp from 0 to DC carrier value or to 0 when mode is disabled
8	Convert the sine signal to pulses. When this flag is on, the output is a square wave with the same frequency (at A amplitude)		0	0, 1	

References

TW[N], CA[N]



SF – Smooth Factor

SF specifies the motion smoothing factor, in milliseconds, for PTP and jogging.

CANopen/CoE

Elmo_parameters_objects: **0x31D9**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	0 to 63
Default	0
Unit modes	All
Non-volatile	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 effect the motion in moderate portions. When **SF**=0, all the requested acceleration is used by the profiler to build the speed command.

A total of 63 msec is allowed, this is actually limited by the moving average buffer of 255 entries of 250 µsec each.

References

AC, DC, PA, PR, JV, JP



SN[N] – Serial Number Identity

SN[N] obtains the identity number according to the DS-301 reflecting object 0x1018. The object is relevant to CANopen and EtherCAT users.

CANopen/CoE

Object: **0x1018**

Elmo_parameters_objects: **0x31E1**

Attributes

Attribute	Description
Type	Integer, Read Only
Source	All
Restrictions	None
Range	Define in Table
Index Range	1 to 4
Default	See the table below.
Unit Modes	All modes
Non-volatile	Yes

Remarks



Indices

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

Ind	Description	Value	Comment
1	Vendor ID	0x9A	Constant value relevant for CANopen and EtherCAT and use to identify “Elmo” as the vendor.
2	Product code	Identify the Gold drive product. CANopen: 0x030923 EtherCAT: 0x30924 (No ECAT switches) EtherCAT: 0x30925 (with ECAT address switches)	Note that the Product Code does not distinguish between the drive’s name e.g. G-TWI, G-GUI etc.
3	Revision number	The stack revision number. (example) CANopen: 1 EtherCAT: 0x00010400	The rev number depends on the firmware version.
4	Serial number	Drive serial number as in the drive plate.	e.g.: 10471921

References



SO – Servo Enabled

Refer to **MO/SO** in this manual.

CANopen/CoE

When **SO** is set to 1, the Operation Enable state is reported in object Statusword **0x6041**

Elmo_parameters_objects: **0x31E2**

Attributes

Attribute	Description
Type	Long, Read
Source	All
Restrictions	None
Range	0 1 (when the drive is ready to handle a profiler set point)
Default	
Unit modes	N/A
Non-volatile	No

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_parameters_objects: **0x31E3**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	Effective on the next call to BG[N]
Range	0.....(2^{31})-1
Index range	1 to 4
Default	100000
Unit modes	UM = 5
Non-volatile	Yes

Indices

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

Remarks

SP[N] is used during PTP motion to limit the speed of the profiler. In cases in which the **SP[N]** value is too high for the position target, a triangular motion will be performed by the profiler. If the **SP[N]** value is low enough, a trapezoidal motion will be performed.

On **BG[1]**, **SP[1]** overrides CANopen object **0x6081**.

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

References

AC, **DC**, **PA**, **BG**, **FS**, **FC**



SR – Status Register

SR reports the status of different functions in the drive. It returns a snapshot of the system 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.

CANopen/CoE

Manufacturer_status_register Object **0x1002** returns the same function.

Note: Object **0x6041** as well as **SW** command return the status with respect to DS-402 functionalities.

Elmo_parameters_objects: **0x31E5**

Attributes

Attribute	Description
Type	Read, Bit-field
Source	All
Restrictions	None
Range	None
Default	None
Unit modes	All
Non-volatile	No

Remarks

The following table details the different functions in the bit-field format.

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 Red LED is set, and the AOK function is active. 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.	.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	



Bits	Description	Values		Notes
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	In 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 OV[2] . Profilers are activated after BG or according to DS-402 profiling method.
		1	Profile position mode (PTP)	
		2	N/A	
		3	Profile Velocity mode (JV)	
		4	Profile Torque mode (TC)	
		5	N/A	
		6	Homing mode (DS-402 only)	
		7	Interpolated position mode (DS-402 only)	
		8	Cyclic sync position mode (DS-402 only)	
		9	Cyclic sync velocity mode (DS-402 only)	
		10	Cyclic sync torque mode (DS-402 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.	



Bits	Description	Values		Notes
13	Current Limit is on	0	No Current Limit, Peak current (PL[1]) can be applied.	
		1	Current is limited to CL[1] .	
14	Safety Input 1 (STO_DSP)	0	The drive is in safety state. The motor cannot be activated.	Safety state is reflected in bits 0 to 3. If motor was on during the safety state, MF latches the event and the motor will be turned off.
		1	The drive is not in safety state. The motor may be enabled.	
15	Safety Input 2 (STO_PWM)	0	The drive in safety state. The motor cannot be activated.	
		1	The drive is not in safety state. The motor may be enabled.	
16 to 17	Recorder Status	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 been fetched by the drive.	
18	Target Reached	According to TR[] and the relevant motion mode (OV[2]) 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.	



Bits	Description	Values		Notes
21	Shunt is active	0	Shunt is switched on	The shunt indication is relevant for specific drives only.
		1	Shunt is switched off	
22	Motor On (MO)	0	Motor is Off (MO=0)	
		1	Motor is On (MO=1)	
23	Movement defined by:	0	Standstill. No movement	Refer to objects 0x606F velocity threshold (OF[8]) and 0x6070 velocity threshold time (OF[9])
		1	Active movement. Not in standstill.	
24 to 26	Hall A, Hall B, Hall C state			
27	Safe Torque Off diagnostics error	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 OV[62] and 0x2086 object.		Note: This bit is cleared during power-up only.
28	The profiler stopped due to a switch.	Either a Hard Stop, FLS, RLS or Soft Stop 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.
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 not measuring the 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 bits 14 and 15.
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.
13 (0xD)	Over-temperature: The drive is sensing a temperature which exceeds the maximum allowed temperature limit.	9 4310	The actual temperature is reported by the TI[1] (TI[2] in Fahrenheit) command.
15 (0xF)	Additional Abort was activated. The drive sensed an input switch that is defined as Additional Abort (refer to IL[] command)	81 5442	The fault is similar to <i>Abort</i> function with different value report. This allows user to distinguish between two different faults states such as Inhibit and PTC.

