

Parameter-Referenz

Grundlagen	2
Alle Parameter in alphabetischer Übersicht.....	4
Achsparameter im Detail.....	9
Globale Parameter im Detail.....	125

Grundlagen Es gibt drei Parametertypen:
Init-Parameter, Benutzerparameter und Programmparameter:

Init-Parameter ... sind die Defaultwerte, die ab Werk in der Positioniersteuerung fest gespeichert sind. Sie sind nach der Auslieferung aktiv und können vom Benutzer verändert werden.

Die Init-Parameter können durch einen Parameter-Reset im Menü **Steuerung** → **Reset** jederzeit zurückgesetzt werden. Das Löschen des gesamten Speichers im Menü **Steuerung** → **Speicher** → **EPROM löschen** setzt ebenfalls die Parameter auf die Werkseinstellungen zurück.

Benutzerparameter ... sind die durch den Benutzer geänderten (Init-)Parameter. Sie werden im EPROM gespeichert und bleiben auch nach dem Abschalten der Steuerung erhalten.

Die Benutzerparameter sind bei jedem Einschalten der Steuerung aktiv bzw. werden vor dem Ausführen jedes Programms aktiviert.

Sie gelten für alle Programme.

Programmparameter ... werden die Parameter genannt, die innerhalb eines Programms (mit den Befehlen SET verändert wurden. Damit kann man kurzfristig Parameter ändern, ohne dass dies bleibende Konsequenzen hat.

Im Gegensatz zu den Init- bzw. Benutzerparametern sind sie nur während einer Programmausführung und im Direktmodus gültig; danach werden sie gelöscht und durch die Benutzerparameter ersetzt.

Wenn Sie zum Beispiel für eine spezielle Anwendung den maximalen Schleppabstand sehr groß wählen wollen, setzen Sie diesen Filterparameter mit dem SET Befehl. Nach der Programmausführung sind wieder die Benutzerparameter gültig.

Parametergruppen Die Parameter für das APOSS-Programm sind der besseren Übersicht wegen in Gruppen eingeteilt; sie gelten generell für die ganze Steuerung:

Globale Parameter GL Alle achsen-unabhängigen Parameter wie die **Aktivierte Programmnummer** PRGPAR und die I/O-Parameter wie **Fehler löschen** I_ERRCLR sind in der Gruppe GL zusammengefasst und werden mit GET und SET bearbeitet.

Achsparemeter AX... Dieser Gruppe sind alle achsenspezifischen Parameter zugeordnet, die mit den Befehlen GET und SET bearbeitet werden und im Befehl eine Angabe der Achsen benötigen:

Grundeinstellungen (AXB, war bisher AXV)

Gebersystem (AXE)

Vordefinierte Ein-/Ausgänge (AXI) – I/O Konfiguration

Referenzierung (AXH) – Homefahrt

Synchronisation und Sync Marker (AXS)

Pos. Regelung (AXR)

Verstärker (AXA) –Verstärkerdaten, PI Geschwindigkeit und Drehzahlsteuerung

Die Spalte Parametergruppe (AXE, AXR etc.) verweist auf die Dialogfenster in denen Sie die Parameter mit **Steuerung** → **Parameter** → **Bearbeiten** können.

Parameter benutzen	Es gibt drei Methoden die Parameter zu bestimmen: <ul style="list-style-type: none">– PC Software APOSS– Externe SPS– CAN bus
Parameter ändern und speichern	<p>Parameter, die mittels Steuerung → Parameter → Bearbeiten geändert wurden, werden in das EPROM gespeichert wurden, bleiben auch nach dem Stromabschalten erhalten.</p> <p>Parameter, die durch das APOSS Anwendungsprogramm mit dem Befehl SET geändert wurden, werden nur im RAM gespeichert und sind daher nach dem Stromabschalten verloren.</p> <p>Parameter, die durch das APOSS Anwendungsprogramm mit dem Befehl SET geändert wurden, sind nur aktiv während das Anwendungsprogramm läuft. Diese Parameter können mit dem Befehl SAVEEPROM in das EPROM gespeichert werden und bleiben dann auch nach dem Stromabschalten erhalten.</p> <p>!!! Bitte beachten Sie, dass ein EPROM eine begrenzte Lebenszeit hat; es kann aber ungefähr 10000-mal programmiert werden.</p>
Allgemeines zu den Parameterwerten	Einige Grenzwerte sind auf Grund der besseren Lesbarkeit mit 1 Mrd. angegeben. Der exakte Wert beträgt jedoch 1.073.741.823 (= MLONG).
Eingabebereich	<p>Die Überschreitung der angegebenen Eingabebereiche wird vom Programm nicht geprüft, da es wegen der großen Wertebereiche keine sinnvollen Kontrollmöglichkeiten gibt.</p> <p>!!! Schon innerhalb der angegebenen Bereiche kann es durch die großen Leistungsunterschiede der Motoren und den vielseitigen Anwendungsmöglichkeiten zu unsinnigen Eingaben kommen. Es liegt daher in der Verantwortung der Programmierer und Anwender, auf die zulässigen Leistungsbereiche der Antriebe und des Systems zu achten.</p> <p>!!! Wenn der Parameterwert außerhalb des definierten Wertebereichs ist, wird der Befehl nicht korrekt dargestellt und ausgeführt.</p>

Parameter-Referenz ♦ Alle Parameter in alphabetischer Übersicht

Alle Parameter in alphabetischer Übersicht

Die Reihenfolge der Parameter wird durch die interne Parameternummer bestimmt. Orientieren Sie sich am besten zuerst in der Übersicht; dann finden Sie die Detail-Informationen ganz schnell anhand der internen Parameternummer oder mit Hilfe des Links.

Parameterkennung	Parameterbezeichnung	interne Par. Nr.	Par.-Gruppe	Einheit	Init-Wert
ACCMAXQC	Maximalbeschleunigung	10	AXB	qc/st ² * 1/65536	
AINFTIME	Definiert die Filterzeit für den analogen Eingang.	109	GL	–	0
AMPCOMMTYPE	Konfiguration des Motortyps, der an MACS-Steuerungen mit integrierten Leistungsstufen angeschlossen ist.	95	AXA	–	0
AMPENCNO	Definition des Encoder-Eingangs, der als Rückführung für die Geschwindigkeitsregelung verwendet wird.	126	AXA	–	-1
AMPENCRES	Auflösung des Encoders, der für die Geschwindigkeitsregelung verwendet wird.	127	AXA	qc/Umdr.	-1
AMPENCRPM	Maximale Drehzahl des Encoders der für die Drehzahlregelung verwendet wird.	128	AXA	Umdr./Min.	0
AMPMAXCUR	Begrenzung des maximal zulässigen Motorstroms bei MACS4-Steuerungen mit integrierten Leistungsstufen.	94	AXA	mA	2000
AMPMODE	Auswahl der Reglerstruktur/-kaskadierung für MACS-Steuerungen mit integrierten Endstufen.	97	AXA	–	0
AMPPWMFREQ	PWM-Taktfrequenz von MACS4 Steuerungen mit integrierten Leistungsstufen.	96	AXA	Hz	24000
BANDWIDTH	PID-Bandbreite	35	AXR	%	1000
CANBAUD	CAN-Baudrate	101	GL	–	5
CANNR	CAN-Teilnehmernummer	100	GL	–	0
CANSYNCTIMER	Bestimmt die Zykluszeit für den Versand von SYNC-Telegrammen auf dem CAN-Bus.	114	GL	ms	0
CURKILIM	Grenzwert für die Integralsumme der PI Strom-/Drehmomentregelung.	104	AXA	–	1000
CURKINT	Integralfaktor der PI Strom-/Drehmomentregelung.	93	AXA	–	100
CURKPROP	Proportionalfaktor der PI Strom-/Drehmomentregelung	92	AXA	–	200
DFLTACC	Default-Beschleunigung	34	AXB	%	50
DFLTVEL	Default-Geschwindigkeit	33	AXB	%	50
DRIVETYPE	Definiert die zu verwendende Art und den Typ des Sollwertausgangs für die Achsregelung.	0	AXR	–	0
ENCCONTROL	Konfiguration, wie die Position nach einem Wechsel der Drehgeberquelle aufgeholt werden soll.	107	AXE		0

Parameter-Referenz ♦ Alle Parameter in alphabetischer Übersicht

Parameterkennung	Parameterbezeichnung	interne Par. Nr.	Par.-Gruppe	Einheit	Init-Wert
ENCODER	Auflösung des Istwertgebers (Geberstrichzahl)	2	AXE	Strichzahl/ Umdr.	500
ENCODERCLOCK	Interne oder externe Takterzeugung für SSI Encoder	73	AXE	–	1
ENCODERFREQ	Definiert die Frequenz für SSI Encoder.	74	AXE	kHz	262000
ENCODERTYPE	Definiert die Signalquelle des Istwertgebers für die Achsregelung.	27	AXE	–	0
ENDSWMOD	Verhalten bei Endschalter	44	AXI	–	0
ERRCOND	Verhalten im Fehlerfall	43	AXI	–	0
ESCCOND	Verhalten bei Programmabbruch	70	AXI	–	0
FEEDDIST	Faktor Benutzereinheit um CANopen-kompatible Benutzereinheiten zu erhalten.	112	AXE		1
FEEDREV	Faktor Benutzereinheit für die CANopen-Kompatibilität.	113	AXE		1
FFACC	Beschleunigungs-Feedforward	37	AXR	%	0
FFVEL	Geschwindigkeits-Feedforward	36	AXR	%	0
HOME_FORCE	Homefahrt erzwingen?	3	AXH	–	0
HOME_OFFSET	Nullpunkt-Offset bezüglich Maschinen-nullpunkt (MN) bzw. Home-Position	42	AXH	qc	0
HOME_RAMP	Rampe für Homefahrt	41	AXH	%	10
HOME_VEL	Homefahrt-Geschwindigkeit	7	AXH	%	10
HOME_TYPE	Homefahrt Verhalten	40	AXH	–	0
HOMEZEROVEL	Geschwindigkeitsteiler, um auf den Drehgeberindex zu fahren, als Teil des Befehls HOME oder bei CANopen Homefahrt-Prozeduren.	116	AXH	typ. %	10
I_BREAK	Eingang für Abbruch	105	GL	–	0
I_CONTINUE	Programm fortsetzen	106	GL	–	0
I_ERRCLR	Fehler löschen	107	GL	–	0
I_NEGLIMITSW	Endschalter negativ	47	AXI	–	0
I_POSLIMITSW	Endschalter positiv	46	AXI	–	0
I_PRGCHOICE	Eingang für Programmwahl Anfang	104	GL	–	0
I_PRGSTART	Eingang für Programmstart	103	GL	–	0
I_REFSWITCH	Eingang für Referenzschalter	45	AXI	–	0
I2TFTIME	Filterzeit in ms um den gefilterten I ² Wert zu bilden.	117	AXA	ms	0
I2TLIMIT	Begrenzung des I2TVALUE	118	AXA	A ² (1/1000)	0
JERKMIN	Zeitspanne zum Erreichen der definierten Maximalbeschleunigung	98, 100, 101, 102	AXB	ms	0
KDER	Differentialfaktor für PID-Regelung	12	AXR	–	1

Parameter-Referenz ♦ Alle Parameter in alphabetischer Übersicht

Parameterkennung	Parameterbezeichnung	interne Par. Nr.	Par.-Gruppe	Einheit	Init-Wert
KILIM	Grenzwert für die Integralsumme der PID-Regelung	21	AXR	–	0
KILIMTIME	Zeitspanne, über welche die Integrationsgrenze des Positionsreglers auf den definierten KILIM-Wert erhöht bzw. auf null reduziert wird.	105	AXR	ms	0
KINT	Integralfaktor für PID-Regelung	13	AXR	–	0
KPROP	Proportionalfaktor für PID-Regelung	11	AXR	–	30
MENCCONTROL	Konfiguration, wie die Master-Position nach einem Wechsel der Drehgeberquelle aufgeholt werden soll.	108	AXE		0
MENCODER	Auflösung des Drehgeber 1 (Master)	30	AXE	Strichzahl/ Umdr.	500
MENCODERCLOCK	Interne oder externe Takterzeugung für SSI Master-Encoder.	77	AXE	–	1
MENCODERTYPE	Definiert die Signalquelle des Master-Gebers.	67	AXE	–	0
NEGLIMIT	Negative Software-Wegbegrenzung	4	AXI	qc	–500000
O_AXMOVE	Ausgang für Fahrbefehl aktiv	64	AXI	–	0
O_BRAKE	Ausgang für Bremse	48	AXI	–	0
O_ERROR	Ausgang für Fehler	108	GL	–	0
POSDRCT	Drehrichtung	28	AXE	–	1
POSENCOC	Faktor Benutzereinheit für die CANopen-Kompatibilität	114			1
POSENCREV	Faktor Benutzereinheit für die CANopen-Kompatibilität	115			1
POSERR	Maximal tolerierter Positionsfehler	15	AXR	qc	20000
POSERRTIME	Zeitfenster [ms] für das Auslösen eines Positionsfehlerstatus.	111	AXR	ms	0
POSFAC_T_N	Nenner Benutzerfaktor	26	AXE	–	1000
POSFAC_T_Z	Zähler Benutzerfaktor	23	AXE	–	1000
POSLIMIT	Positive Software-Wegbegrenzung	5	AXI	qc	500000
PRGPAR	Aktivierte Programmnummer	102	GL	–	–1
PROFTIME	Abtastzeit für Profilgenerator	29	AXR	ms	1000
RAMPMIN	Kürzeste Rampe	31	AXB	ms	1000
RAMPTYPE	Rampenform	32	AXB		0
REGWMAX	Größe des Regelfensters (Aktivierung)	38	AXR	qc	0
REGWMIN	Größe des Regelfensters (Deaktivierung)	39	AXR	qc	0
REVERS	Reversierungsverhalten der geregelten Achse.	63	AXR	–	0
SWNEGLIMACT	Negative Software-Wegbegrenzung aktiv	19	AXI	–	0
SWPOSLIMACT	Positive Software-Wegbegrenzung aktiv	20	AXI	–	0
SYNCCACCURACY	Größe des Genauigkeitsfensters für Positionssynchronisation	55	AXS	qc	1000
SYNCFAC_T_M	Synchronisationsfaktor Master (M:S)	49	AXS	qc	1

Parameter-Referenz ♦ Alle Parameter in alphabetischer Übersicht

Parameterkennung	Parameterbezeichnung	interne Par. Nr.	Par.-Gruppe	Einheit	Init-Wert
SYNCFACFS	Synchronisationsfaktor Slave (M:S)	50	AXS	qc	1
SYNCFALT	Markeranzahl für Fault	57	AXS	–	10
SYNCFVEL	Geschwindigkeits-Feedforward [per mill von VCMD] für den Synchronisations-Modus.	109	AXS		0
SYNCMARKM	Markeranzahl Master	52	AXS	–	1
SYNCMARKS	Markeranzahl Slave	53	AXS	–	1
SYNCMFPAR	Markerfilterkonfiguration	17	AXS	–	0
SYNCMFTIME	Filterzeit für die Markerkorrektur	18	AXS	1 ms	0
SYNCMMAXCORR	Maximale Markerkorrektur	6	AXS	qc	0
SYNCMPULSM	Markerabstand Master	58	AXS	qc	500
SYNCMPULSS	Markerabstand Slave	59	AXS	qc	500
SYNCMSTART	Startverhalten für Markersynchronisation	62	AXS	–	0
SYNCMTYPM	Markertyp Master	60	AXS	–	0
SYNCMTYPS	Markertyp Slave	61	AXS	–	0
SYNCMWINM	Master-Marker Toleranzfenster	68	AXS	qc	0
SYNCMWINS	Slave-Marker Toleranzfenster	69	AXS	qc	0
SYNCOFFTIME	Offset Filterzeit	16	AXS	ms	0
SYNCPOSOFFS	Positionsoffset bei Positionssynchronisation	54	AXS	qc	0
SYNCREADY	Markeranzahl für Ready	56	AXS	–	1
SYNCSFTIME	Filterzeit [ms] für die Slave Marker Korrektur.	106	AXS	–	0
SYNCTYPE	Typ der Synchronisation	51	AXS	–	0
SYNCVELREL	Relative Geschwindigkeitsbegrenzung Slave	66	AXS	%	0
SYNCVFLIMIT	Synchronisationsfehler-Fenster [qc] für die automatische Deaktivierung von SYNCVFTIME.	110	AXS		0
SYNCVFTIME	Geschwindigkeitsfilter	65	AXS	τ_{filt} (μs)	0
TESTTIM	Messzeit im Zielfenster	24	AXI	ms	0
TESTVAL	Grenzwert für Messwert im Zielfenster	25	AXI	qc	1
TESTWIN	Größe Zielfenster	8	AXI	qc	0
TIMER	Abtastzeit für PID-Regelung und zum Senden des Ausgabewertes an den Antrieb	14	AXR	ms	1
VELKILIM	Grenzwert für die Integralsumme der PI Drehzahlregelung.	103	AXA	–	1000
VELKINT	Integralfaktor der PI Drehzahlregelung	91	AXA	–	5
VELKPROP	Proportionalfaktor der PI Drehzahlregelung	90	AXA	–	200
VELMAX	Maximalgeschwindigkeit	1	AXB	U/Min	1000
VELMAXOC	Maximalgeschwindigkeit (read only)	9	AXB	qc/st * 1/65536	
VELRES	Geschwindigkeitsteiler	22	AXB		100
VMENCMTYP	Virtueller Master Drehgeber-Markertyp.	118	GL	–	0
VMENCRES	Virtuelle Master Geberauflösung.	117	GL	qc	500