Notes

- In the case of the Safety bits in the Gold Guitar and the Gold Trombone there might be situations in which STO 2 is not reported to the CPU. In these cases, when Safety 2 is activated (safety state), the PWM hardware will be inhibited, causing the motor to be in a freewheel



state, but the CPU will not be aware of it and will not report it to the user. Typically, a tracking error would be triggered.

- The indication that the hardware is suffering from that the user can detect by **PB**. If the value is 0x5AA5 (23205), STO 2 will not be reported.
- STO 1 works normally in all drives.

References

MF



ST[N] – Stop Profiler

ST[N] stops the profiler in stop deceleration.

CANopen/CoE

Elmo_parameters_objects: **0x31E7**

Attributes

Attribute	Description
Type	Command
Source	All
Restrictions	None
Range	N/A
Index range	1,4
Default	N/A
Unit modes	UM = 2, 3, 5
Non-volatile	N/A

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 at the value specified by the **SD[N]** command.

ST[N] has no effect over the auxiliary 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[N]**, **TC**, **UM**



SV – Save Parameters

SV saves non-volatile parameters from the RAM to the flash memory.

CANopen/CoE

Store_parameters: **0x1010** data bytes 0 to 3 's', 'a', 'v', 'e'

Elmo_parameters_objects: **0x31E9**

Attributes

Attribute	Description
Type	Command
Source	All, except the user program
Restrictions	MO = 0, and the user program is not running.
Range	None
Default	N/A
Unit modes	All
Non- Volatile	None

Remarks

The **SV** burns the parameters into the none volatile memory after it checks the parameters integrity. If one of the parameters fails and an error is produced, the saving procedure is not performed. The **CD** command details the parameter which failed and the reason for the failure.

The **SV** command may take a few hundreds of milliseconds to execute, during which the communication drivers are disabled.

References

LD, **CD**



SW – Status Word

SW informs the user regarding the status of the DS-402 (0x6041), via the Status Word.

CANopen/CoE

Statusword: **0x6041**Elmo_parameters_objects: **0x31EA**

Attributes

Attribute	Description
Type	Bit-Field, Read/Write
Source	All
Restrictions	None
Range	N/A
Default	0
Unit modes	All
Non- Volatile	No

Remarks

The Status Word is used in conjunction with the Control Word in the DS-402 CANopen standard for drives and motion. The Status Word is received with the CANopen or EtherCAT communication channel with object **0x6041**.

The **SW** 'get' command should normally be followed by the **CW** 'set' command, in order to be synchronized with the DS-402 state machine principles.

For more details about the **SW** bit field, refer to the object **0x6041** in the CANopen DS-402 manual.

References

CW



TC – Torque Command

TC specifies the torque command and switches to the current control loop.

CANopen/CoE

Elmo_parameters_objects: **0x31F0**

Attributes

Attribute	Description
Type	Float, Read/Write
Source	All
Restrictions	The motor must be on.
Range	Torque limits (-PL[1] to PL[1])
Default	0, Volatile. Cleared automatically when MO = 1
Unit modes	All
Non-volatile	N/A

Remarks

The **TC** command sets the torque (motor current) 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 reference value **IQ** (the **ID** command is zero unless phase advanced was used).

References

MO, **UM**, **IQ**, **ID**, **CL[]**, **PL[]**, **MC**, **LC**



TI[] – Temperature Information

TI[] gets temperature information from the drive hardware.

CANopen/CoE

Elmo_parameters_objects: **0x31F6**

Attributes

Attribute	Description
Type	Long, Read-only
Source	All
Restrictions	None
Range	TI[1] range: –40°C to 100°C (Celsius) TI[2] range: –40°F to 212°F (Fahrenheit) TI[3] range: –40°C to 120°C (Celsius)
Index range	1 to 3
Default	N/A
Unit modes	All
Non-volatile	No

Remarks

The drive hardware includes temperature sensors to measure the temperature of the driver's heat sink.

In cases in which the temperature exceeds 85°C, the servo will be shut off (**MO** = 0) with an over-temperature indication in the **MF** command and in the **SR** command.

The servo, in this case, can be enabled again when the temperature is ≤80°C.

In some cases when absolute encoders are used, the encoder manufacturer provides the sensor temperature and some more information about possible errors. Please refer to the **EE[]** command for more information.

In cases in which the sensor reports an error, the motion will be aborted with the proper **MF** command value.



Indices

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

Index	Description	Type	Values	Restrictions
1	Reads the heat sink temperature in Celsius	Long	Celsius	
2	Reads the heat sink temperature in Fahrenheit	Long	Fahrenheit	
3	Reads the BISS Absolute encoder temperature	Long	Celsius	

References



TM – Internal Time

TM reads and writes the 32-bit system time, in microseconds.

CANopen/CoE

Elmo_parameters_objects: **0x31FA**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	
Range	0 to 2^{32}
Default	0
Unit modes	All

Remarks

The **TM** command is based on the Real Time process internal counter, which counts time in microseconds.

This counter can be set and read using the **TM** command.

The **tdif()** function uses the **TM** internal counter to calculate time differences.

Users can measure time periods of activities using the **TM** command and the **tdif()** function.

For example (using parameter **UI[N]**):

```
UI[1]=TM
```

... any activity

```
UI[2]=tdif(UI[1])
```

UI[2] holds the time period of the measured activity, in milliseconds.

In the absence of CANopen SYNC signals and Stamp time objects, the microsecond counter runs freely, completing a cycle approximately once every 71.5 minutes.

In a CANopen system the high-resolution time stamp protocol may modify the internal time in order to synchronize between drives on the CAN bus.

The **tick** function also gets the system time of the Gold drive, but it is based on a different timer counter (hardware timer) and returns the time in microseconds.



TR[] – Target Radius

TR[] specifies the threshold which determines whether the load has reached its target.

CANopen/CoE

Elmo_parameters_objects: **0x31FF**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	See the table below.
Index range	1 to 6
Default	See the table below.
Unit modes	UM = 5, UM = 2
Non-volatile	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.

It is with conjunction of the CANopen target reached definition for position and velocity. The difference is that the CANopen objects are declared in User Units, while TR[] is given in counts and counts/sec.

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 (**OV[2]** or 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 250 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
0	Reserved			
1	Target Position window [counts]	-1 not active 0 to $2^{31} - 1$ [counts]	100 counts	Overrides object 0x6067
2	Target Position Window time	[msec]	20 msec	Overrides object 0x6068
3	Target Velocity window	[counts/sec]	100 counts/sec	Overrides object 0x606D
4	Target Velocity window time	[msec]	20 msec	Overrides object 0x606E
5	Torque Window	> 0 [1/1000 of rated torque]	40	Overrides object 0x2E06 , See OF[50] as well
6	Torque Window Time	> 0 [msec]	20 msec	Overrides object 0x2E07 , See OF[51] as well

References

SR



TS – Sampling Time

TS specifies the sampling time of the drive, in microseconds, which is used as the update time of the current controller.

CANopen/CoE

Elmo_parameters_objects: **0x3200**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	MO = 0, and the user program is not running.
Range	50 to 125
Default	50 (RS), Non-volatile
Unit modes	All
Non-volatile	Yes

Remarks

TS is the sampling time of the current loop. The sampling time of the velocity and position controller is either the same cycle or twice **TS** (**depend on TS**). By default, for a **TS** lower than 60 microseconds, the sampling time of velocity and position is twice **TS**. For **TS** higher or equal to 60 microseconds, the sampling time of velocity and position is every **TS**. For example, for **TS** = 50, the torque/commutation controller runs once every 50 microseconds, the speed and position controller executes every 100 microseconds, whereas for **TS** = 100, the torque/commutation controller and the speed and position controller executes every 100 microseconds.

The user can select to change the velocity/position loop sampling time for **TS** ≥ 60, to work every two times **TS**, by setting **XA[5]**=2 (see **XA[N]** command).

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 on the CPU, so that less computing power remains for executing interpreter and user program commands.

The drive does not allow an excessively low value for **TS** to prevent an overflow of the required CPU computing power.

For all unit modes, **WS[28]** gives the actual sampling time of the speed and position controller.

When **TS** is modified, the controllers loop gains must be re-tuned 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**.

TS must be an even number (when **XP[2]** = 4).

The PWM frequency (in Hz) is calculated using the formula $\text{XP}[2]/(2 \cdot \text{TS})$, and the current ripple frequency at the motor is twice as large.

References

[WI\[\]](#), [WS\[\]](#), [XP\[\]](#), [XA\[\]](#)



TW – Wizard Internal Identification

TW[M] is used to set the internal parameters during running of a Wizard and Simulation.

CANopen/CoE

Elmo_parameters_objects: **0x3204**

Attributes

Attribute	Description
Type	Long, Write-only
Source	All
Restrictions	
Range	
Default	Depends on the variable
Unit modes	
Non-volatile	All

Remarks

All values and functions set by **TW[]** are reset to default, during:

- Power-up
- The **RS** command
- The **SV** command

Indices

The following table describes the **TW** entries.

Index	Description	Units	Default	Restrictions
14	Sets command to Gantry Yaw	Counts	0	The motor must be on
18	Socket number (1 to 4) of Panasonic or Tamagawa to reset the sensor serial absolute position (single turn).			The motor must be off
19	Socket number (1 to 4) of Panasonic, Tamagawa or EnDat 2.2 to reset the sensor serial absolute multi turn position.			The motor must be off
20	Socket number (1 to 4) of EnDat 2.2 (ID=30) to reset the sensor errors and warnings			The motor must be off



Index	Description	Units	Default	Restrictions
80	Launch sine profile SE[N] , sensor ID = 8			
95	Enable software capture for analog encoders (alias CA[60] bit 9) 0 - disabled 1 - enabled		0	

References



UF[N] – Float User Interface

UF[N] specifies the float user variable array.

CANopen/CoE

Elmo_parameters_objects: **0x320D**

Attributes

Attribute	Description
Type	Parameter, Float, Read/Write
Source	All
Restriction	None
Range	9.999999E+36 to -9.999999E+36
Default	None
Index range	1 to 24
Unit modes	All
Non-volatile	Yes

Remarks

Users can use the 24 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[N] – Integer User Interface

UI[N] specifies the Long user variable array.

CANopen/CoE

Elmo_parameters_objects: **0x3210**

Attributes

Attribute	Description
Type	Parameter, Long, Read/Write
Source	All
Restriction	None
Range	-2147483647 to 2147483647
Default	None
Index range	1 to 24
Unit modes	All
Non-volatile	Yes

Remarks

Users can use the 24 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 – Unit Mode

UM specifies the higher allowed control loop.

CANopen/CoE

Elmo_parameters_objects: **0x3214**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	The motor must be off. Note that control loops can be freely switched without disabling the motor.
Range	1 to 6 (excluding 4)
Default	3
Unit modes	AN
Non-volatile	Yes

Remarks

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

UM Value	Description
1	Torque Control loop. The reference is set directly by TC . Values are processed immediately in the next control loop.
2	Speed 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.
3	Stepper. 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 512 ticks for 1 pole pair. TC must be set to excite the motor phases that allow the movement.
4	Reserved.