Parameter-Referenz ♦ Alle Parameter in alphabetischer Übersicht

Parameterkennung	Parameterbezeichnung	interne Par. Nr.	Par.- Gruppe	Einheit	Init-Wert
VMENCTYP	Definiert die Quelle des virtuellen Encoders und den Encodertyp der verwendet wird, falls der virtuelle Encoder deaktiviert wird.	116	GL	–	0

Achsparameter im Detail

DRIVETYPE - 0

Inhalt	Definiert die zu verwendende Art und den Typ des Sollwertausgangs für die Achsregelung.
Beschreibung	<p>DRIVETYPE legt den für die Achsregelung zu verwendenden Sollwert-Ausgang fest. Als Sollwertausgang können die folgenden „Ausgangskanäle“ dienen und definiert werden:</p> <ul style="list-style-type: none">- Lokaler analoger Ausgang der Steuerung- Analoger Ausgang auf einem CANopen I/O-Modul- Prozessdatenwort (PDO) für eine CANopen-Leistungsendstufe <p>Der Parameterwert von DRIVETYPE wird als ‚Device-Typ‘ bezeichnet. Die Nummer des Device-Typs „steht“ für die verwendete Antriebseinheit unter dem Gesichtspunkt der Sollwert-Vorgabe wie auch der Istwert-Feedbackquelle. Der Device-Typ wird so ebenfalls bei der Parameterwertberechnung von ENCODERTYPE, (27) und MENCODERTYPE (67) verwendet, falls die Drehgeberinformation über den Bus übertragen wird.</p>
DRIVETYPE = 0	Der Default Device-Typ ist 0. Dieses bedeutet für den Parameter DRIVETYPE, dass die Sollwertvorgabe über den lokalen analogen Ausgang der Steuerung erfolgt.
DRIVETYPE ≠ 0 = Sollwerte werden über einen Bus übertragen	Alle Einstellungen abweichend vom Defaultwert (0) bedeuten, dass die Sollwert-Information über den Bus (z.B. CAN-Bus) zum Teilnehmer (z.B. analoges CANopen-Modul oder CANopen-Leistungsendstufe) übertragen wird. Der Teilnehmer muss bei der Verwendung von Steuerungen mit mehreren, getrennten CAN-Bus-Kreisen (z.B. MACS3) zwingend an dem so genannten Slave-Bus angeschlossen sein. Die Bus-ID des Teilnehmers muss über den Parameter REVERS (63) festgelegt werden.
Behandlung der automatisch angelegten CAN-Objekte	<p>Für den Sollwert-Datenaustausch über den Bus werden die benötigten CAN-Objekte (PDOs, GUARD-, SYNC-Objekt) automatisch angelegt und der Teilnehmer initialisiert (NMT0). Diese im Hintergrund ausgeführte Objekt-Erzeugung entspricht im Wesentlichen dem CANINI Befehl. Im Gegensatz zur applikationsseitigen Objekt-Erzeugung mit dem CANINI Befehl, bleiben die mit DRIVETYPE automatisch erzeugten Objekte auch bei einem erneuten applikationsseitigen CANINI unverändert erhalten.</p> <p>Ebenfalls werden die mit DRIVETYPE automatisch erzeugten Objekte auch durch ein „CANDEL - 1“ nicht gestoppt. Die automatisch erzeugten Objekte können innerhalb der Applikation nur durch einen erneuten, abweichenden SET DRIVETYPE Befehl gelöscht bzw. umkonfiguriert werden. Auch beim Programmabbruch (mittels ESC) bleiben die Objekte noch bis zum nächsten Programmstart erhalten. Erst beim Neustart eines Programms werden die Objekte gelöscht und entsprechend der Einstellung im permanenten Parametersatz oder entsprechend der Definition im Programm neu angelegt.</p>
Guarding	Bei konfiguriertem Guarding (d.h. DRIVETYPE > 0), wird der Bus-Teilnehmer überwacht und eine fehlende Rückmeldung auf das GUARD-Objekt (z.B. aufgrund des Ausfalls des Teilnehmers) führt zu dem Fehler 88. Es liegt somit in der Verantwortung des Anwenders, darauf zu achten, dass zum Zeitpunkt der Definition des Parameters DRIVETYPE der Parameter REVERS (63) (entsprechend der Bus-ID) korrekt gesetzt ist, der Bus-Teilnehmer korrekt konfiguriert (Baudrate, Bus-ID), angeschlossen, eingeschaltet und für die Kommunikation bereit ist.
!!!	Falls bei einem DRIVETYPE > 0 der Bus-Teilnehmer ausfällt, muss von der Applikation (in der Fehlerbehandlungsroutine) durch das Setzen des negativen DRIVETYPE-Werts die Überwachung abgeschaltet werden, bis der Fehler auf Seiten des Teilnehmers behoben ist. Ansonsten tritt nach jedem Löschen des Steuerungsfehlerzustands 88 (mit ERRCLR) sofort wieder die identische Fehlermeldung auf.

Anpassung Reglerzykluszeit	Falls die Sollwert-Information(en) an Bus-Teilnehmer (CANopen-I/O-Module, CANopen-Leistungsendstufen) gerichtet sind, die das PDO nicht im 1 ms Takt empfangen und auswerten können, muss die Reglerzykluszeit mit dem SET TIMER Befehl entsprechend höher (z.B. Lenze auf 5 ms, CANopen-I/O-Modul 3 ms) gesetzt werden.
	!!! Es ist zu beachten, dass die Periodizität des SYNC-Telegramms bis zur Firmware-Version 6.5.16 ebenfalls durch den kleinsten Parameterwert von TIMER definiert war, der bei einer Achse eingestellt wurde. Bei aktuellen Firmware-Versionen (6.5.16 oder höher) erfolgt die Einstellung der SYNC-Rate über den globalen Parameter CANSYNCTIMER. Dieser Parameter muss auf einen Wert gesetzt werden, der auf den langsamsten CAN-Bus Teilnehmer abgestimmt ist.
PDO-Mapping für CANopen-Antriebe	Bei der Verwendung von CANopen-Leistungsendstufen muss das so genannte PDO-Mapping, d.h. welche Objekte der Leistungsendstufe in die übertragenen PDOs verlinkt sind, in der Applikation vor dem Setzen der Parameter DRIVETYPE und ENCODERTYPE ausgeführt werden. Die Wahl der am besten geeigneten Objekte hängt von der CANopen-Leistungsendstufe ab. Die zub machine control AG gibt hierzu gerne Beispiel-Sourcecode passend zu den unten genannten Device-Typen ab. Nennen Sie uns dafür bitte die eingesetzte Leistungsendstufe per e-Mail (info@zub.ch).
Analogen Ausgang eines CANopen I/O-Moduls verwenden	Um einen Verstärker anzutreiben kann statt des analogen Ausgangs der Steuerung auch der analoge Ausgang eines CANopen I/O-Moduls benutzt werden. Dazu wird der Parameter DRIVETYPE auf 91 – 94 gesetzt, abhängig von der Nummer des analogen Ausgangs der an diesem Modul (1..4) benutzt wird. Zusätzlich muss REVERS auf ID * 100 gesetzt werden, um der Steuerung mitzuteilen, welches I/O-Modul benutzt wird. Bei den meisten der I/O-Module sollte der TIMER auf den Wert 3 gesetzt werden, weil sie sich manchmal nicht korrekt verhalten, wenn sie jede ms ein PDO erhalten. Achten Sie darauf, dass die Baudrate entsprechend gesetzt ist und dass andere Module nicht den CAN-BUS überlasten.
	!!! Diese Funktion arbeitet nur mit dem <u>Slave-Bus</u> . Diese Funktion wird hauptsächlich von einer MACS3-2ax benutzt um die zweite Achse zu steuern. In diesem Fall wird normalerweise der Master-Encoder für die Rückführung (Feed-back) benutzt. (Dies ist der Default für ein 2-Achssystem). Wenn Sie ein 3-Achssystem haben, denken Sie daran, entweder die Achse 3 zu deaktivieren (MOTOR OFF x(3)) oder einen CAN-Encoder für Achse 3 zu definieren. Andernfalls würde die Achse 3 als Default den gleichen Encoder (Master) benutzen und das wird zu einem Schleppfehler auf Achse 3 führen, wenn die Achse 2 verfahren wird. Die folgenden Zeilen zeigen wie die Funktion benutzt wird: canid = 3 MOTOR OFF x(2) SET TIMER 1 / 3 SET DRIVETYPE x(2) -91 // DRIVETYPE auf CAN analogen Ausgang nr. 1 ohne Guarding setzen SET REVERS x(2) (canid * 100) // CAN I/O Modul mit ID canid benutzen MOTOR ON x(2)
DRIVETYPE für Danfoss Antriebe	Mit „DRIVETYPE 10“ werden Danfoss-Antriebe unterstützt. Hierfür wird PDO-2 genutzt, da Mapping nicht unterstützt wird. Jeder DRIVETYPE kann kombiniert werden mit ±100. (Danfoss d. h. 110 oder -110). Das heißt, dass das Steuerwort (ControlWord) nicht mehr durch die Steuerung gesetzt wird. Weder MOTOR OFF noch andere Befehle beeinflussen weiterhin das Steuerwort. Die Anwendung ist für die Pflege des Steuerworts durch den Befehl OUTDA zuständig.

Siehe folgendes Beispiel:

```
#define FC_RUN      (0x047C)
#define FC_COAST   (0x0474)
#define FC_STOP    (0x063C) // Ramp 2 aktiviert

SET ERRCOND 5 // im Fall eines Fehler überhaupt nichts tun
PRINT "FC_RUN"
OUTDA FU1_PROCESS (FC_RUN << 16) // FU auf aktiven Modus setzen

CSTART
CVEL 50
DELAY 600

PRINT "jetzt stoppen "
OUTDA FU1_PROCESS (FC_STOP << 16) // FU mit RAMP 2 stoppen
DELAY 5 // warten bis FU das neue ControlWord gesehen hat
MOTOR OFF // nun Sollwert auf null setzen
DELAY 2000 // warten bis Herunterfahren beendet ist
PRINT "nun Freilauf "
OUTDA FU1_PROCESS (FC_COAST << 16) // FU auf Freilauf-Modus setzen
```

DRIVETYPE für Servostar Dieser DRIVETYPE benutzt PDO1 für die Kommunikation. Solche PDOs müssen wie folgt konfiguriert werden:

Rx – ControlWord (2 bytes), Referenz (4 bytes)
Tx – Statusword (2 bytes), Position (4 bytes)

Die folgenden CANopen-Befehle werden ausgegeben:

Error -> ShutDown (0x06)
Motor Off -> ShutDown (0x06)
Motor On -> Enable (0x0F)

Im Fall von ERRCOND = 4 (constref) führt der Fehler oder Motor off zu einem Schnellstopp (0x02). D.h. dass ERRCOND nach einem Schnellstopp auf 0 gesetzt und dann ein MOTOR OFF ausgeführt werden muss, um zu einem ReadyToSwitchOn Status zurückzukehren. Danach kann ERRCOND auf 4 zurückgesetzt und mit einem MOTOR ON in den betriebsbereiten Status (OperationEnable) gewechselt werden.

DRIVETYPE für Kollmorgen AKD Ab Firmware 6.7.53 werden mit dem neuen „DRIVETYPE 13“ PDOs mit Positions- und zusätzlichen Latching-Informationen unterstützt, wie sie beim Servoverstärker AKD von Kollmorgen zur Verfügung stehen:

RxPdo-1 enthält StatusWord(2), ActualPosition(2), LatchPosition(2), LatchStatus(2)
TxPdo-1 enthält ControlWord(2), CommandVelocity(4), LatchControl(2).

Die Geschwindigkeit (= CommandVelocity) wird in 1/1000 U/min übertragen. Um diese Geschwindigkeit zu berechnen, muss VELMAX korrekt gesetzt sein, bevor der Parameter DRIVETYPE gesetzt wird. Wenn VELMAX nicht dem geforderten Skalierungsfaktor entspricht, ist es auch möglich den Parameter AMPENCRPM zu verwenden, um die maximale Geschwindigkeit zu definieren.

Portabilität DRIVETYPE ab Controller Version 6.1.14, Compiler Version 6.1.0;
DRIVETYPE 91 .. 94 (analogen Ausgang eines CANopen I/O-Moduls nutzen) ab Version 6.4.41
DRIVETYPE 10 für Danfoss Drives ab Version 6.5.08
DRIVETYPE 11 für Servostar ab Version 6.6.32
DRIVETYPE 12 für Lenze 8400 ab Version 6.6.81
DRIVETYPE 13 für Kollmorgen Servostar ab Version 6.7.53

Parametergruppe (Wird nicht im Parameter-Dialogfenster angezeigt.)

Querverweise	ENCODERTYPE, MENCODERTYPE, REVERS, TIMER, CANSYNCTIMER
Einheit	-
Bereich	-94 ... 94
	= Device-Typen bei Parameterwertbereich von ENCODERTYPE (27) und MENCODERTYPE (67)
	0 = Standard Antrieb (analoger Ausgang)
	1 = CAN Antrieb des Typs Lenze
	2 = CANopen-Servoverstärker nach DS402
	3 = CANopen-Servoverstärker maxon EPOS (Hersteller-spezifisch erweiterter DS402-Modus)
	4 = CANopen-Servoverstärker zub DSA (Hersteller-spezifisch erweiterter DS402-Modus)
	5 = CAN-Antrieb des Typ Lenze mit einem maximalen Sollwert von 0x4000 (statt 0x5000), was Profibus-konform ist und beim Lenze Antrieb über den Parameter C011 konfiguriert werden kann.
	10 = Danfoss Drives (Verwendung von PDO 3) ±100 (110, -110) Steuerwort (ControlWord) ist <u>nicht</u> gesetzt
	11 = Servostar
	12 = Lenze 8400 (wie Danfoss Drives, d.h.. DRIVEYTYPE 10, aber Verwendung von PDO.)
	13 = Kollmorgen AKD
	91 = Analoger Ausgang 1 auf einem CANopen I/O-Modul
	92 = Analoger Ausgang 2 auf einem CANopen I/O-Modul
	93 = Analoger Ausgang 3 auf einem CANopen I/O-Modul
	94 = Analoger Ausgang 4 auf einem CANopen I/O-Modul
	!!! Negative Werte deaktivieren das Guarding!
	Die Bus-Teilnehmer müssen bei Verwendung von Steuerungen mit mehreren, getrennten CAN-Bus-Kreisen (z.B. MACS3) zwingend an dem so genannten Slave-Bus angeschlossen sein. Der Einbezug der Bus-Offset-Kennung im Parameterwert von DRIVEYTYPE entfällt durch diese feste Zuordnung.
Init-Wert	0

VELMAX - 1

Inhalt Maximalgeschwindigkeit

Beschreibung VELMAX definiert die Nenngeschwindigkeit des Antriebs in U/Min. Der Wert wird zur Berechnung von Rampen und Ist-Geschwindigkeiten benötigt.

!!! Die Nenngeschwindigkeit bezieht sich auf die Drehzahl des Drehgebers.

Parametergruppe AXB Grundeinstellungen

Einheit U/Min

Bereich 1 ... 65535

Init-Wert 1000

ENCODER - 2

Inhalt	Auflösung des Istwertgebers (Geberstrichzahl)	
Beschreibung	<p>Der Parameter Geberstrichzahl enthält die Auflösung des Istwertgebers (Inkremental- oder Absolutgeber) bezogen auf eine Drehgeberumdrehung.</p> <p>Aus dieser Information wird die Anzahl der Quadcounts [qc] pro Umdrehung berechnet. Quadcount ist die für alle Wegmessungen zugrunde liegende Einheit. Quadcounts werden durch die Auswertung aller Flanken von der A- und B-Spur erzeugt. Ein Geberstrich eines Inkrementalgebers entspricht dabei vier Quadcounts. Bei Absolutgebern werden die absoluten Werte 1:1 zurückgeliefert.</p> <p>Die Anzahl der Quadcounts pro Umdrehung wird für die Indexpulssuche bei der Referenzfahrt sowie für die Umrechnung von Geschwindigkeiten und Beschleunigungen in interne Einheiten benötigt.</p> <p>Die Geberstrichzahl liefert auch die Information, ob bei einer HOME- oder INDEX-Fahrt das Indexsignal verpasst wurde. Wenn mehr als eine volle Umdrehung ausgeführt wurde, ohne einen Indeximpuls zu registrieren, erfolgt eine entsprechende Fehlermeldung.</p>	
	!!!	Jede Änderung der Geberauflösung im Parameter ENCODER erfordert auch eine Anpassung der Parameter KPROP, KDER, KINT des Positionsreglers: Bei Verdopplung des Werts von ENCODER muss KPROP, KDER, KINT halbiert werden. Bei Halbierung des Werts von ENCODER muss KPROP, KDER, KINT verdoppelt werden, um ein identisches Verhalten des Positionsreglers zu erhalten.
Parametergruppe	AXE Gebersystem	
Querverweis	KPROP, KDER, KINT	
Einheit	Geberstrichzahl/Umdrehung	
Bereich	1 ... MLONG	
	!!!	M1 und neuere Steuerungen wie MACS Es sind keine negativen Werte erlaubt. Ob mit oder ohne Indexpuls gearbeitet wird, wird in Parameter HOME_TYPE (40) angegeben.
	!!!	MOCON, MK1, MODAK Bei allen Steuerungen bis Version 5 entscheidet das Vorzeichen, ob mit oder ohne Indexpuls gearbeitet wird. Wenn ohne Indexpuls gearbeitet wird, muss die Auflösung als negativer Wert eingetragen werden; im anderen Fall als positiver Wert.
Init-Wert	500	
Grenzwerte	Um eine einwandfreie Funktion der Steuerung zu gewährleisten, darf das Produkt aus der Geberstrichzahl ENCODER (2) und Maximalgeschwindigkeit in Geberumdrehungen/s die Grenzfrequenz der Drehgeber-Eingangsstufe nicht überschreiten. Die Grenzwerte für die einzelnen MACS-Steuerungen finden Sie in den entsprechenden Datenblättern.	

HOME_FORCE - 3

Inhalt Homefahrt erzwingen?

Beschreibung Wenn dieser Parameter auf Ja = 1 gesetzt ist, muss eine Homefahrt ausgeführt werden, bevor irgendeine andere Positionierfahrt ausgeführt werden kann.

Bei einem Fahrbefehl ohne erfolgreich ausgeführte Homefahrt wird der Fehler 6 ausgelöst.

!!! Aus Sicherheitsgründen und zur Vermeidung von Fehlpositionierungen sollte der Parameter immer auf 1 gesetzt und damit eine Homefahrt erzwungen werden. Es muss in diesem Fall jedoch berücksichtigt werden, dass dann alle Programme vor dem ersten Fahrbefehl einen HOME Befehl ausführen müssen, damit sie auch ohne angeschlossenen PC einwandfrei funktionieren.

Parametergruppe AXH Referenzierung

Einheit –

Bereich 0 ... 1

0 = Homefahrt nicht erzwingen.

Nach dem Einschalten gilt die Istposition als Realnullpunkt

1 = Homefahrt erzwingen

Nach dem Einschalten der Steuerung sowie nach dem Ändern von Achsparametern muss vor einem Fahrbefehl – ob direkt oder durch ein Programm ausgeführt – zwingend zuerst eine Homefahrt erfolgen.

Intern kann der Parameter auch den Wert 255 enthalten; dieser zeigt an, dass ein Anfahren der Home-Position erzwungen wurde und bereits erfolgt ist.

Init-Wert 0

NEGLIMIT - 4

Inhalt	Negative Software-Wegbegrenzung
Beschreibung	<p>NEGLIMIT gibt die negative Wegbegrenzung für alle Fahrbewegungen an. Wird dieser Wert überschritten, wird ein Fehler ausgelöst. NEGLIMIT ist nur aktiv, wenn SWNEGLIMACT (19) gesetzt ist.</p> <p>Ein Positionierbefehl, der außerhalb der eingestellten Grenzen liegt, wird nicht ausgeführt.</p> <p>!!! Bei der Verwendung des Befehls DEFORIGIN wird die Wegbegrenzung automatisch angepasst, sodass die ursprüngliche Position des Verfahrbereichs erhalten bleibt.</p> <p>!!! Die Wegbegrenzung wird immer in Quadcounts angegeben.</p>
Parametergruppe	AXI Vordefinierte Eingänge/Ausgänge
Einheit	qc
Bereich	-MLONG ... MLONG
Init-Wert	-500000

POSLIMIT - 5

Inhalt	Positive Software-Wegbegrenzung
Beschreibung	<p>POSLIMIT gibt die positive Wegbegrenzung für alle Fahrbewegungen an. Wird dieser Wert überschritten, wird ein Fehler ausgelöst.</p> <p>POSLIMIT ist nur aktiv, wenn SWPOSLIMACT (20) gesetzt ist. Ein Positionierbefehl, der außerhalb der eingestellten Grenzen liegt, wird nicht ausgeführt.</p> <p>!!! Bei der Verwendung des Befehls DEFORIGIN wird die Wegbegrenzung automatisch angepasst, so dass die ursprüngliche Lage des Verfahrbereichs erhalten bleibt.</p> <p>!!! Die Wegbegrenzung wird immer in Quadcounts angegeben.</p>
Parametergruppe	AXI Vordefinierte Ein-/Ausgänge
Einheit	qc
Bereich	-MLONG ... MLONG
Init-Wert	500000

SYNCMMAXCORR - 6

Inhalt	Maximale Markerkorrektur
Beschreibung	<p>SYNCMMAXCORR wird benutzt, um die maximale Korrektur, die durch die Markerkorrektur vorgenommen wird, zu begrenzen. Der Befehl arbeitet mit SYNCM und SYNCC. Der Wert wird durch den Anwender mit SET SYNCMMAXCORR in qc (Slave) eingegeben.</p> <p>Der Wert > 0 begrenzt die Markerkorrektur durch den vorgegeben Wert. Wenn also die Korrektur größer werden würde, ist sie auf diesen Wert begrenzt.</p> <p>Der Wert < 0 setzt den Parameter so, dass überhaupt keine Korrektur gemacht wird. Der Anwender kann so die Markerkorrektur ausschalten.</p> <p>!!! Wenn man SYNCMFTIME (18) oder SYNCFVTIME (65) (negativ) gesetzt hat, wird die Korrektur abhängig von diesen Faktoren über eine gewisse Zeit gedehnt.</p>
Portabilität	Unterstützung des Wertes < 0 ab Version 6.6.37.
Parametergruppe	AXS Synchronisation
Einheit	qc
Bereich	-MLONG ... MLONG
	0 = keine Begrenzung
	> 0 = begrenzt die Markerkorrektur auf den definierten Wert
	< 0 = keine Korrektur
Init-Wert	0

HOME_VEL - 7

Inhalt Homefahrt-Geschwindigkeit

Beschreibung HOME_VEL bestimmt die Geschwindigkeit, mit der die Fahrt zum Referenzschalter ausgeführt wird. Die Angabe ist auf die Nenngeschwindigkeit bezogen und von dem Parameter VELRES (22) abhängig. Standardgemäß ist die Angabe in % von der Nenngeschwindigkeit.

$$\text{Homefahrt-Geschwindigkeit [U/Min]} = \text{HOME_VEL (7)} \cdot \frac{\text{VELMAX (1)}}{\text{VELRES (22)}}$$

!!! Da immer in der gleichen Drehrichtung (abhängig vom Vorzeichen) nach dem Referenzschalter gesucht wird, sollte dieser an den Grenzen des Fahrbereichs angebracht werden. Nur so kann sichergestellt werden, dass sich der Antrieb bei einer Homefahrt aus allen Positionen auch tatsächlich in Richtung des Referenzschalters und nicht von ihm weg bewegt.

Um eine gute Repetierbarkeit der Referenzfahrt zu erhalten, sollte mit höchstens 10 % der maximalen Drehzahl gefahren werden.

Parametergruppe AXH Referenzierung

Einheit % (VELRES = 100)

Bereich -VELRES ... VELRES

Ein negatives Vorzeichen heißt, dass die Suche in der anderen Richtung erfolgt.

Init-Wert 10

TESTWIN - 8	
Inhalt	Größe des Zielfensters
Beschreibung	<p>TESTWIN gibt die Größe des Zielfensters an. Eine Position gilt erst dann als erreicht, wenn die Sollfahrt (Trapez) abgearbeitet ist, die Istposition innerhalb des Fensters liegt und die Geschwindigkeit kleiner ist als TESTVAL. (Voraussetzung: TESTWIN und TESTTIM sind aktiviert.)</p> <p>Hierbei ist die Geschwindigkeit TESTVAL in qc/TESTTIM angegeben.</p> <p>Die Steuerung wartet mit dem Ausführen des jeweils nächsten Befehls, bis die Istposition innerhalb des Zielfensters liegt.</p> <p>Wenn TESTWIN nicht aktiviert ist (Parameter 0), gilt das Ziel als erreicht, sobald die Sollposition gleich der Zielposition ist. Diese muss jedoch nicht mit der tatsächlichen Position des Antriebs übereinstimmen.</p> <p>!!! Wird das Zielfenster um die Endposition zu klein gewählt, könnte sich der Antrieb in einer sehr kleinen Umgebung um die Endposition bewegen, ohne das Zielfenster zu erreichen, so dass das Programm bei dem entsprechenden Positionierbefehl „hängen“ bleibt.</p> <p>Zielfenster „0“ deaktiviert die Überwachung der Istposition und überwacht lediglich die Sollposition.</p> <p>!!! <u>Modifizierte Behandlung</u> von TESTWIN um den Gebrauch von CANopen anzupassen: Wenn TESTTIM gesetzt ist, aber nicht TESTVAL, wird automatisch CANopen angenommen. In diesem Fall wird geprüft, ob die Zeit innerhalb von TESTWIN ist länger als TESTTIM. Falls dem so ist, ist die Position erreicht. Andernfalls ist die Position nicht erreicht worden.</p>
Portabilität	Modifizierte TESTWIN Behandlung für CANopen ab Version 6.6.53
Parametergruppe	AXI vordefinierte Eingänge/Ausgänge
Einheit	qc
Bereich	0 ... 10000 0 = aus
Init-Wert	0
	!!! TESTWIN muss immer kleiner als TESTVAL sein.

VELMAXQC - 9

Inhalt	Maximalgeschwindigkeit (Read only)
Beschreibung	<p>VELMAXQC bestimmt die <i>Maximalgeschwindigkeit</i>. Auf diese Geschwindigkeit beziehen sich alle Angaben, die mit Hilfe von VELRES (22) gemacht werden. Die Abtastzeit [st] beträgt 1 ms.</p> <p>VELMAXQC ist ein interner Parameter, der nicht verändert werden kann. Er wird automatisch aus VELMAX (1) berechnet, die der Benutzer in U/Min eingibt. Diese internen Werte sind entscheidend für den zulässigen Wertebereich. In der Praxis spielen diese Grenzwerte jedoch keine Rolle, da sie weit über der, durch die Hardware bestimmte Drehgeber-Eingangsfrequenz liegen.</p>
Parametergruppe	AXB (Wird nicht im Parameter-Dialogfenster angezeigt.)
Einheit	qc/[st] * 1/65536
Bereich	1 ... MLONG (Read only)

ACCMAXQC - 10

Inhalt	Maximale Beschleunigung (Read only)
Beschreibung	<p>ACCMAXQC legt die maximale Beschleunigung fest. Dies ist die Zeitspanne, die der Antrieb bei angeschlossener Last benötigt, um die Maximaldrehzahl zu erreichen. Auf diese Beschleunigung beziehen sich alle anderen Angaben, die über die Normierung VELRES (22) gemacht werden.</p> <p>Errechnet wird dieser interne Parameter automatisch aus RAMPMIN (31), die der Benutzer in ms eingibt.</p> <p>!!! Wird eine zu niedrige Anlaufzeit eingegeben, die eine unter den gegebenen mechanischen Bedingungen nicht erreichbare Beschleunigung bedingt, tritt im Betrieb typischerweise ein Schleppfehler auf.</p>
Parametergruppe	AXB (Wird nicht im Parameter-Dialogfenster angezeigt.)
Einheit	qc/st ² * 1/65536
Bereich	1 ... MLONG (Read only)

KPROP - 11

Inhalt	Proportionalwert für PID-Regelung				
Beschreibung	<p>Der Proportionalfaktor KPROP gibt den linearen Korrekturfaktor an, mit dem die Abweichung zwischen der aktuellen Soll- und Istposition bewertet und eine entsprechende Korrektur der Motordrehzahl vorgenommen wird.</p> <p>Faustregel: KPROP größer = Antrieb wird „steifer“ KPROP zu hoch = Neigung zum starken Überschwingen</p> <p>Typischerweise liegt das Verhältnis KPROP:KDER im Bereich von 4:1 bis 1:4 falls eine Leistungsstufe angesteuert wird, die als Drehzahlregler konfiguriert ist.</p> <p>!!! Der Wert des Parameters KPROP (ebenso wie KDER, KINT) muss neu angepasst werden, falls die Einstellung des Parameters ENCODER geändert wird. Beispiele:</p> <table><tr><td>Verdopplung des Werts von ENCODER</td><td>KRPOP muss halbiert werden</td></tr><tr><td>Halbierung des Werts von ENCODER</td><td>KRPOP muss verdoppelt werden</td></tr></table> <p>Beim Einsatz einer MACS-Steuerung mit integrierten Endstufen und einer Anpassung an der Reglerstruktur (= Parameter AMPMODE) muss der Wert des Parameters KPROP in der Regel neu optimiert werden.</p>	Verdopplung des Werts von ENCODER	KRPOP muss halbiert werden	Halbierung des Werts von ENCODER	KRPOP muss verdoppelt werden
Verdopplung des Werts von ENCODER	KRPOP muss halbiert werden				
Halbierung des Werts von ENCODER	KRPOP muss verdoppelt werden				
Parametergruppe	AXR Pos. Regelung				
Querverweis	KDER, KINT, KILIM, BANDWIDTH, ENCODER, AMPMODE				
Einheit	–				
Bereich	1 ... 65000				
Init-Wert	30				