5	Position 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 or closed 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 512 ticks for one pole pair. In stepper closed loop, the position control is performed by changing the profile of the stepper angle. The Current during movement profile or standstill depends on the HT[] and FF[] commands. The units are 512 ticks for one pole pair. It is necessary to set the sensor ID=34 in this mode.

References

[CA\[N\]](#)



US[] – User Saturation Parameters

US[] specifies the User Saturation Parameters for the controller output.

CANopen/CoE

Elmo_parameters_objects: **0x321A**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Range	0 to 100
Index range	1 to 4
Default	100
Unit modes	According to array index
Non-volatile	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	Values	Restrictions
1	Limit PWM (voltage) command max range is WS[54] when US[1]=100	% of max PWM range	0 to 100	All unit modes
2	Limit the current integral. The limit is in additional 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. For compatibility reasons 0 is interpreted as 100%.	% 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. Limiting the integral reduces overshoots allowing faster settling time and keeping the use of integral for small errors. For compatibility reasons 0 is interpreted as 100%.	% of maximum current PL[1]	0 to 100	In position and velocity mode
4	Limit gantry's Yaw current	% of PL[1]	0 to 100	Only in gantry master

References

[PL\[\]](#), [UM](#), [WS\[\]](#)



VB – Software Boot Version

VB returns the drive's boot version.

CANopen/CoE

Remarks

Each drive includes boot software that loads the FW after power-up and enables updating of the firmware version. The boot software version can be retrieved using the **VB** command.

The version format is as follows: DSP Boot <version number><date>;

For example: "DSP Boot 1.1.1.2 21Apr2010;"

Attributes

Attribute	Description
Type	String, read-only.
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	None
Default	None
Unit modes	All
Non-volatile	No

References



VE – Velocity Error

VE gets the velocity tracking error.

CANopen/CoE

Elmo_parameters_objects: **0x3226**

Attributes

Attribute	Description
Type	Status report, Long
Source	Program, RS-232, CANopen
Restrictions	None
Range	N/A
Index range	N/A
Default	N/A
Unit modes	Position and velocity: UM = 2, 5
Non-volatile	N/A
Activation	

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 speed and position reference limits.

CANopen/CoE

VH[2] – **0x607F**

VL[3] – **0x607D.1**

VH[3] – **0x607D.2** **VH** - Elmo_parameters_objects: **0x3229**

VL - Elmo_parameters_objects: **0x322D**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	VH[3] > VL[3]
Range	VH[1] : 0 to $(2^{31} - 1)$ VH[2] : 1 to 2000000000 VH[3] : -2^{31} to $(2^{31} - 1)$ VL[3] : -2^{31} to $(2^{31} - 1)$
Index range	VH[N] : N=2, 3 VL[N] : N=3
Default	VH[2] = 2000000000 VH[3] = 900000000 VL[3] = -900000000
Unit modes	VH[2] : UM = 2, 3, 5 VL[3], VH[3] : UM = 5
Non-volatile	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]** (object **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 = $\min(\text{VH}[2], \text{object } 0x6080)$
- In the case of a linear motor: Velocity Limit = **VH[2]**

VH[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 **VH[1]**, sets the dynamic brake. However, if the speed is above value of **VH[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 is done to protect a 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 might endanger the drive.

Indices

The following table describes the **VH[]/VL[]** entries.

Index	Description	Units	Range
0	Reserved		
1	Maximum speed for dynamic brake	counts/sec	0 to $(2^{31} - 1)$
2	The reference to the speed controller is limited to the range $[-\text{VH}[2] \dots \text{VH}[2]]$.	User-defined	0 to 2000000000
3	The reference to the position controller is limited to the range $[\text{VL}[3] \dots \text{VH}[3]]$.	User-defined	-2^{31} to $(2^{31} - 1)$

Notes

- **VH[3]** and **VL[3]** should be set to 0 in 32-bit position modulo mode. In this mode all position limits are ignored.
- 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 **XA[4]** is set. Note that this setting of the **XA[4]** is the factory default.

References

XM[N], **HL[N]**, **LL[N]**



VO – Software OTP version

VO retrieves the drive's OTP (boot loader) version.

CANopen/CoE

Elmo_parameters_objects: **0x3230**

Attributes

Attribute	Description
Type	String, read-only.
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	None
Default	None
Unit modes	All
Non-volatile	No
Attribute	None

Remarks

Each drive includes an OTP version that is burned during the manufacturing process.

This command provides a way to retrieve the OTP version that resides in the drive.

The version format is as follows: DSP OTP <version number><date>;

For example: "DSP OTP 2.08 21Apr2010;"

References



VP – PAL Version

VP gets the PAL version.

CANopen/CoE

Attributes

Attribute	Description
Type	Long, read-only.
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	0 to 255
Default	None
Unit modes	All
Non-volatile	No
Attribute	None

Remarks

This command returns the PAL version that is directly read from the PAL.

In cases where the PAL version cannot be supported by the drive firmware version, the motor enable (MO=1) will not operate, producing the error code 182.

Note:

In cases in which the version cannot be read (e.g., Gold Guitar drives), the value will be 0.

Reference



VR – Software Firmware Version

VR retrieves the drive's firmware version.

CANopen/CoE

Attributes

Attribute	Description
Type	String, read-only
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	None
Default	None
Unit modes	All
Non-volatile	No
Attribute	None

Remarks

The **VR** command reports the firmware software version as a vstring.

The string includes:

1. The product name
2. The software version
3. The software release date
4. Hardware core type:
 - d. None – SCORE hardware core
 - e. G – GCON hardware core

The format is as follows: <Product name> <Software version> <Release date> <Core Type>

For example: "Whistle 01.01.04.34 04Oct2010G"

References



VS – STO Software Module Version

VS retrieves the Safe Torque Off software module version.

CANopen/CoE

Attributes

Attribute	Description
Type	String , read-only.
Source	RS232, USB, TCP, EoE
Restrictions	None
Range	None
Default	None
Unit modes	All
Non-volatile	No
Attribute	None

Remarks

This command returns the STO software module version included in the drive firmware version.

VS format example: "1.0.11.0"

Reference



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

0x606CElmo_parameters_objects: **0x3236**

Attributes

Attribute	Description
Type	Float, Read-Only
Source	All
Restrictions	None
Range	–2e9 to 2e9
Default	0
Unit modes	All
Non-volatile	No

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.

VX returns the same quantity as **VU**, but in counts per seconds.

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

0x6069Elmo_parameters_objects: **0x3239**

Attributes

Attribute	Description
Type	Float, Read-Only
Source	All
Restrictions	None
Range	–2e9 to 2e9
Default	0
Unit modes	All
Non-volatile	No

Remarks

In a velocity loop or a position loop with one sensor **VX** returns values in counts/sec.

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, **VX** returns the position sensor. If it is equal to 1, **VX** returns the velocity of the velocity sensor, but multiplied by a factor to be in the same units as the position sensor.

VU returns the same quantity as **VX**, but in user units.

References

VU, **FC[]**, **XM[]**



WI[] – Wizard Integer Parameters

WI[] gets internal information.

CANopen/CoE

Elmo_parameters_objects: **0x3244**

Attributes

Attribute	Description
Type	See the table below.
Source	All
Restrictions	None
Range	See the table below.
Index range	According to the table below
Default	N/A
Unit modes	All
Non-volatile	No

Remarks

Indices

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

Index	Description	Type	Values
20	Returns the maximum allowed recorder length	Long	16384
21	Returns the actual recorder length, depending on the number of selected signals	Long	1 to 16384
35	Returns the overvoltage maximum threshold	Long	0 to <BV
36	Returns the actual overvoltage threshold	Long	0 to ≤WI[35]
37	Returns the undervoltage minimum threshold	Long	≥0
38	Returns the actual undervoltage threshold	Long	≥WI[37]
40	Returns the High Temperature threshold of the drive	Long	N/A
51-54	Returns the hardware position readout of the socket	Long	
70	Returns the Channel A of the external 16 bits A2D, if supported if supported by the hardware	Long	-32768 to 32767



71	Returns the Channel B of the external 16 bits A2D, if supported by the hardware	Long	-32768 to 32767

References



WS – Miscellaneous Reports

The **WS** command provides information about the state and internal variables of the drive. Mainly used for internal use only.

CANopen/CoE

Elmo_parameters_objects: **0x324E**

Attributes

Attribute	Description
Type	Floats, Long, Read only
Source	All
Restrictions	
Range	
Index range	Refer to the table below
Default	
Unit modes	All
Non-volatile	No

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	Type	Units\Values	Notes
3	CPU main clock frequency. Default value: 150000000. Refer to the product manual.	Long	Hz	
4	PWM frame Default=150xTS/XP[2]	Long	150 MHz clock	
5	Bits per ampere (MaxADCvalue/MC)	Float		
7	Quad Port B flag	Long		Bit0: first pulse arrived after reset



Index	Description	Type	Units\Values	Notes
8	hardware board type	Long	0: SCORE board 1:GCON revision A 2: GCON revision C 3: GCON revision E	SCORE: GGUI GCON: GWHI, GDRU, GTRO
9	ADC instant delay. Units in 150 MHz clock.	Long	150 MHz clock	For internal use only.
11	PWM emulation duty cycle	Float	%	
12	Dead Band delay. Units in 150 MHz clock.	Long	150 MHz clock	For internal use only.
16	Planar: bit 0 – commutation known bit 1 – Amplitude at motor off over CA[52]	Long		For planar only Also in object 0x2085
17	Correlation in current experiment $\Sigma(0.25*fTorqueCmdOld^2)$	Long		For internal use only.
18	VDC average $\Sigma(stStateVec.SAMPLE_VBUS)$	Long		For internal use only.
19	ERR_AVG stWizCor.fErrorAvg	Float		For internal use only.
22	1/Torque scale	Float		WS[22] convert from Ampere to ADC (internal units)
23	Factor to convert user A/D bits to volts: basically, $10.0/(2048.0 - 205.0)$	Float		
24	Returns the internal offset of Analog Input 1.	Float	Volts	
28	Returns the velocity loop cycle time.	Long	microseconds	For compatibility
29	Returns the position loop recording cycle time.	Long	Default: 2 * TS	For compatibility



Index	Description	Type	Units\Values	Notes
30	Product information Bits 0 to 7 Product Name Bits 8 to 11 Reserved for product name Bit 12 0: Atmel is connected 1: No Atmel Bit 13 Always 0 Bit 14 Project (always 1 for Gold Line, 0 for SimplIQ) Bit 15 Always 0 Bit 16 CAN integrated Bit 17 0: EtherCAT 1: TCP/IP Bits 18 to 20 Feedback type: 0: E type (Encoder + Encoder, Analog sensor) 1: A type (Absolute + Encoder, Analog sensor) 2: R type (Encoder, Analog sensor + Resolver) 3: M type (Absolute + Resolver) Bit 21 Define R type drive: PL[1] toque can be used as continuance Bit 22 EtherCAT ID switches Bits 23 ECAT LEDs exists Bit 24 Ethernet hardware:	Long		