KDER - 12

Inhalt	Differentialwert für PID-Regelung								
Beschreibung	<p>Der <i>Differentialfaktor</i> KDER ist der Korrekturfaktor, mit dem die Geschwindigkeit der Änderung eines Motorpositionsfehlers bewertet wird.</p> <p>Der <i>Differentialfaktor</i> wirkt der durch einen hohen Proportionalanteil verursachten Überschwingungsneigung entgegen und „dämpft“ das System. Ein zu groß gewählter Differentialfaktor führt jedoch zu einem „nervösen“ Antrieb und deutlich hörbarem Motorbrummen.</p> <p>Typischerweise liegt das Verhältnis KPROP:KDER im Bereich von 4:1 bis 1:4 falls eine Leistungsendstufe angesteuert wird, die als Drehzahlregler konfiguriert ist.</p> <p>!!! Der Wert des Parameters KDER (ebenso wie KPROP und KINT) muss neu angepasst werden, falls die Einstellung der Parameter ENCODER oder TIMER geändert wurde. Beispiele:</p> <table><tr><td>Verdopplung des Werts von ENCODER</td><td>KDER muss halbiert werden</td></tr><tr><td>Halbierung des Werts von ENCODER</td><td>KDER muss verdoppelt werden</td></tr><tr><td>Verdopplung des Werts von TIMER</td><td>KDER muss halbiert werden</td></tr><tr><td>Halbierung des Werts von TIMER</td><td>KDER muss verdoppelt werden</td></tr></table> <p>Beim Einsatz einer MACS-Steuerung mit integrierten Endstufen und einer Anpassung an der Reglerstruktur (= Parameter AMPMODE), muss der Wert des Parameters KDER in der Regel neu optimiert werden.</p>	Verdopplung des Werts von ENCODER	KDER muss halbiert werden	Halbierung des Werts von ENCODER	KDER muss verdoppelt werden	Verdopplung des Werts von TIMER	KDER muss halbiert werden	Halbierung des Werts von TIMER	KDER muss verdoppelt werden
Verdopplung des Werts von ENCODER	KDER muss halbiert werden								
Halbierung des Werts von ENCODER	KDER muss verdoppelt werden								
Verdopplung des Werts von TIMER	KDER muss halbiert werden								
Halbierung des Werts von TIMER	KDER muss verdoppelt werden								
Parametergruppe	AXR Pos. Regelung								
Querverweis	KPROP, KINT, KILIM, BANDWIDTH, ENCODER, TIMER, AMPMODE								
Einheit	–								
Bereich	0 ... 65000								
Init-Wert	0								

KINT - 13

Inhalt	Integralwert für PID-Regelung								
Beschreibung	<p>Der Integralfaktor KINT ist der Gewichtungsfaktor, mit dem im Zeitpunkt n die Summe aller Motorpositionsfehler bewertet wird.</p> <p>Der Integralanteil des PID-Filters bewirkt ein entsprechend zeitlich anwachsendes, korrigierendes Motordrehmoment. Durch den Integralanteil wird ein statischer Positionsfehler zu Null ausgeregelt, auch wenn eine konstante Last am Motor anliegt.</p> <p>Ein zu großer Integralfaktor führt jedoch zu einem „nervösen“ Antrieb. Die Höhe des Integralanteils kann mit dem Parameter KILIM begrenzt werden.</p> <p>Der Integralanteil der PID Positionsregelung kann in Abhängigkeit vom Bewegungszustand automatisch aktiviert oder deaktiviert werden. Das erforderliche Verhalten wird mit dem Parameter KILIMTIME konfiguriert. Ein automatische Aktivierung / Deaktivierung kann bei spezifischen Positionier- oder Synchronisationsanforderungen deutliche Vorteile für die Regelqualität im Stillstand und während der Bewegung bringen.</p> <p>!!! Der Wert des Parameters KINT (ebenso wie KDER, KPROP) muss neu angepasst werden, wenn die Einstellung der Parameter ENCODER oder TIMER geändert wurde. Beispiele:</p> <table><tr><td>Verdopplung des Werts von ENCODER</td><td>KINT muss halbiert werden</td></tr><tr><td>Halbierung des Werts von ENCODER</td><td>KINT muss verdoppelt werden</td></tr><tr><td>Verdopplung des Werts von TIMER</td><td>KINT muss verdoppelt werden</td></tr><tr><td>Halbierung des Werts von TIMER</td><td>KINT muss halbiert werden</td></tr></table> <p>Beim Einsatz einer MACS-Steuerung mit integrierten Endstufen und einer Anpassung an der Reglerstruktur (= Parameter AMPMODE), muss der Wert des Parameters KINT in der Regel neu optimiert werden.</p>	Verdopplung des Werts von ENCODER	KINT muss halbiert werden	Halbierung des Werts von ENCODER	KINT muss verdoppelt werden	Verdopplung des Werts von TIMER	KINT muss verdoppelt werden	Halbierung des Werts von TIMER	KINT muss halbiert werden
Verdopplung des Werts von ENCODER	KINT muss halbiert werden								
Halbierung des Werts von ENCODER	KINT muss verdoppelt werden								
Verdopplung des Werts von TIMER	KINT muss verdoppelt werden								
Halbierung des Werts von TIMER	KINT muss halbiert werden								
Parametergruppe	AXR Pos. Regelung								
Querverweis	KPROP, KDER, KILIM, KILIMTIME, BANDWIDTH, ENCODER, TIMER, AMPMODE								
Einheit	–								
Bereich	0 ... 65000								
Init-Wert	0								

TIMER - 14

Inhalt	Abtastzeit der PID Positionsregelung und zum Senden des Ausgabewertes an den Antrieb.
Beschreibung	<p>Der Parameter TIMER bestimmt die Abtastzeit der Positionsregelung und die Taktrate für die Erzeugung von SYNC-Telegrammen für CANopen-Servoverstärker und Frequenzumrichter.</p> <p>Erhöhen Sie zum Beispiel bei sehr kleinen Pulsfrequenzen wie 1 bis 2 qc per Abtastzeit den Wert der Werkseinstellung. Sie brauchen mindestens 10 bis 20 qc per Abtastzeit. Oder bei sehr trägen Systemen mit einer großen Totzeit. Würde man hier mit 1 ms regeln, würden große Motoren schwingen.</p> <p>Dieser Parameter beeinflusst, wie oft ein Ausgabewert zum Verstärker geschrieben wird. Für Antriebe, die am CAN-Bus hängen, muss er in Übereinstimmung mit den Leistungen des Antriebs justiert werden. (Einige Antriebe akzeptieren nicht jede ms neue Sollwerte.) In vielen Fällen liegt die maximal mögliche Update-Rate von Frequenzumrichtern im Bereich von 2 ... 4 ms.</p> <p>Bis zur Firmware-Version 6.5.16 hatte der Parameter TIMER ebenfalls die Periodenzeit für den Versand von SYNC-Telegrammen zur Anforderung der Status- und Encoder-Information des Antriebs bestimmt. Bei Mehrachs Anwendungen musste hierbei beachtet werden, dass der kürzeste eingestellte Wert von TIMER für die zyklische Generierung des SYNC-Signals bestimmend war. Da alle angeschlossenen CANopen-Geräte auf jedes SYNC reagieren, musste sich die TIMER-Einstellung am langsamsten Teilnehmer orientieren und dessen TIMER-Wert musste ebenfalls bei allen (!) anderen Antrieben identisch eingestellt werden.</p> <p>Ab der Firmware-Version 6.5.16 gibt es den globalen Parameter CANSYNCTIMER, der die Periodenzeit für den zyklischen Versand von SYNC-Telegrammen bestimmt. Der Wert des achs-spezifischen Parameters TIMER ist somit nicht mehr bestimmend für die Periodizität des SYNC-Telegramms, bestimmt aber weiterhin die Rate, mit welcher neue Sollwerte über Bus übertragen werden. Die Rate, mit welcher Sollwerte über den CAN-Bus übertragen werden, kann somit für jeden Servoverstärker und Frequenzumrichter unterschiedlich gewählt werden. Die Übertragungsrate der Rückinformation (Status, Encoder-Position) der externen Leistungsstufe wird über den globalen Parameter CANSYNCTIMER definiert. Falls die Encoder-Information über den Bus übertragen wird, ist hierbei zu beachten, dass in jedem Positionsreglertakt auch eine aktualisierte Positionsinformation zur Verfügung steht. Dies bedeutet, dass die Einstellung des globalen Parameters CANSYNCTIMER entsprechend dem Takt des Positionsreglers und der Sollwert-Übernahme durch die Leistungsstufe erfolgen muss.</p> <p>!!! Jede Änderung des Parameters TIMER erfordert auch eine Anpassung der Parameter KDER und KINT des Positionsreglers:</p> <ul style="list-style-type: none">Bei Verdopplung des Werts von TIMER muss KDER halbiert werden.Bei Halbierung des Werts von TIMER muss KDER verdoppelt werden.Bei Verdopplung des Werts von TIMER muss KINT ebenfalls verdoppelt werden.Bei Halbierung des Werts von TIMER muss KINT ebenfalls halbiert werden. <p>Die Einstellung von KPROP muss nach einer Änderung von TIMER jedoch nicht angepasst werden. Zwischen diesen beiden Parametern besteht keine gegenseitige Abhängigkeit</p>
Portabilität	Ab Firmware-Version 6.5.16 ist die Einstellung von TIMER ist nicht mehr bestimmend für den zyklischen Versand von SYNC-Telegrammen!
Parametergruppe	AXR Pos. Regelung
Querverweise	KPROP, KDER, KINT, DRIVETYPE, CANSYNCTIMER
Einheit	ms
Bereich	1 ... 1000
	Sinnvollerweise setzt man den Wert nicht höher als 1000 (= 1 s). Das wäre bereits eine sehr träge Regelung.
Init-Wert	1

POSERR - 15

Inhalt Maximal tolerierter Positionsfehler

Beschreibung Der *maximale Schleppabstand* POSERR definiert die erlaubte Toleranz zwischen der aktuellen Istposition und der errechneten Sollposition. Wird der mit POSERR definierte Wert überschritten, wird die Lageregelung aller Achsen abgeschaltet und ein Schleppfehler ausgelöst.

!!! Der Schleppabstand hat keinen Einfluss auf die Positioniergenauigkeit, sondern bestimmt lediglich, wie exakt der theoretisch errechnete Verfahrenweg eingehalten werden muss, ohne dass ein Fehler ausgelöst wird.

!!! Der Schleppabstand darf aus Sicherheitsgründen nicht zu groß gewählt werden, um Mensch und Maschine nicht zu gefährden.

Zu kleine Werte für den Schleppabstand können hingegen häufige Fehlermeldungen zur Folge haben. Als Richtwert kann die vierfache Strichzahl des Drehgebers angesetzt werden, was wiederum einer Drehgeberumdrehung entspricht.

Parametergruppe AXR Pos. Regelung

Einheit qc

Bereich 1 ... MLONG

Init-Wert 20000

SYNCOFFTIME - 16

Inhalt	Offset Filterzeit (1. Aufsynchronisieren; 2. neuer Offset)
Beschreibung	<p>Die Offset-Filterzeit beeinflusst auch die Art, wie ein neuer SYNCPOSOFFS Wert gehandhabt wird. Der Offset, der ausgeführt werden muss, wird Schritt für Schritt realisiert. Eine Schrittweite, die pro Abtastperiode (ms) auszuführen ist, wird wie folgt berechnet:</p> $\text{Schrittweite} = \frac{\text{SYNCMPULSM}}{\text{SYNCOFFTIME (Integer Anteil)}}$ <p>Daher wird es also SYNCOFFTIME dauern, um einen Offset von SYNCMPULSM. auszuführen. SYNCOFFTIME beeinflusst auch die Marker-Startkorrektur und die Korrektur der Markerfehler (siehe SYNCMFTIME).</p> <p>SYNCOFFTIME definiert die Zeit die benutzt werden soll, um einen Markerabstand auszugleichen.</p> <p>Ab Firmware Version 6.6.49 wird nicht nur der Offset über die gegebene Zeit verteilt, sondern es wird auch eine trapezförmige Bewegung berechnet und zur normalen Synchronisationsfahrt hinzugefügt.</p> <p>Damit werden begrenzte Beschleunigungen und Geschwindigkeiten berechnet, mit der Annahme dass normalerweise 20 % für Beschleunigung, 60 % mit konstanter Geschwindigkeit und weitere 20 % der Zeit zum Bremsen benutzt werden. Basierend auf dieser Auslegung wird berechnet, welche Beschleunigungs- und Verzögerungsrampen benutzt werden sollen und wie die maximale Geschwindigkeit begrenzt werden muss, um dies zu erreichen.</p> <p>Diese neue Verteilung wird nicht nur für Offsets benutzt, sondern auch für Startkorrekturen (erste Markerkorrektur), normale Markerkorrekturen und kumulierte Fehler, die durch eine REVERS Einstellung verursacht werden.</p>
Portabilität	Standardbefehl ab Controller Version 6.2.35, trapezförmige Fahrt ab Firmware 6.6.49
Parametergruppe	Achsparameter Synchronisation AXS
Einheit	ms
Bereich	0 ... MLONG
Init-Wert	0

SYNCMFPAR - 17

Inhalt	Markerfilter Konfiguration
Beschreibung	<p>Dieser Parameter wird benutzt, um das Verhalten des Markerfilters zu beeinflussen, siehe SYNCMFTIME (18) und SYNCSFTIME (106).</p> <p>Wenn SYNCMFTIME gesetzt ist wird MMarkerDist ein wenig anders berechnet. Wenn die gesetzte Zeit zu einer Anzahl von Markern führt, die kleiner ist als 100 ist, werden mindestens 100 genommen um den durchschnittliche Markerabstand (Filter) zu berechnen. Dies betrifft nur die Berechnung des Markerabstands. Für die anderen Filterberechnungen (Abweichung) wird weiterhin die Anzahl der Marker verwendet, die durch SYNCMFTIME und aktueller Master-Geschwindigkeit berechnet wurden.</p> <p>Siehe auch Abbildung Marker Correction im Kapitel „Technische Referenz“.</p>
Portabilität	<p>Standardbefehl ab Controller Version 6.2.35 (in früheren APOSS-Versionen konnte dieser Parameter durch ORGACC gesetzt werden);</p> <p>veränderte Berechnung des Markerabstands (wenn Marker < 100) ab Version 6.7.16.</p>
Parametergruppe	AXS Synchronisation
Einheit	–
Bereich	<p>0 oder 1 oder 2 oder 4 oder 16 oder 64</p> <p>Die folgenden Werte sind Bit-Wertigkeiten und können kombiniert werden:</p> <p>0 = Normale Funktion des Filters, siehe <i>Filterzeit für Markerkorrektur</i> SYNCMFTIME (18).</p> <p>1 = Statt der dynamischen Markerfilter-Konstante wird ein fester Wert von SYNCMFTIME / 300 verwendet.</p> <p>2 = Getriebekorrektur wird <u>nicht</u> durchgeführt.</p> <p>4 = Zur Berechnung der Zeitkonstante für den Filter des Korrekturwertes (G_Korrektur) wird die <i>Filterzeit für Markerkorrektur</i> anstelle der <i>Offset Filterzeit</i> SYNCOFFTIME (16) verwendet.</p> <p>16 = Es wird nicht der gefilterte Markenabstand und die Abweichung berechnet, sondern lediglich der Korrekturwert mittels eines PT-Filters geglättet. Zeitkonstante für diesen Filter gemäß Bitwertigkeit 4 (siehe oben).</p> <p>64 = Marker Mittelung und Marker Check werden auch durchgeführt wenn SYNCM nicht aktiv ist.</p> <p>Für weitere Beschreibungen der Auswahlmöglichkeiten siehe SYNCMFTIME (18).</p>
Init-Wert	0

SYNCMFTIME - 18

Inhalt	Filterzeit für Markerkorrektur
Anwendungsbeispiel	<p>Bei der Zeitungsproduktion wird diese Art des Filters benötigt, um eine Förderkette zu synchronisieren. Da die Zeitungen nicht völlig regelmäßig aus der Druckmaschine kommen, wären die Bewegungen der Kette sehr hart und dynamisch, falls man ohne Filter synchronisieren würde. Mit allen anderen Arten des Filterns würde das System beginnen in sinusförmigen Wellen zu schwingen.</p> <p>Benutzt man aber diese komplexe Filtermethode, arbeitet die Synchronisation sehr gut und löst das Problem.</p>
Beschreibung	<p>SYNCMFTIME wird in ms eingegeben und wie folgt benutzt:</p> <p>!!! Der Master-Geschwindigkeitsfilter SYNCVFTIME (65) wird zur besseren Auflösung in 1/1000 ms eingegeben, der Markerfilter (SYNCMFTIME) dagegen in Einheiten von 1 ms.</p> <p>Beispiel:</p> <pre>SET SYNCVFTIME -50000 SET SYNCMFTIME 2000</pre> <p>Das heißt, dass die Master-Geschwindigkeit über eine Periode von 50 ms gemittelt wird. Ein Markerfehler wird also innerhalb von 2000 ms korrigiert.</p> <p>Der aktuelle gefilterte Markerabstand kann mit SYSVAR Index 4238 ausgelesen werden, wenn dieser Filter durch das Setzen von SYNCMFTIME aktiviert wurde. Um diesen Filterwert zu erhalten, wird intern berechnet, wie oft Marker passieren, wenn man mit der maximal erlaubten Geschwindigkeit über eine Periode von SYNCMFTIME fährt.</p> <p>SYNCMFTIME und die Parameter SYNCOFFTIME (16) und SYNCMFPPAR (17) werden benutzt um das Verhalten des Markerfilters zu beeinflussen (siehe unten).</p> <p>Das Filtern wird wie folgt gehandhabt:</p>
Markerfilter-Berechnung nur wenn SYNCMFTIME > 0	<p>Wenn SYNCMFPPAR = 1</p> <p>kann jedes Mal, wenn ein echter Master-Marker erkannt wird, die Markerfilter-Konstante (MarkerFilter) als SYNCMFTIME/300 berechnet werden.</p> <p>Wenn SYNCMFPPAR = 0,</p> <p>kann jedes Mal, wenn ein echter Master-Marker erkannt wird, die Markerfilter-Konstante wie folgt berechnet werden:</p> $\text{Gefilterte alte Master Geschwindigkeit} * \frac{\text{SYNCMFTIME}}{\text{SYNCMPULSM} * 3}$ <p>das heißt, dass die Markerfilter-Konstante als Zeitkonstante zum Filtern benutzt wird. Dann sollte die benötigte Zeit für eine Ausgabe entsprechend eines konstanten Eingangswertes nahezu SYNCMFTIME sein.</p> <p>Die ausführliche Beschreibung des Markerfilters und seine Wirkungsweise lautet:</p> <pre>IF (SYNCMFPPAR & 1) THEN MarkerFilter = max(SYNCMFTIME / 300, 1) ELSE MarkerFilter = Oldmfilvel * SYNCMFTIME/(SYNCMPULSM*3)</pre> <p>das heißt, wenn MarkerFilter in einem PT-Glied als Tau benutzt wird, dann sollte die benötigte Zeit für eine Ausgabe entsprechend eines konstanten Eingangswertes nahezu SYNCMFTIME sein. Die Berechnung ist notwendig, weil der Filter (PT) bei jedem Marker ausgeführt wird und nicht jede ms.</p> <p>ENDIF</p>

Dieser Markerfilter (MarkerFilter) wird nun benutzt, um den Markerabstand (LastMMDist) mit einem PT-Element zu filtern. Das Ergebnis wird im gefilterten Markerabstand (MmarkerDist) gespeichert. Mit diesem Ergebnis wird die notwendige Getriebekorrektur (MmarkCorr) wie folgt berechnet:

$$MMarkCorr = (\text{SYNCPULSM} - \text{MMarkerDist}) / \text{MMarkerDist}$$

oder mit anderen Worten:

$$\text{Getriebekorrektur} = \frac{\text{SYNCPULSM} - \text{gefilterten Markerabstand}}{\text{gefilterten Markerabstand}}$$

Filter Master-Geschwindigkeit und Getriebekorrektur

Pro Abtastperiode wird die Master-Geschwindigkeit neu berechnet (Differenz der aktuellen zur letzten Master-Position).

IF (SYNCFVTIME < 0)

wird die gefilterte alte Master-Geschwindigkeit (Oldmfilvel) durch das Ausführen eines PT-Elements berechnet (Tau = SYNCFVTIME / 1000)

ELSE

Oldmfilvel = Master-Geschwindigkeit

andernfalls wird die gefilterte alte Master-Geschwindigkeit gleich der aktuellen gesetzt.

ENDIF

Wenn SYNCFMFTIME > 0 und SYNCFMFPAR = 2 wird die Getriebekorrektur durchgeführt, indem zur aktuellen Getriebeübersetzung die mit der Getriebeübersetzung multiplizierte Master-Geschwindigkeit addiert wird.

Tmpdelta = Oldmfilvel

IF (SYNCFMFTIME > 0 und !(SYNCFMFPAR & 2)) THEN

Die Getriebekorrektur wird wie folgt vorgenommen:

Tmpdelta = Tmpdelta + Master-Geschwindigkeit * MMarkCorr

ENDIF

Startkorrektur nur wenn SYNCFMFTIME > 0

Die Startkorrektur ist jene Korrektur, die ausgeführt werden muss, sobald die Startbedingungen erfüllt sind. Das heißt, entweder mussten die ersten zwei Marker beobachtet werden (SYNCFMSTART 1,6) oder es musste die Master-Geschwindigkeit erreicht und zusätzlich die ersten zwei Marker beobachtet werden (SYNCFMSTART 2,3,4,5). Diese Startkorrektur wird so aufgeteilt, dass sie nach SYNCOFFTIME erledigt sein wird. (Derzeit wird sie durch die Anzahl der Marker geteilt, die in SYNCOFFTIME mit der aktuellen Master-Geschwindigkeit passiert wurden und der erhaltene Wert wird zur normalen Markerkorrektur addiert, siehe unten; das Ergebnis wird in PFG_STARTKORRVAL gespeichert.)

Wenn SYNCOFFTIME == 0 wird die Startkorrektur sofort ausgeführt, das bedeutet, dass STARTKORRVAL die gesamte Startkorrektur beinhaltet und die Korrektur innerhalb von zwei Markern erledigt ist.

Markerkorrektur SYNCFMFTIME > 0

Frühere Version (6.2.20 – 6.2.33) siehe „Online-Hilfe“.

Die Korrektur des Markerfehlers wird wie folgt durchgeführt:

Zuerst wird vom Markerfehler (PFG_KORRUNFILT) die verbleibende Startkorrektur abgezogen (PFG_STARTKORRREST). Dann wird ein PT-Filter für den restlichen Fehler benutzt (Tau = MarkerFilter) und der Wert von PFG_STARTKORRVAL zum Ergebnis von PT addiert. Das Ergebnis wird wie immer in PFG_KORREKTUR gespeichert. Dann wird versucht, diese Markerkorrektur über einen Markerabstand zu spreizen. Das wird durch Teilen der Korrektur durch die Anzahl der Samples erreicht, die notwendig ist, um einen Markerabstand mit der aktuellen Master-Geschwindigkeit zu passieren. Der Wert wird in PFG_KORRVAL gespeichert und bei jeder Abtastperiode benutzt, um die berechnete Slave-Position zu korrigieren.

Folgende SYNCMFPAR Anpassung ist inbegriffen:

SYNCMFPAR & 4 → Offset Filter wird statt Marker Filter für die PT-Korrektur benutzt

SYNCMFPAR & 16 → Es wird kein PT-Filter benutzt. Stattdessen wird entweder G_MDEFDIST / (kein Marker innerhalb SYNCOFFTIME) benutzt oder G_KORREKTUR, je nachdem welcher kleiner ist.

Ab Version (6.2.35)

Zuerst wird vom Markerfehler (PFG_KORRUNFILT) die verbleibende Startkorrektur abgezogen (PFG_STARTKORREST). Dann wird die Korrekturfilterzeit (PFG_KORRTAU) entsprechend dem Parameter SYNCOFFTIME zum Filtern gesetzt (abhängig von der Master-Geschwindigkeit, siehe Startkorrektur Markeranzahl).

Nun wird die Summe aller Markerabstandsfehler (PFG_INTMMERROR) in ein PT (Tau = PFG_MARKERFILTER) geschrieben um die gefilterte Summe zu berechnen (PFG_MMARKERR). Danach wird die gefilterte Summe der Fehler von der ungefilterten abgezogen. Dieses Ergebnis wird schließlich benutzt, um die Markerkorrektur zu korrigieren.

Diese bereinigte Korrektur wird in PT (Tau = PFG_KORRTAU) geschrieben. Das Ergebnis dieses Korrekturfilters (PT) wird in PFG_KORREKTUR gespeichert (plus dem Anteil der Startkorrektur, falls notwendig).

Dann wird versucht, diese Markerkorrektur über einen Markerabstand zu spreizen. Das wird durch Teilen der Korrektur durch die Anzahl der Samples erreicht, die notwendig sind, um einen Markerabstand mit der aktuellen Master-Geschwindigkeit zu passieren. Der Wert wird in PFG_KORRVAL gespeichert und bei jeder Abtastperiode benutzt, um die berechnete Slave-Position zu korrigieren.

Folgende SYNCMFPAR Werte verändern das Verhalten:

SYNCMFPAR & 4 → PFG_KORRTAU wird gesetzt, um PFG_MARKERFILTER statt SYNCOFFTIME zu benutzen.

SYNCMFPAR & 16 → Es wird keine den Markerabstand betreffende Korrektur durchgeführt,

Markerkorrektur
SYNCMFTIME = 0

In diesem Fall enthalten _KORRUNFILT und _KORREKTUR den gleichen Wert, der auch die Startkorrektur im ersten Fall (Markerkorrektur bei SYNCMFTIME > 0) einschließt. Dieser Wert wird wie folgt gehandhabt:

```
IF (SYNCVFTIME < 0) THEN
  _KORRVAL = _KORREKTUR / max((-SYNCVFTIME / 100) , 1)
ELSE
  _KORRVAL = 0
ENDIF
```

Im ersten Fall, wenn die Markerkorrektur > 0, wird die Korrektur über ein Zeit von (-SYNCVFTIME / 100) ms gespreizt.

Im zweiten Fall wird die Korrektur sofort zur Sollposition addiert.

In jedem Fall wird die Reaktion durch die aktuelle Beschleunigung und Verzögerung begrenzt.

Berechnung
Markerabstand

Wenn SYNCMFTIME gesetzt ist, wird MMarkerDist ein wenig anders berechnet. Wenn die angesetzte Zeit zu einer Anzahl von Markern führt, die kleiner ist als 100 ist, dann werden mindestens 100 genommen um den durchschnittliche Markerabstand (Filter) zu berechnen. Dies betrifft nur die Berechnung des Markerabstands. Für die anderen Filterberechnungen (Abweichung) wird weiterhin die Anzahl der Marker verwendet, die durch SYNCMFTIME und aktueller Master-Geschwindigkeit berechnet wurden.

Siehe auch Abbildung Marker Correction in Kapitel „Technical Referenz“.

Portabilität	Standardbefehl mit Software Version 6.2.20 aufwärts; ab 6.2.35 Modifikation wie oben erläutert. Wenn SYNCMFTIME = 0, verhält sich das System genauso wie bis zur Controller Version 6.1.14, d.h. die Filterzeit für die Markerkorrektur hängt vom Wert des Parameters SYNCVFTIME (65) ab. Änderung der Berechnung des Markerabstands (Wenn Marker < 100) ab Version 6.7.16.
Parametergruppe	AXS Synchronisation (Noch nicht im Parameter-Dialogfenster realisiert.)
Einheit	ms
Bereich	-MLONG ... MLONG 0 = Wenn SYNCVFTIME (65) negativ ist, wird die Markerkorrektur durch SYNCVFTIME /100 gespreizt.
Init-Wert	0

SWNEGLIMACT - 19

Inhalt	Negative Software-Wegbegrenzung aktiv.
Beschreibung	<p>Durch Setzen dieses Parameters auf „1“ wird die Überwachung der Negativen Wegbegrenzung in der Steuerung aktiviert. Dann wird bei jeder Bewegung überprüft, ob die Zielposition außerhalb des zulässigen Verfahrbereichs liegt. Wenn dies der Fall ist, wird eine Fehlermeldung ausgelöst und die Antriebsregelung abgeschaltet.</p> <p>Im Positioniermodus bedeutet dies, dass der entsprechende Positioniervorgang nicht gestartet wird und der Fehler durch einen ERRCLR Befehl behoben werden kann.</p> <p>Im Synchronisations- und Drehzahlmodus kann der Fehler erst beim Überfahren des Endschalters erkannt werden, wodurch sich der Antrieb beim Auftreten der Fehlermeldung bereits außerhalb des zulässigen Verfahrbereichs befindet.</p> <p>Es ist möglich, den Software-Endschalterfehler zu löschen und dann in die entgegengesetzte Richtung zu fahren. Wenn dann aber erneut versucht wird, in die falsche Richtung zu fahren, tritt der Fehler 98 F_LIMIT_VIOLATION auf. In diesem Fall müssen Sie den Antrieb von Hand wieder in den zulässigen Bereich zurück bewegen und den Fehler löschen, oder im Menü Steuerung → Parameter → Bearbeiten vorübergehend die entsprechende Software-Wegbegrenzung abschalten und dann den Fehler löschen.</p>
Portabilität	Verbessertes Verhalten der Endschalter ab Version Firmware 6.6.60
Parametergruppe	AXI vordefinierte Eingänge/Ausgänge
Einheit	–
Bereich	0 ... 1 0 = inaktiv 1 = aktiv
Init-Wert	0

SWPOSLIMACT - 20

Inhalt	Positiver Software-Begrenzung aktiv
Beschreibung	<p>Durch Setzen dieses Parameters auf „1“ wird die Überwachung der Positiven Software-Wegbegrenzung in der Steuerung aktiviert. Dann wird bei jeder Bewegung überprüft, ob die Zielposition außerhalb des zulässigen Verfahrbereichs liegt und im gegebenen Fall eine Fehlermeldung ausgelöst und die Antriebsregelung abgeschaltet.</p> <p>Im Positioniermodus bedeutet dies, dass der entsprechende Positioniervorgang nicht gestartet wird und der Fehler durch einen ERRCLR Befehl behoben werden kann.</p> <p>Im Synchronisations- und Drehzahlmodus kann der Fehler erst beim Überfahren des Endschalters erkannt werden, wodurch sich der Antrieb beim Auftreten der Fehlermeldung bereits außerhalb des zulässigen Verfahrbereichs befindet.</p> <p>Es ist möglich, den Software-Endschalterfehler zu löschen und dann in die entgegengesetzte Richtung zu fahren. Wenn dann aber erneut versucht wird, in die falsche Richtung zu fahren, tritt der Fehler 98 F_LIMIT_VIOLATION auf. Dann müssen Sie den Antrieb von Hand wieder in den zulässigen Bereich zurück bewegen und den Fehler löschen, oder im Menü Steuerung → Parameter → Bearbeiten vorübergehend die entsprechende Software-Wegbegrenzung abschalten und den Fehler löschen.</p>
Portabilität	Verbessertes Verhalten der Endschalter ab Version Firmware 6.6.60
Parametergruppe	AXI vordefinierte Eingänge/Ausgänge
Einheit	–
Bereich	0 ... 1 0 = inaktiv 1 = aktiv
Init-Wert	0

KILIM - 21

Inhalt	Grenzwert für Integralsumme für PID-Regelung
Beschreibung	<p>Das <i>Integrationslimit</i> KILIM der <i>Positionsregelung</i> begrenzt die Integralsumme, um zu verhindern, dass durch einen hohen aufaddierten Fehler eine zu starke Gegenregelung erfolgt, die das System zum Schwingen bringen würde.</p> <p>Ein Wert von 0 bedeutet, dass der Integralanteil deaktiviert ist.</p> <p>Der Integralanteil der PID Positionsregelung kann in Abhängigkeit vom Bewegungszustand automatisch aktiviert oder deaktiviert werden. Das erforderliche Verhalten wird mit dem Parameter KILIMTIME konfiguriert. Ein automatische Aktivierung / Deaktivierung kann bei spezifischen Positionier- oder Synchronisationsanforderungen deutliche Vorteile für die Regelqualität im Stillstand und während der Bewegung bringen.</p>
Parametergruppe	AXR Pos. Regelung
Querverweise	KPROP, KDER, KINT, KILIMTIME
Einheit	–
Bereich	0 ... 65000
Default	0

VELRES - 22	
Inhalt	Geschwindigkeitsteiler
Beschreibung	Der <i>Geschwindigkeitsteiler</i> VELRES definiert eine Bezugsgröße für die Geschwindigkeitswerte der Fahrbefehle und Parameter. Die Angabe der Geschwindigkeit und Beschleunigung kann dann in ganzen Zahlen, bezogen auf diese Normierung, erfolgen. Der Wert 100 bedeutet, dass sich die Angaben in den Befehlen auf 100 beziehen, also in Prozent.
Parametergruppe	AXB Grundeinstellungen
Einheit	–
Bereich	1 ... 10000
Init-Wert	100

POSFAC_Z - 23

Inhalt Zähler Benutzerfaktor
bzw. im CAM-Mode Umrechnung der Einheiten qc in BE

Beschreibung Wegangaben in Fahrbefehlen erfolgen in Benutzereinheiten und werden intern in Quadcounts umgerechnet. So ist es durch eine entsprechende Wahl dieser Normierungsgröße möglich, mit beliebigen technischen Maßangaben (zum Beispiel mm) zu arbeiten.