Index	Description	Type	Units\Values	Notes
	0: Absent 1: Exist Bit 25 Drive includes digital current sensor Bits 26 to 31 Reserved			
31	Return the Gantry Yaw cycle time	Long	microseconds	
33	The instantaneous saturated high torque value	Float	Amperes	Nominal PL[1]
34	The instantaneous saturated low torque value.	Float	Amperes	Nominal CL[1]
35	Gantry differential position or additional position error (stepper closed loop)	Long	Counts	
36	ADC value at nominal Bus voltage	Long	ADC	
39	PWM Duty Cycle – On Time	Long	150 MHz clock	Sensor ID=35
40	PWM Duty Cycle – Cycle Time	Long	150 MHz clock	Sensor ID=35
41	PWM Duty Cycle – Duty Cycle	Float	Percent	Sensor ID=35
50	Sum of Reference Signal used in tuner $\Sigma(fTorqueCmdOld)$	Float	Internal torque units	For internal use only
51	Wizard state counter used in tuner	Long		For internal use only
52	Minimum Bus Voltage for schedule current controller	Long	Internal A2D units	For internal use only
53	Convert values from internal units to bus voltage	Float		
54	Maximum PWM (voltage) command allowed in drive	Long	150 MHz clock	
55	Position Loop cycle time	Long	microseconds	
56	Minimum PWM (voltage) command allowed in drive	Long	150 MHz clock	



Index	Description	Type	Units\Values	Notes
57	PWM (voltage) command range (WS[54]-WS[56])	Long	150 MHz clock	
61	Update cycle time of velocity/position loop. 0 – every 2*TS 1 – every TS	Long		
70	PWM reference from user			Similar to Object 0x2E20
72	Angle of Sin/Cos encoder	Long		For internal use only
75	Mode of sine virtual sensor: 0 – stopped 1 – ramp 2 – start the two sine signals 3 – wait for high frequency to stop 4 – wait for low frequency to stop 5 – ramp down	Long		For internal use only
91	Signal #1 sine coefficient	Float		Provide Valid info after TW[81]=N . N is a bit field that determines the vectors to identify. For internal use only
92	Signal #1 cosine coefficient	Float		
93	Signal #2 sine coefficient	Float		
94	Signal #2 cosine coefficient	Float		
95	Signal #3 sine coefficient	Float		
96	Signal #3 cosine coefficient	Float		
97	Signal #4 sine coefficient	Float		
98	Signal #4 cosine coefficient	Float		
99	Signal #1 quality function	Float		
100	Signal #2 quality function	Float		
101	Signal #3 quality function	Float		
102	Signal #4 quality function	Float		
103	Offset of signal #1	Float		
104	Offset of signal #2	Float		
105	Offset of signal #3	Float		
106	Offset of signal #4	Float		



Index	Description	Type	Units\Values	Notes
111	Temperature ADC Out	Long	ADC digital out	For internal use only

References

[TW\[N\]](#), [WI\[N\]](#)



XA[] – Extra Parameters

XA[] specifies some extra and special parameters.

CANopen/CoE

Elmo_parameters_objects: **0x3256**

Attributes

Attribute	Parameter, Integer
Type	Long, Read/Write
Source	All
Restrictions	MO = 0
Range	See below.
Index range	2 to 5
Default	See below.
Unit modes	All
Non-volatile	Yes

Remarks

Indices

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

Index	Description	Default	Values	Restrictions
0	Reserved			
1	Reserved			
2	Maximum hot plug in speed. Measured in electric cycles per seconds	5	>0	For compatibility, zero is same as 5 Hz
3	Current exceeded detection time in microseconds. Executes fault when current is above 15 of maximum current MC for same period.	600	0-9000	For compatibility, zero is same as 600 microseconds



4	Cancel Stop Manager features:		0	0, 2, 4, 6	In Interpolated Position Mode and in Cyclic Synchronous Position Mode (DS402 object 0x6061 is 7 or 8) the Acceleration limiting, denoted by SD, is bypassed regardless of bit 1 setting.
	Bit	Description			
	0	Reserved			
	1	Bypass acceleration limiting			
	2	Ignore FLS, RLS, hardware Stop & SW position limits (VH[3] \ VL[3]) in Interpolated Position and all Cyclic Synchronous modes (DS-402 Object 0x6061 is 7, 8, 9, 10)			
5	Set the cycle time of velocity/position control loop		0	0, 1,2	It is recommended to leave this parameter at default value.
	0	Using TS . TS <60: Velocity/Position loop is updated every 2* TS TS >=60: Velocity/Position loop is updated every TS			
	1	Reserved			
	2	Velocity/Position control loop is updated every 2* TS .			

References

TS, **CA[60]**, **AC**, **DC**, **SD**



XC – Resume Program

XC resumes a halted user program.

CANopen/CoE

Attributes

Attribute	Description
Type	Command
Source	All, except the user program
Restrictions	PS ≥ 0
Range	None
Default	None
Unit modes	All
Non- Volatile	None

Remarks

While the user program is running, the user can halt the program temporarily, using the **HP** command.

The **XC** command is used to continue running the user program from the point at which it was halted.

The **XC** command cannot release the program from a breakpoint. For that purpose, use the **DB##GO[0]** command.

References

HP, **XQ**, **KL**



XI[] – Extended Digital Input Bits

XI[] reads the extended digital input bits. The command can also sets the logic for the extended inputs.

Extended Inputs are hardware depended. A drive can have any number of extended inputs between 0 to 32. Consult the installation guide of the specific drive.

When the hardware is not supported, the command returns 0.

CANopen/CoE

Elmo_extended_inputs: **0x2202**Elmo_parameters_objects: **0x325E**

Attributes

Attribute	Description
Type	Long Index 1: Read\Write (write has no effect and does not return error) Index 2: Read\Write Index 3: Read\Write (Reserved, setting has no affect)
Source	All
Restrictions	None
Range	[0...0xFFFFFFFF]
Index range	1 to 3
Default	None
Unit modes	All
Non-volatile	Yes

Remarks

The extended digital inputs are updated every 250μSec.

The extended inputs are general purpose inputs and are not subjected to any internal function.

Index 3 is reserved

The extended inputs can be mapped to a PDO via CANopen and EtherCAT.