Der Faktor ist ein Bruch, der sich aus Zähler und Nenner zusammensetzt:

$$\text{Parameter POSFACT}_Z (23) / \text{POSFAC}_N (26) = 1 \text{ Benutzereinheit BE}$$

Die Normierung bestimmt, wie viele Quadcounts eine Benutzereinheit ergeben: Wenn der Faktor zum Beispiel 50375/1000 beträgt, entspricht 1 BE genau 50,375 qc.

Im CAM-Mode wird der Parameter benutzt, um die Einheit für den Slave-Antrieb festzulegen, damit man im CAM-Editor mit sinnvollen Einheiten arbeiten kann. Siehe Beispiel 2.

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} \text{qc} = 1 \text{ BE}$$

vorausgesetzt dass:

$$\text{Getriebefaktor} = \frac{\text{Motorumdrehungen}}{\text{Umdrehung am Abtrieb}}$$

Encoder = Inkrementalgeber (bei Absolutgebern entfällt der Multiplikator 4)

Skalierfaktor = Anzahl der Benutzereinheiten BE [qc], die einer Umdrehung am Antrieb entsprechen

Außerdem kann man die Kurven mit diesem Faktor stauchen oder strecken, ohne jeweils neue Kurven definieren zu müssen. Die Verwendung von Zähler und Nenner für den Getriebefaktor führt zu einem sehr präzisen Ergebnis, da in fast allen Fällen Übersetzungen als Bruch darstellbar sind.

Portabilität Der Faktor ist ab Version 6.4.43 nicht mehr begrenzt auf kleine Werte, siehe auch [Skalierung](#) in Kapitel „Programmieren mit APOSS“.

Parametergruppe AXE Gebersystem

Einheit –

Bereich 1 ... MLONG/max. Position [BE]

Der obere Grenzwert hängt von der maximalen Zielposition ab. Es gilt hierbei folgender Zusammenhang:

$$\begin{aligned} \text{max. Position [BE]} * \text{POSFAC}_Z &< \text{MLONG} \\ \text{wobei POSFACT}_Z &= \text{Benutzerfaktor} * 1000 \end{aligned}$$

Beispiel: POSA max. Position [BE]

Init-Wert 1000

Beispiel 1 Welle oder Spindel:

25 Motorumdrehungen ergeben 1 Spindelumdrehung; Getriebefaktor = 25/1

Drehgeberauflösung (Inkrementalgeber) = 500

Spindelsteigung = 1 Umdrehung der Spindel = 5 mm

Skalierfaktor, wenn mit 1/10 mm Auflösung gearbeitet werden soll = 5 * 10 = 50

$$\frac{25/1 * 500 * 4}{50} \text{qc} = \frac{25 * 10 * 4}{1} \text{qc} = \frac{1000}{1} \text{qc} = 1 \text{ BE}$$

Benutzerfaktor Zähler (23) = 1000

Benutzerfaktor Nenner (26) = 1

Beispiel 2 Walze:

Getriebefaktor = 5/1

Drehgeberauflösung (Inkrementalgeber) = 500

Eine Walzenumdrehung beträgt 360 Grad. In diesem Beispiel soll mit einer Auflösung von 1/10 Grad gearbeitet werden. Das bedeutet, dass eine Walzenumdrehung in 3600 Arbeitseinheiten eingeteilt wird:

Skalierfaktor = 3600

$$\frac{5/1 * 500 * 4}{3600} qc = \frac{5 * 500 * 4}{3600} qc = \frac{25}{9} qc = 1 \text{ BE}$$

$$1 \text{ BE} = \frac{\text{POSFACT_Z (23)}}{\text{POSFACT_N (26)}}$$

Benutzerfaktor Zähler [23] = 25

Benutzerfaktor Nenner [26] = 9

TESTTIM - 24

Inhalt	Messzeit im Zielfenster
Beschreibung	Nach dem Erreichen des Zielfensters wird zweimal die Istposition gemessen und mit dem Parameter TESTVAL (25) verglichen. Ist das Ergebnis kleiner als TESTVAL, gilt die Position als erreicht, andernfalls wird erneut gemessen. TESTTIM gibt den Zeitabstand zwischen diesen beiden Messungen an.
Parametergruppe	AXI vordefinierte Eingänge/Ausgänge
Einheit	ms
Bereich	0 ... 10
!!!	Die Einschränkung auf 10 ms ist dadurch begründet, dass die Funktion 'diffval' wirklich wartet und solange auch keine Endschalter- und Schleppfehler-Überwachung aktiv ist. Deswegen sollte diese Zeit nicht zu lange sein.
Init-Wert	0

TESTVAL - 25

Inhalt	Grenzwert für Messwert im Zielfenster
Beschreibung	Nachdem das Zielfenster erreicht ist, wird mit dem in TESTTIM (24) festgelegten Abstand zweimal die Position ausgelesen und der Abstand mit diesem Zielfenster-Grenzwert TESTVAL verglichen. Das Ergebnis entscheidet, ob die Position als erreicht gilt oder nicht.
!!!	Bei größeren Zeitabständen muss berücksichtigt werden, dass das Erreichen der Zielposition auf jeden Fall um diese Zeit verzögert wird.
Parametergruppe	AXI vordefinierte Eingänge/Ausgänge
Einheit	qc
Bereich	1 ... 10000
Init-Wert	1

POSFACT_N - 26

Inhalt	Nenner Benutzerfaktor bzw. im CAM-Mode Umrechnung der Einheiten qc in BE
Beschreibung	<p>Wegangaben in Fahrbefehlen erfolgen in Benutzereinheiten und werden intern in Quadcounts umgerechnet. So ist es durch eine entsprechende Wahl dieser Normierungsgröße möglich, mit beliebigen technischen Maßangaben (zum Beispiel mm) zu arbeiten.</p> <p>Der Faktor ist ein Bruch, der sich aus Zähler und Nenner zusammensetzt:</p> $\text{POSFACT_Z (23)} / \text{POSFACT_N (26)} = 1 \text{ Benutzereinheit BE}$ <p>Die Normierung bestimmt, wie viele Quadcounts eine Benutzereinheit ergeben: Wenn der Faktor zum Beispiel 50375/1000 beträgt, entspricht 1 BE genau 50,375 qc.</p> <p>Im CAM-Mode wird der Parameter benutzt, um die Einheit für den Slave-Antrieb festzulegen, damit man auch im CAM-Editor mit sinnvollen Einheiten arbeiten kann. Siehe Voraussetzung der Formel und Beispiel bei POSFACT_Z (23)</p> $\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ BE}$ <p>Außerdem kann man die Kurven mit diesem Faktor stauchen oder strecken, ohne jeweils neue Kurven definieren zu müssen. Die Verwendung von Zähler und Nenner für den Getriebefaktor führt zu einem sehr präzisen Ergebnis, da in fast allen Fällen Übersetzungen als Bruch darstellbar sind.</p>
Portabilität	Der Faktor ist ab Version 6.4.43 nicht mehr begrenzt auf kleine Werte, siehe auch Skalierung in Kapitel „Programmieren mit APOSS“.
Parametergruppe	AXE Gebersystem
Einheit	–
Bereich	1 ... MLONG
Init-Wert	1000
Beispiel	Siehe POSFACT_Z (23)

ENCODERTYPE - 27

Inhalt	Definiert die Signalquelle des Istwert-Gebers für die Achsregelung.
Beschreibung	<p>ENCODERTYPE legt die Signalquelle des Feedbackgebers (typ. Drehgeber) für die Achsregelung fest. Die Istwert-Information kann über die folgenden „Kanäle“ bezogen und entsprechend definiert werden:</p> <ul style="list-style-type: none">- Lokaler Encoder-Eingang der Steuerung- CANopen-Drehgeber via Prozessdatenwort (PDO)- CANopen-Leistungsendstufe via Prozessdatenwort (PDO)- Encodertyp via Prozess Datenwort (PDO) auf dem EtherCAT-Bus <p>Als Drehgeber können Inkremental- und Absolutgeber eingesetzt werden.</p>
Besonderheiten bei Absolutgebern	<p>Bei einigen Typen von linearen Absolutgebern können sporadisch falsche Positionswerte gemeldet werden (typischerweise Sprünge im Positionswert). Mit den Drehgebertypen 3 und 4 können solche Fehler erkannt werden, sofern der Sprung größer als die halbe Drehgeberauflösung ist. Die Korrektur erfolgt mittels eines künstlichen Positionswertes, der sich aus der letzten Geschwindigkeit errechnet. Sollte die Störung länger als 100 Auslesungen ($\gt 100$ ms) anliegen, wird nicht mehr korrigiert, was dann tatsächlich zu einem Schleppfehler führt. Die gesamte Anzahl aufgetretener Störungen wird in einer internen Variablen gespeichert, die über SYSVAR[16] ausgelesen werden kann.</p>
!!!	<p>Mit Absolutgebern können folgende Befehle nicht benutzt werden: DEFORIGIN, HOME, INDEX, IPOS und WAITNDX.</p>
Device-Typen	<p>Bei Drehgeber-Informationen, die über den Bus bezogen werden, wird der Parameterwert anhand der, im Parameter DRIVETYPE definierten Device-Typen nach folgender Formel berechnet:</p> $\text{SET ENCODERTYPE } nn$ <p>wobei $nn = \text{guard} * (\text{busoffset} + \text{Device-Typ} * 1000 + \text{id})$</p> <p>Die Drehgebertypen mit Device-Typ 1 bis 49 werden genauso behandelt wie Inkrementalgeber, bei denen die relative Änderung des über den Bus übertragenen Positionswerts (PDO) zur letzten Position addiert wird.</p> <p>Die Drehgebertypen mit Device-Typ 50 werden behandelt wie Absolutgeber, bei denen ein Überlauf durch den Drehgeber auftritt (vgl. CANopen Drehgeber-Definition).</p> <p>Andere Encoder-Typen bekommen ihre aktuelle Position und Index-Position von einem PDO auf dem EtherCAT-Bus. Beispiel:</p> <pre>SET ENCODERTYPE 251000 // PDO[0] als EtherCAT Encoder benutzen // 200000 = EtherCAT Bus // 51(000) = EtherCAT Encoder // 100er = PDO nr (z.B. 200 = pdo 2) // 1er = Offset in PDO in Longs (0 = 1. Long in PDO benutzen)</pre> <p>Der <u>erste</u> Long auf diesem Platz wird für die <u>Position</u> benutzt, der <u>zweite</u> für die <u>Index-Position</u> z.B. SET ENCODERTYPE 251105</p> <pre>// PDO1 Long mit Offset 5 (PCD[6]) für Position benutzen, // und Long mit Offset 6 (PCD[7]) für Index-Position</pre> <p>Die Implementierung ist ähnlich der CAN Encoder (in can_opn.c).</p>
ENCODERTYPE 9	<p>In Übereinstimmung mit ENCODERTYPE 9 (Slave Simulation) hat sich die Bedeutung von SYNCMTYPS, der den Typ des Markereingangs beschreibt, geändert. In älterer Firmware gab es keine Markersimulation oder nur mit SYNCMTYPS 9. Das gilt so nicht mehr.</p> <p>Die Definition ist nun wie folgt.</p>

SYNCMTYPS unterstützt die folgenden Werte (if ENCODERTYPE == 9)

- 0– virtuelle Marker werden mit einem Abstand von ENCODERFREQ erzeugt.
- n2– Bei positiver Flanke des Eingangs n wird die aktuelle Position als Markerposition genommen.
- n3– Bei negativer Flanke des Eingangs n wird die aktuelle Position als Markerposition genommen.

Die Genauigkeit ist natürlich begrenzt auf 1 ms, die interne Update-Rate.

Behandlung der automatisch erzeugten CAN-Objekte	Für die Informationsübertragung der Istposition über den Bus werden die benötigten CAN-Objekte (PDOs, GUARD-, SYNC-Objekt) automatisch angelegt und der Teilnehmer initialisiert (NMT0). Diese im Hintergrund ausgeführte Objekt-Erzeugung entspricht im Wesentlichen dem CANINI Befehl. Im Gegensatz zu der applikationsseitigen Objekt-Erzeugung mit dem CANINI Befehl, bleiben die mit ENCODERTYPE automatisch erzeugten Objekte auch bei einem erneuten applikationsseitigen CANINI unverändert erhalten. Ebenfalls werden die mit ENCODERTYPE automatisch erzeugten Objekte auch durch ein CANDEL -1 nicht gestoppt. Die automatisch erzeugten Objekte können innerhalb der Applikation nur durch einen erneuten, abweichenden SET ENCODERTYPE Befehl gelöscht bzw. umkonfiguriert werden. Auch beim Programmabbruch (mittels Esc) bleiben die Objekte noch bis zum nächsten Programmstart erhalten. Erst beim Neustart eines Programms werden die Objekte gelöscht und entsprechend der Einstellung im permanenten Parametersatz oder entsprechend der Definition im Programm neu angelegt.
Guarding	Bei konfiguriertem Guarding (d.h. ENCODERTYPE > 0) wird der Bus-Teilnehmer überwacht und eine fehlende Rückmeldung auf das GUARD-Objekt (z.B. aufgrund des Ausfalls des Teilnehmers) führt zu dem Fehler 88. Es liegt somit in der Verantwortung des Anwenders darauf zu achten, dass zum Zeitpunkt der Definition des Parameters ENCODERTYPE der Parameter REVERS (entsprechend der Bus-ID) korrekt gesetzt ist, der Bus-Teilnehmer korrekt konfiguriert (Baudrate, Bus-ID), angeschlossen, eingeschaltet und für die Kommunikation bereit ist. !!! Falls bei einem ENCODERTYPE > 0 der Bus-Teilnehmer ausfällt, muss von der Applikation (in der Fehlerbehandlungsroutine) durch das Setzen des negativen ENCODERTYPE-Werts die Überwachung abgeschaltet werden, bis der Fehler auf Seiten des Teilnehmers behoben ist. Ansonsten tritt nach jedem Löschen des Steuerungsfehlerzustands 88 (mit ERRCLR) sofort wieder die identische Fehlermeldung auf.
PDO-Mapping für CANopen-Antriebe	Werden CANopen-Leistungsendstufen für die Drehgeber-Rückmeldung verwendet, muss das so genannte PDO-Mapping, d.h. welche Objekte der Leistungsendstufe in die übertragenen PDOs verlinkt sind, in der Applikation vor dem Setzen der Parameter DRIVETYPE und ENCODERTYPE ausgeführt werden. Die Wahl der am besten geeigneten Objekte hängt von der CANopen-Leistungsendstufe ab. Die zub machine control AG gibt gerne Beispiel-Sourcecode passend zu den unten genannten Device-Typen ab. Nennen Sie uns hierzu bitte die eingesetzte Leistungsendstufe per e-Mail (info@zub.ch).
SYNC Periodizität	Parameterwert von TIMER definiert war, der bei einer Achse eingestellt wurde. Bei aktuellen Firmware-Versionen (6.5.16 oder höher) erfolgt die Einstellung der SYNC-Rate über den globalen Parameter CANSYNCTIMER. Dieser Parameter muss auf einen Wert gesetzt werden, der auf den langsamsten CAN-Bus Teilnehmer abgestimmt ist.
Positionswerte-Erfassung anhand von Marker-Signalen	Bei Markersynchronisationen oder Positionierungen, die von der Echtzeiterfassung von Positionswerten anhand von Marker-Signalen (siehe Befehl IPOS, Parameter SYNCMTYPS) abhängig sind, wird empfohlen die entsprechenden Encoder-Signale auf die lokalen Encoder-Eingänge der Steuerung zu legen. Nur in diesem Fall ist das (Hardware)-Latching der Positionswerte mit der maximalen Genauigkeit von 1 qc möglich.