Indices

Index	Description	Values	Restrictions
1	Reads the extended digital input	0..0xFFFFFFFF	



2	Read\Write the logic of the input. Result is XOR operation: Value = XI[2] ^{Inputs}	0xFFFFFFFF	
3	Reserved	0x0	

References

xO[]



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.2**Elmo_parameters_objects: **0x3262**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	- The motor must be off. - Homing must not be active (HM[1] = 0).
Range	-2^{31} to $(2^{31} - 1)$
Index range	1, 2
Default	XM[1] = -1000000000 XM[2] = 1000000000
Unit modes	All
Non-volatile	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]** .
- Modulo motion.** The position counts in a position range limit (object **0x607B** 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]**.
- 32-Bit modulo motion.** If **XM[2] = XM[1] = VH[3] = VL[3] = 0**, neither the software position limit nor the position range limit affects the motion.

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.

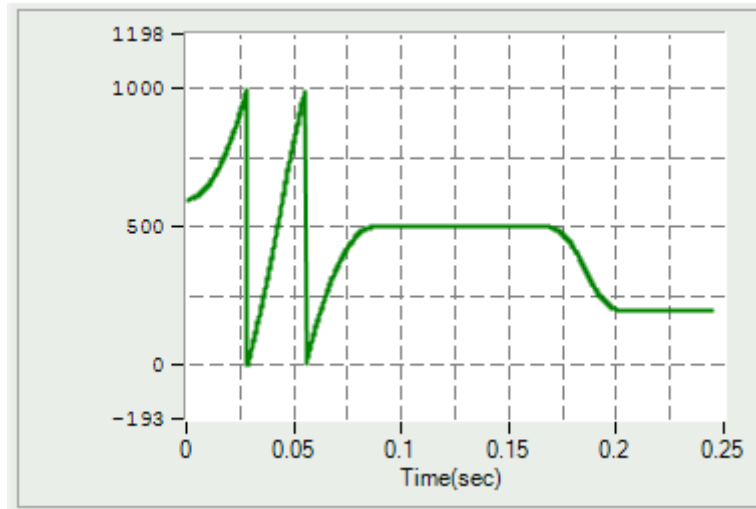
Using object **0x60F2** bits 7:6 (or **PO** bits 7:6):

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 DS402 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** (or **OV[32]**) is set using the control word (object **0x6040**), the setting of object **0x60F2** (**OV[28]**) 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]**).

32-Bit modulo motion

If **XM[2]** = **XM[1]** = **VH[3]** = **VL[3]** = 0, neither the Software Position limit nor the Range limit affects the motion. Modulo mode should be set to normal positioning.

This mode is designed in order to support an interface to a 64-bit master. On the master side, there could be two possible approaches for supporting this mode:

- The position command passes a 32-bit modulo ($0 \dots 2^{32} - 1$);
- The position command is non-modulo, but the drive receives only the lower 32 bits of the position target.

The master command is subject to the following restriction: the difference between two consecutive 64-bit position targets must not exceed $2^{31} - 1$.

Example

In order to move 6000000000 counts, three consecutive movements should be requested:

```
0x607A (or OV[32]) = 0x77359400;    // set abs. target 2000000000
```



```
0x607A (or OV[32]) = 0xee6b2800; // set abs. target 4000000000
0x607A (or OV[32]) = 0x65a0bc00; // set abs. target
// 6000000000=0x165a0bc00
```

Indices

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

Index	Description	Type	Values
1	Minimum position range limit	User-defined	-2^{31} to $(2^{31} - 1)$
2	Maximum position range limit	User-defined	-2^{31} to $(2^{31} - 1)$

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** is be set to the range by taking the modulo:

$$PU = (PU - XM[1]) \bmod (XM[2] - XM[1]) + XM[1], \text{ if } PU > XM[2], \text{ or}$$

$$PU = -(-PU + XM[2]) \bmod (XM[2] - XM[1]) + XM[2], \text{ if } PU < XM[1].$$

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 250 μ sec. The drive will not indicate this as a failure, and it is up to the user to take care of this 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 Digital Output Bits

XO[] sets the extended digital output bits. The command can also read the logic for the extended inputs.

Extended Outputs are hardware depended. A drive can have any number of extended outputs between 0 to 32. Consult the installation guide of the specific drive.

CANopen/CoE

Elmo_extended_inputs: **0x22A1**Elmo_parameters_objects: **0x3264**

Attributes

Attribute	Description
Type	Long Index 1: Read\Write (Read returns the last setting including XO[2]) Index 2: Read\Write Index 3: Read\Write
Source	All
Restrictions	None
Range	[0...0xFFFFFFFF]
Index range	1 to 3
Default	None
Unit modes	All
Non-volatile	Yes

Remarks

The extended digital output are updated every 250µSec. The output can be received via **XO[1]** or tPDO (CANopen \ EtherCAT).

The extended outputs are for general purpose only and are not subjected to any internal function.

Index 3 is reserved

Indices

Index	Description	Values	Restrictions
1	Reads the extended digital input	0	
2	Reserved (setting has no effect)	0xFFFFFFFF	
3	Output mask. Relevant only when	0xFFFFFFFF	



	output are set via XO[1] . Not applicable when output are set via rPDO (object 0x22A1)		

References

[XI\[\]](#)



XP[] – Extra General Parameters

XP[] specifies extra general parameters for adapting the drive to special situations.

CANopen/CoE

Elmo_parameters_objects: **0x3265**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	The motor must be off.
Range	See the table below.
Index range	1 to 13
Default	See the table below.
Unit modes	All
Non-volatile	Yes

Remarks

Indices

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

Index	Description	Default	Values	Notes
0	Reserved			
1	Defines the overvoltage threshold in volts. Can only be reduced from the default value (BV)	WI[35]	0 to WI[35]	
2	Defines the PWM frequency as a factor of the controller sampling time: $TS/TPWM = TS \cdot fpwm = XP[2]/2$ The current ripple frequency at load is equal to XP[2]/TS .	Drive module depended	1 to 6	Some values are blocked by the drive. Thermal limitations may also apply.



Index	Description	Default	Values	Notes
3	Used in CANopen Interpolated Position mode to determine the filter frequency [Hz] between the SYNC signal to the actual interpolated time. The purpose of the filter is to prevent drifts between the master and slave as well as filter inherent jitters of the CAN bus SYNC message.	1	1 to 4000	
4	Filter constant of bus voltage measurements in Hz.	1500	100 to 3,000	
5	Maximum current command rate in one cycle in percent of MC .	20	10 to 200	
6	Low-pass constant for pre-filter of current loop in Hz.	3000	0 to 20000	If it equals zero, the filter is bypassed.
7	Reserved			
8	Phase offset of PWM slave synchronization in 150MHz clock	0		
9	Sync PWM to other drives	0	0 to 2	
	0 No sync			
	1 Sync as master			
	2 Sync as slave			
10 to 12	Reserved			
13	Defines the under voltage	WI[38]		

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

PL[], **CL[]**, **BV**, **TS**



XQ – Execute User Program

XQ executes the user program from a specified label or runs a specified function.

CANopen/CoE

Attributes

Attribute	Description
Type	Command , String
Source	All, except the user program
Restrictions	PX = -1 CC was executed correctly. Wizard mode is not active.
Range	XQ##func_name For example, if function to be run is main , the syntax is: XQ##main (or xq##main)
Default	None
Unit modes	All
Non-volatile	None

Remarks

XQ## executes a valid user program.

This command is 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 and c as arguments to the function.
- **XQ##LABEL** runs from **##LABEL**.

The **XQ** command clears the error status of the program along with the run-time error flags.

It does not reset program variables and does not clear the interrupt mask.



Notes

- **XQ** must include **##**. If it is omitted, an error is returned.
- **XQ##**, beside acknowledge, does not return a value.

References

CC, CP, HP, KL, PS, XC



XT[] – Threshold Parameters setting

XT[] sets threshold for various parameters. If this threshold is violated a warning is indicated via Extra Status Register (**OV[61]** & object **0x2085**).

CANopen/CoE

Elmo_parameters_objects: **0x3269**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	None
Range	See the table below.
Index range	1 to 4
Default	See the table below.
Unit modes	All
Non-volatile	Yes

Remarks

The threshold is checked every 250uSec and are not latched. The indication when the threshold is violated is via the map-able object **0x2085**.

The analog threshold signals (**XT[3]**, **XT[4]**) are relevant to Sine\Cosine Analog Feedback, Gurley absolute sensor, Hipreface & Resolvers. The amplitude is calculated as follows:

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

Amplitude can be recorded via EASII: “SinCos amplitude”

Nominal value for the 1V p-p is ~2,380,000.

The voltage is calculated from the amplitude as:

$$\text{Amplitude(v)} = \text{Sqrt}(\text{SinCos amplitude}) / 3085.$$

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

Index	Description	Default	Values
0	Reserved		
1	Defines the low bus voltage in milli volts that if power drops below this value bit 2 of the Extra Status reg will be set. The bus voltage is retrieved via AN[6] .	0	[0... BV *1000]



Index	Description	Default	Values
2	Defines the highest bus voltage in milli volts that If the high power is increases over this voltage than bit 3 of the Extra Status register will be set. The bus voltage is retrieved via AN[6] .	BV * 1000	[0... BV *1000]
3	Defines the highest amplitude threshold of the sine\cosine analog feedbacks in internal A2D ticks. See details above. If the amplitude is violated bit 4 of the Extra Status Register will be set.	0	[0...10E6]
4	Defines the highest drive temperature threshold in °C. Drive temperature is retrieved via TI[1]b or TI[2] commands. If the drive temperature is higher than the threshold bit 5 of the Extra Status Register will be set.	85	[0...125]

Notes

References

OV[], **TI[]**, **BV**, **AN[6]**



YM[] – External Reference Modulo

YM[] specifies the counting range for the external reference, which is **[YM[1]...YM[2] - 1]**.

The external reference can be retrieved by object **0x20A0** or the **PY** command.

CANopen/CoE

YM[1] – **0x207B.1**

YM[2] – **0x207B.2**Elmo_parameters_objects: **0x327C**

Attributes

Attribute	Description
Type	Long, Read/Write
Source	All
Restrictions	The motor must be off
Range	-2^{31} to $(2^{31} - 1)$
Index range	1, 2
Default	YM[1] = -1000000000 YM[2] = 1000000000
Unit modes	All
Non-volatile	Yes

Remarks

- The position of the external reference 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 **YM[1]**=-5 and **YM[2]**=5, the external reference is counted in a cycle length of 10. The external reference will always be in the range [-5...4]. If the external reference rotates in the positive direction, the external reference count will proceed from 0, 1, 2, 3, 4 to -5, -4, -3, -2, -1, 0, 1..... and so on.
- A new **YM[1]/YM[2]** setting is activated after the setting of **YM[2]**, at power-up or upon issuing **MO=1** command.
- YM[2]** must be bigger than **YM[1]** (**YM[2]**>**YM[1]**).
- If **YM[2]** = **YM[1]** = 0, the external reference modulo functionality is disabled. Actually, this means that **PY** counts in the Long counting range (-2^{31} to $(2^{31} - 1)$).
- If **YM[1]** or **YM[2]** is set so that **YM[2]** > **YM[1]**, but **PY** is out of the range **[YM[1]...YM[2]]**, **PY** will be set to range by taking the modulo: **PY** = (**PY** - **YM[1]**) mod (**YM[2]** - **YM[1]**) + **YM[1]**
- If the **PY** value setting is out of the range **[YM[1]...YM[2]]** which is requested, the request will be ignored and **PY** will not change



- If the **YM[N]** value is selected low and the external reference speed is too high, more than one full revolution of the counter may elapse within a single sampling time. This will cause the **PY** counter to behave unpredictably.

References

PY



Inspiring Motion

Since 1988

For a list of Elmo's branches, and your local area office, refer to the Elmo site www.elmomc.com