!!! Bei Encoder-Informationen, die über den Bus übertragen werden, verringert sich die Genauigkeit der Markererfassung in Abhängigkeit von der Zykluszeit des Bus (Parameter TIMER, CANSYNCTIMER) und der Drehzahl des Antriebs.

Portabilität	Nur für 32-Bit Steuerungen, wie M1 und MACS; Master-Simulation via CAN-Bus ab Controller Version 6.1.14; Encoder-Information via PDO an EtherCAT ab Version 6.5.01; absoluter passiver Encoder (SSI) ab Version 6.7.05 (MACS4); geänderte Bedeutung von SYNCMTYPS betreffend Slave-Simulation ab Version 6.7.19.
Parametergruppe	Achsparameter Encoder AXE
Querverweise	MENCODERTYPE, DRIVETYPE, REVERS, IPOS, CANINI, TIMER, CANSYNCTIMER
Einheit	–
Bereich	–MLONG ... MLONG
	0 = Inkrementalgeber
	1 = Absolutgeber, Standard ca. 262 kHz
	2 = Absolutgeber, ca. 105 kHz
	3 = Absolutgeber ohne Überlauf (linear), jedoch mit Fehlerkorrektur, ca. 262 kHz
	4 = Absolutgeber ohne Überlauf (linear), jedoch mit Fehlerkorrektur ca. 105 kHz
	5 = Sinus / Cosinus Encoder
	9 = Simulierter Drehgeber, d.h. Sollposition wird automatisch als Istposition übernommen (z.B. für Programmtest ohne angeschlossenen Motor)
	10 = Inkrementalgeber am lokalen Encoder-Anschluss 1
	20 = Inkrementalgeber am lokalen Encoder-Anschluss 2
	30 = Inkrementalgeber am lokalen Encoder-Anschluss 3
	> 1000 = nn = Drehgeber-Information via Bus übertragen
	nn = guard * (busoffset + Device-Typ * 1000 + id)
	guard = -1, +1 (ohne / mit Guarding)
	busoffset = 100000, 0 (Slave-Bus, Master-Bus)
	Device-Typ (siehe auch DRIVETYPE)
	0 = Standard Antrieb (analoger Ausgang)
	1 = CAN-Antrieb des Typs Lenze (Standard)
	2 = CANopen-Servoverstärker nach DS402
	3 = CANopen-Servoverstärker maxon EPOS
	4 = CANopen-Servoverstärker zub DSA
	5 = CAN-Antrieb des Typ Lenze (Sollwert-Limit)
	10 = Danfoss Drives
	50 = CANopen-Drehgeber
	91 .. 94 = Analoge Ausgänge zu einem CANopen I/O Modul
	id = 1...127 = Bus-ID, mit Device-Typ 1 – 50 und Datenaustausch über PDO 2
	id = 257...511 = Bus ID 1...127 mit Device-Typ 1, aber Datenaustausch über PDO 3
	> 100000 = nn = Encoder Information via EtherCAT erhalten
	nn = busoffset + device type * 1000 + PDO no. + long offset
	device type =
	51 = Encoder Information via PDO an EtherCAT

Beispiel:

```
SET ENCODERTYPE 251100 // pdo[1] als EtherCAT Encoder benutzen
// 200000 = EtherCAT Bus
// 51(000) = EtherCAT Encoder
// 100er = PDO nr (z.B. 200 = pdo 2)
// 1er = Offset in PDO in Longs (0 = 1. Long im PDO benutzen)
```

Der erste Long auf diesem Platz wird für die Position benutzt, der zweite für die Index-Position.

z.B. SET ENCODERTYPE 251105

```
// PDO1 Long mit Offset 5 (PDO[6]) für Position benutzen,
// und Long mit Offset 6 (PDO[7]) für die Index-Position
```

Init-Wert 0

POSDRCT - 28

Inhalt	Drehrichtung
Beschreibung	<p>Normalerweise ist es für die Lageregelung erforderlich, dass ein positiver Sollwert auch eine positive Änderung der Position bewirkt. Falls dies nicht der Fall ist (Drehgeber vertauscht oder Motor falsch angeschlossen o. ä.), kann die Drehrichtung durch die Parameterwerte $-2 \dots 2$ intern umgedreht werden.</p> <p>Bei manchen Anwendungen ist es dagegen wünschenswert, die Drehrichtung in Benutzereinheiten umzudrehen. Dies wird durch das negative Vorzeichen bei POSDRCT erreicht.</p> <p>Die Richtung der Synchronisation (Verhältnis zum Master) kann durch negativen SYNCFAC TM umgedreht werden.</p> <p>Bei einer Synchronisation im CAM-Mode können Sie mit POSDRCT für den Slave eine positive Drehrichtung bestimmen, die Voraussetzung für die CAM-Funktionalität ist.</p> <p>Grundsätzlich wird intern immer die Drehgeberinformation verwendet, die vom FPGA eingeht.</p> <p>Ab Compiler 6.5.36 kann die Rückführung des Geschwindigkeitsreglers mit den Parametern AMPENCNO, AMPENCRES und AMPENCRPM getrennt von der Rückführung des Positionsreglers festgelegt werden. Hierfür kann mit dem Parameter POSDRCT auch die Auswertung (d.h. normal oder invertiert) für den Feedback-Encoder des Geschwindigkeits- und Positionsreglers getrennt festgelegt werden:</p> <p>Die Zehnerstellen konfigurieren die Art der Auswertung (normal oder invertiert) für den Geschwindigkeitsregler.</p> <p>Werden einstellige Konfigurationswerte früherer Systeme benutzt, werden beide Signalquellen auf die identische Art ausgewertet.</p> <p>Alle Konfigurationswerte können mit negativem Vorzeichen kombiniert werden, wodurch das Vorzeichen der Benutzereinheiten invertiert wird.</p>
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS; ab Compiler 6.5.36: Die Art der Auswertung (normal oder invertiert) kann für die Geschwindigkeits- und Positionsrückführung getrennt festgelegt werden.
Parametergruppe	Achsparameter Encoder AXE
Einheit	–
Bereich	$-22 \dots 22$ 1 = Keine Veränderung, d.h. positive Sollwerte ergeben positive Drehgeberwerte: Drehzahl- und Positionsrückführung sowie Benutzereinheiten bleiben unverändert. -1 = Benutzereinheiten (BE) umgedreht Das Vorzeichen der Benutzereinheit wird gedreht. Positive Sollwerte ergeben demnach positive Drehgeberwerte, die aber negativ angezeigt werden. Dies gilt für alle Ausgaben (APOS, CPOS, ...), alle Benutzereingaben (POSA, POSR, ...) und alle Synchronisationsfaktoren sowie Geschwindigkeiten (CVEL, HOME_VEL). Drehzahl- und Positionsrückführung unverändert, Benutzereinheiten werden mit invertiertem Vorzeichen versehen. 2 = Sollwert umgedreht Das Vorzeichen des Sollwertes wird intern getauscht (Plus wird Minus und umgekehrt). Dies kommt einem Umdrehen der Motorleitungen gleich, bzw. dem Vertauschen der A- und B-Spur beim Drehgeber. Drehzahl- und Positionsrückführung verwenden ein invertiertes Vorzeichen, Benutzereinheit unverändert

- 2 = BE und Sollwert umgedreht
Wie „2“, also das Vorzeichen des Sollwertes wird intern getauscht und zusätzlich wird wie bei „-1“ das Vorzeichen der Benutzereinheit (BE) negiert.
Drehzahl- und Positionsrückführung verwenden ein invertiertes Vorzeichen,
Benutzereinheiten werden mit invertiertem Vorzeichen versehen.
- 12 = Positionsrückführung verwendet ein invertiertes Vorzeichen,
Drehzahlrückführung unverändert,
Benutzereinheit unverändert
- 21 = Positionsrückführung unverändert,
Drehzahlrückführung verwendet ein invertiertes Vorzeichen,
Benutzereinheit unverändert
- 22 = wie POSDRCT 2, d.h.
Drehzahl- und Positionsrückführung verwenden ein invertiertes Vorzeichen,
Benutzereinheit unverändert.

Jede der zweistelligen PODRCT-Konfigurationen kann ebenfalls mit einem negativen Vorzeichen vorgegeben werden um das Vorzeichen der Benutzereinheiten zu invertieren.

Init-Wert 1

PROFTIME - 29

Inhalt	Abtastzeit für Profilgenerator
Beschreibung	<p>Der Parameter bestimmt die Abtastzeit für den Profilgenerator, der unabhängig von der Abtastzeit der PID-Regelung (TIMER) ist.</p> <p>Bei anspruchsvollen Regelaufgaben im Hintergrund (SYNCP, SYNCM, SYNCC) kann die Ausführungszeit des APOSS-Programms drastisch ansteigen. In solchen Fällen kann die Abtastzeit des Profilgenerators auf 2 erhöht werden, um mehr Zeit für das APOSS-Programm zur Verfügung zu haben. Höhere Werte als 2 ms sind aber kaum von Vorteil.</p> <p>!!! Nach einem SET PROFTIME Befehl müssen die Befehle VEL, ACC und DEC gesetzt werden.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXR Pos. Regelung
Einheit	ms
Bereich	1 ... 5
	1 = 1 ms
	2 = 2 ms
	3 = 3 ms
	4 = 4 ms
	5 = 5 ms
Init-Wert	1

MENCODER - 30

Inhalt	Auflösung des Master-Drehgebers
Beschreibung	<p>MENCODER gibt die Strichzahl des Master-Drehgebers in Strichen pro Umdrehung an. Diese Angabe ist für alle Synchronisationsbefehle wichtig.</p> <p>Durch Setzen von MENCODER auf einen negativen Wert, kann man auswählen, dass auch die Master-Inkremete durch SYNCFACTS und SYNCFACTM skaliert werden. Das heißt</p> $1 \text{ Master Unit [MU]} = q_c * \text{SYNCFACTS} / \text{SYNCFACTM}.$ <p>Dies betrifft die Befehle MAPOS, MIPOS, MAVEL, ON MAPOS, SETMORIGIN</p>
Portabilität	<p>Nur bei 32-Bit Steuerungen wie M1 und MACS.</p> <p>Negative Werte werden ab Version 6.4.53 unterstützt.</p>
Parametergruppe	AXE Gebersystem
Einheit	Geberstriche/Umdrehung
Bereich	- MLONG ... MLONG
Init-Wert	500

RAMPMIN - 31

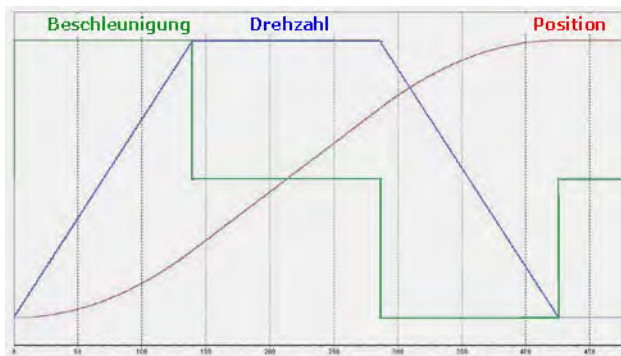
Inhalt	Kürzeste Rampe
Beschreibung	Der Parameter RAMPMIN legt die kürzeste Rampe (maximale Beschleunigung) fest. Er gibt an wie lange die Beschleunigungsphase mindestens dauert, um die Nenngeschwindigkeit zu erreichen.
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXB Grundeinstellungen
Einheit	ms
Bereich	1 ... 360000
Init-Wert	1000

RAMPTYPE - 32

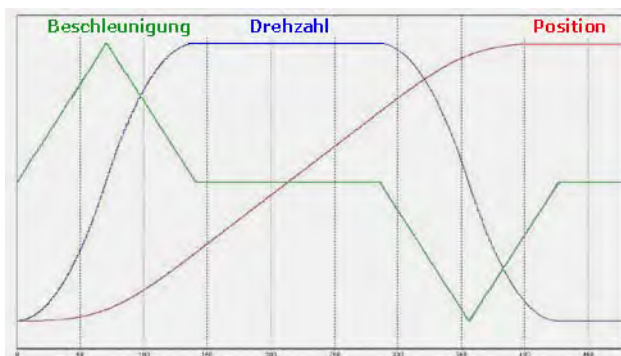
Inhalt Rampenform

Beschreibung RAMPTYPE bestimmt die Rampenform: Trapez, sinusförmig oder mit Ruckbegrenzung. Diese Rampentypen sind für alle Fahrbewegungen (POSA, POSR, CVEL und MOTOR STOP) relevant, aber nicht bei SYNCx-Befehlen.

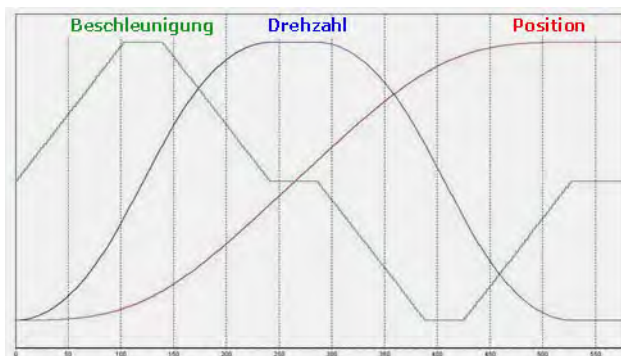
Rampentyp 0
Trapez



Rampentyp 1
S-Rampe



Rampentyp 2
mit Ruckbegrenzung



Bewegungen mit Ruckbegrenzung beginnen mit der Beschleunigung 0 und erhöhen die Beschleunigung durch einen maximalen Ruck so lange, bis die maximale Beschleunigung, die in RAMPMIN festgelegt ist, erreicht ist. Dann wird die Bewegung mit der maximalen Beschleunigung fortgeführt. Am Ende wird die Beschleunigung durch den maximalen Ruck verringert bis die Beschleunigung erneut 0 beträgt. Der maximale Ruck wird durch den Parameter JERKMIN berechnet.

Es stehen vier unterschiedliche JERKMIN Parameter für den Rampentyp mit Ruckbegrenzung zur Verfügung, siehe auch „Ruckbegrenzung“ im Kapitel „Motion Control Praxis“.

Portabilität Nur bei 32-Bit Steuerungen wie M1 und MACS;
Auswahl '2' nur mit MACS3 und Firmware Version ab 6.4.63 sowie APOSS Entwicklungs-
oberfläche Version 6.7.40 mit der integrierten Compiler-Version 6.3.20.

Parameter-Referenz ♦ Achsparameter im Detail

Parametergruppe	AXB Grundeinstellungen
Bereich	0 ... 2 0 = linear, Trapez 1 = S-Rampe, sinusförmig 2 = Bewegungsprofil mit Ruckbegrenzung
Init-Wert	0
Programmbeispiel	Datei → Beispiel: JerkMinTest.m

DFLTVEL - 33

Inhalt	Default-Geschwindigkeit
Beschreibung	DFLTVEL gibt die Default-Geschwindigkeit an, die immer dann verwendet wird, wenn keine Geschwindigkeit im Verfahrssatz definiert wurde. Der Wert bezieht sich auf den <i>Geschwindigkeitsteiler</i> VELRES (22).
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXB Grundeinstellungen
Einheit	1/VELRES Standard = 1/100 = %
Bereich	1 ... VELRES
Init-Wert	50

DFLTACC - 34

Inhalt	Default-Beschleunigung
Beschreibung	DFLTACC gibt die Beschleunigung an, die verwendet wird, wenn keine explizite Angabe vorliegt. Die Angabe erfolgt im Verhältnis zu RAMPMIN bezogen auf den Parameter VELRES (22), also entsprechend der Werkseinstellung in Prozent der maximalen Beschleunigung.
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXB Grundeinstellungen
Einheit	1/VELRES Standard = 1/100 = %
Bereich	1 ... VELRES
Init-Wert	50

BANDWIDTH - 35

Inhalt	Bandbreite innerhalb der der PID-Filter aktiv ist
Beschreibung	<p>Die Bandbreite, in der der PID-Regel-Algorithmus wirken soll, kann begrenzt werden, um zum Beispiel bei schwingungsgefährdeten Systemen das Aufschaukeln der Schwingungen zu vermeiden.</p> <p>Wenn Sie mit Bandbreitenbegrenzung (BANDWIDTH < 1000) arbeiten, müssen Sie für die Parameter FFVEL (36) und FFACC (37) ebenfalls geeignete Werte eintragen, um eine entsprechende Regelung zu erreichen. Ein so eingestelltes System ist zwar nicht mehr so dynamisch, dafür aber wesentlich stabiler und neigt weniger zu unkontrollierten Schwingungen.</p> <p><u>Vorteile der Bandbreitenbegrenzung:</u></p> <p>Die Bandbreitenbegrenzung reduziert die Schwingungsneigung</p> <p><u>Nachteile der Bandbreitenbegrenzung:</u></p> <p>Die Bandbreitenbegrenzung limitiert das Stillstandsmoment der Regelung bzw. des Antriebs. Dies kann bei Vertikalanwendungen, bei welchen eine Last mit einem permanenten Moment gehalten werden muss, ungeeignet sein.</p> <p>!!! Wichtig: Bei Verwendung der Bandbreitenbegrenzung (d.h. BANDWIDTH < 1000) muss die Feedforward-Steuerung aktiviert werden, d.h. der Parameter FFVEL muss typisch auf einen Wert von 1000 eingestellt sein.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXR Pos. Regelung
Einheit	1/10 %
	Der Wert 1000 bedeutet, dass der PID-Filter den vollen Sollwert ausgeben kann. Bei einem BANDWIDTH von 500 werden nur 50 % des Sollwerts ausgegeben. Kleinere Werte als 1000 begrenzen also den Proportionalanteil entsprechend.
Bereich	0 ... 1000
Init-Wert	1000

FFVEL - 36

Inhalt	Geschwindigkeits-Feedforward
Beschreibung	<p>Wenn eine Regelung in der Bandbreite begrenzt ist, muss eine Grundgeschwindigkeit vorgegeben werden, damit ausgeschlossen wird, dass die Regelung durch die eingestellte Begrenzung das Fahren des Antriebs gänzlich verhindert. FFVEL gibt an, mit welcher Geschwindigkeit der Vorwärtsschub ausgeführt wird.</p> <p>Um mit einer reinen PID-Regelung zu arbeiten, muss FFVEL auf 0 eingestellt sein, was auch dem Defaultwert entspricht. In diesem Fall ist die Feedforward-Steuerung inaktiv, d.h. der generierte Sollwert ist lediglich durch die PID-Regelung bestimmt.</p> <p>Die Feedforward-Steuerung eignet sich insbesondere für ...</p> <ul style="list-style-type: none">- größere Antriebe (wie Asynchronmotoren + Frequenzumrichter)- träge Antriebe- Systeme bei denen aufgrund von Schwingungsneigung eine Bandbreitenbegrenzung verwendet wird. <p>Die Feedforward-Steuerung muss verwendet werden, falls eine PID-Bandbreitenbegrenzung (d.h. Parameter BANDWIDTH < 1000) aktiv ist.</p> <p>Mit der Feedforward-Steuerung wird ein Sollwert in Abhängigkeit von der aktuellen Soll-Geschwindigkeit ausgegeben. Der Sollwertanteil der PID-Regelung wird zu dem Sollwert der Feedforward-Steuerung addiert. FFVEL ist auf 1000 skaliert. Dies bedeutet, dass für die definierte Maximaldrehzahl (Parameter VELMAX) ein Sollwert von 10 V generiert wird. Falls der untergeordnete Servoverstärker oder Frequenzumrichter ebenfalls auf die identische Maximaldrehzahl bei 10 V konfiguriert ist, kann für FFVEL typischerweise ein Wert von 1000 gesetzt werden. Andernfalls gilt folgender Zusammenhang:</p> $\text{FFVEL} = 1000 * \text{VELMAX} / (10\text{V-Drehzahl der Leistungsendstufe})$
Portabilität	<p>Nur bei 32-Bit Steuerungen wie M1 und MACS.</p> <p>Mit Controller Version 6.1.14 aufwärts funktioniert der Parameter auch dann, wenn der TIMER ungleich 1 ist.</p>
Parametergruppe	AXR Pos. Regelung
Einheit	-
Bereich	0 ... 100000
Init-Wert	0
Beispiel	<ul style="list-style-type: none">- Der Parameter VELMAX ist auf 1400 Umdr./Min eingestellt.- Beim Servoverstärker oder dem Frequenzumrichter ist für 10V-Sollwert eine Drehzahl von 1600 U/Min konfiguriert. <p>=> $\text{FFVEL} = 1000 * 1400 / 1600 = 875$</p>

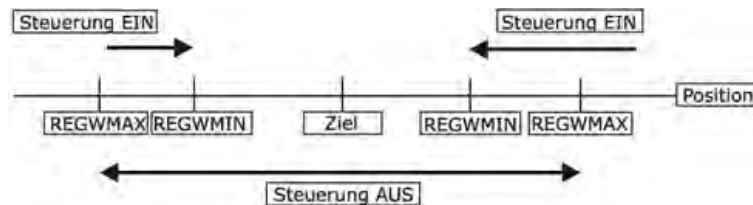
FFACC - 37

Inhalt	Beschleunigungs-Feedforward
Beschreibung	<p>Geben Sie eine Grundbeschleunigung vor, wenn Sie die Regelung in der Bandbreite begrenzt haben. Damit verhindern Sie, dass die Regelung durch die eingestellte Begrenzung überhaupt nicht beschleunigt. FFACC gibt an mit welcher Beschleunigung der Vorwärtsschub ausgeführt wird.</p> <p>Bei einem normalen PID-Algorithmus ist dieser Wert 0.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXR Pos. Regelung
Einheit	–
Bereich	0 ... 100000
Init-Wert	0

REGWMAX - 38

Inhalt Größe des Regelfensters (Aktivierung)

Beschreibung Die Parameter REGWMAX und REGWMIN (39) werden benutzt, um die Lageregelung innerhalb von definierten Bereichen (Regelfenster) an- und abschalten zu können: REGWMAX gibt dabei die Größe des Fensters an, außerhalb dessen die Regelung wieder beginnen soll.



Portabilität Nur bei 32-Bit Steuerungen wie M1 und MACS.

Parametergruppe AXR Pos. Regelung

Einheit qc

Bereich 0 ... MLONG

Init-Wert 0

REGWMIN - 39

Inhalt Größe des Regelfensters (Deaktivierung)

Beschreibung REGWMIN gibt die Größe des Fensters an, innerhalb dessen die Regelung deaktiviert werden soll, bis wieder das Regelfenster REGWMAX (38) erreicht wird.

Portabilität Nur bei 32-Bit Steuerungen wie M1 und MACS.

Parametergruppe AXR Pos. Regelung

Einheit qc

Bereich 0...MLONG

Init-Wert 0

HOME_TYPE - 40

Inhalt	Homefahrt Verhalten
Beschreibung	Es wird festgelegt, wie die Homefahrt erfolgen soll.
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXH Referenzierung
Einheit	–
Bereich	0 ... 3 0 = Reverse und Index Mit Homefahrt-Geschwindigkeit und -Richtung bis zum Referenzschalter fahren, dann Reversieren und langsam den Schalter verlassen; anschließend zum nächsten Indeximpuls fahren. 1 = Reverse, kein Index Wie 0, aber ohne Suchen des Indeximpulses. 2 = Vorwärts und Index Wie 0, aber ohne Reversieren, sondern in gleicher Richtung weiter aus dem Schalter herausfahren. 3 = Vorwärts, kein Index Wie 1, aber ohne Reversieren.
Init-Wert	0

HOME_RAMP - 41

Inhalt	Rampe für Homefahrt normiert auf Einheit VELRES (22).
Beschreibung	<p>Beschleunigung, die für die Homefahrt verwendet wird. Die Angabe bezieht sich auf die minimale Rampe, die in Parameter RAMPMIN (31) definiert ist. Die Einheit ergibt sich durch den Parameter VELRES (22), standardgemäß in % von der kürzesten Rampe; 50 % bedeutet dann halb so schnell, also doppelt so lange.</p> <p>Für HOME_RAMP ergibt sich folgender Zusammenhang:</p> $\text{HOME_RAMP [ms]} = \frac{\text{VELRES (22)}}{\text{HOME_RAMP (41)}} * \text{RAMPMIN (31) [ms]}$ <p>!!! HOME_RAMP kann nie einen höheren Wert haben als die Default-Beschleunigung DFLTACC (34).</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXH Referenzierung
Einheit	VELRES
Bereich	1 ... VELRES
Init-Wert	10

HOME_OFFSET - 42

Inhalt	Nullpunkt-Offset bezüglich Maschinennullpunkt (MN) bzw. Index
Beschreibung	HOME_OFFSET wird benutzt, um einen Offset (Versatz) einzuführen, vergleichbar mit dem Referenzschalter oder Indexpuls. Nach der Homefahrt wird der Antrieb auf HOME_OFFSET positioniert. An dieser Stelle wird auch der Maschinennullpunkt oder Index gesetzt.
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXH Referenzierung
Einheit	qc
Bereich	-MLONG ... MLONG
Init-Wert	0

ERRCOND - 43

Inhalt	Verhalten im Fehlerfall
Beschreibung	<p>Wenn Hardware- oder Software-Endschalter aktiviert sind, kann ein Software-Endschalterfehler gelöscht und danach in die entgegengesetzte Richtung gefahren werden. Wenn danach erneut versucht wird, in die falsche Richtung zu fahren, wird ein neuer Fehler erzeugt.</p> <p>Hardware-Endschalter werden genauso behandelt wie Software-Endschalter. Das heißt, dass der Fehler gelöscht und in die entgegengesetzte Richtung gefahren werden kann. Aber wenn versucht wird, danach erneut in die falsche Richtung zu fahren, tritt der Fehler 98 F_LIMIT_VIOLATION auf.</p>
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS; Parameterwert 4 ab Controller Version 6.1.14, Parameterwert 5 ab Controller Version 6.1.3, Verhalten bei aktivierten Endschaltern ab Firmware 6.6.60
Parametergruppe	AXI vordefinierte Ein-/Ausgänge
Einheit	–
Bereich	0 ... 5
	0 = Freilauf: Standard, d.h. Antrieb geht in Freilauf, Regelschleife wird unterbrochen.
	1 = Freilauf und Bremse: Wie 0, aber Ausgang Bremse (falls definiert) wird aktiviert: siehe O_BRAKE (48).
	2 = Geregelter Stopp: Motorstopp mit max. Verzögerung (Stoppampe), anschließend stillstandgeregelt.
	3 = Geregelter Stopp mit Bremse: Wie 2, zusätzlich wird der Ausgang Bremse aktiviert (falls definiert), aber erst nach MOTOR STOP. Alle anderen Aktivitäten, wie MOTOR OFF und ähnliche müssen in der ON_ERROR Routine gesetzt werden.
	4 = Wird nur für CAN-Antriebe benutzt: Im Fehlerfall wird eine spezielle Leerlauf Funktion benutzt, die den Sollwert nicht auf Null setzt oder den Frequenzrichter (FU) ausschaltet, sondern die Bits setzt, um mit FU-Rampe zu stoppen. Alle weiteren Aktivitäten, wie MOTOR OFF und ähnliche müssen in der ON_ERROR Routine gesetzt werden.
	5 = Es wird in die Error-Routine gesprungen, die Regelung aber nicht automatisch abgeschaltet. Dies kann bzw. muss bei Bedarf durch das Anwendungsprogramm über ein MOTOR OFF in der Error-Routine ausgelöst werden. So kann verhindert werden, dass beim Datenaustausch mit CAN-Terminals ein gestörter CAN-Verkehr bereits zu einem Stoppen des Reglers führt. (Z.B. wenn nur reine Informationsdaten an ein Terminal übertragen werden, deren korrekte Anzeige nicht sicherheitsrelevant ist oder die ohnehin zyklisch immer wieder aktualisiert werden.) In der Error-Routine kann dann zuerst überprüft werden, ob es sich um einen CAN-Fehler (Err. 89) handelt, der sofort wieder gelöscht werden kann. In allen anderen Fällen kann in der Error-Routine die Motorregelung bei Bedarf abgeschaltet und der laufende Prozess entsprechend den Notwendigkeiten der Applikation gestoppt werden.
	!!! In Parameter O_BRAKE (48) muss ein Ausgang Bremse definiert sein.
Init-Wert	0

ENDSWMOD - 44

Inhalt	Verhalten bei Endschalter
Beschreibung	ENDSWMOD gibt an wie sich die Steuerung bei Erreichen des positiven oder negativen Hardware-Endschalters verhalten soll. Fehlerverhalten siehe ERRCOND (43)
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	AXI vordefinierte Ein-/Ausgänge
Einheit	–
Bereich	0 ... 1 0 = Fehler auslösen (Fehler-Unterprogramm aufrufen) 1 = Kontrollierter Stopp (Motorstopp mit max. Verzögerung)
Init-Wert	0

I_REFSWITCH - 45

Inhalt	Referenzschalter Eingang	
Beschreibung	I_REFSWITCH legt fest, welcher Eingang als Referenzschalter dienen soll. Dabei kann man festlegen ob auf eine positive oder negative Flanke reagiert werden soll. Verhalten bei Erreichen siehe HOME_TYPE (40).	
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.	
Parametergruppe	AXI vordefinierte Ein-/Ausgänge	
Einheit	–	
Bereich	1 ... 8	= Reaktion auf positive Flanke an Eingang 1 ... 8
	0	= keine Funktion
	-1 ... -8	= Reaktion auf negative Flanke an Eingang 1 ... 8
Init-Wert	0	

I_POSLIMITSW - 46

Inhalt	Endschalter positiv	
Beschreibung	I_POSLIMITSW legt fest, welcher Eingang als positiver Endschalter benutzt werden soll und ob auf eine positive oder negative Flanke reagiert werden soll. End- und Referenzschalter erlauben den Einsatz jeden beliebigen Eingangs, d.h. es werden nicht nur 1..8 unterstützt, sondern auch höhere Nummern sowie virtuelle Ein- und Ausgänge. !!! Allerdings ist es gefährlich, virtuelle Eingänge als Endschalter zu benutzen, weil sie die direkte Verbindung zwischen Endschalter und der Steuerung entfernen. Das kann zu einer Zeitverzögerung führen (oder zu anderen Fehlerszenarien), die die Fähigkeit der Steuerung begrenzt, schnell auf einen Endschalter zu reagieren.	
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS. Beliebige Eingänge und virtuelle I/Os werden ab Firmware 6.6.60 unterstützt.	
Parametergruppe	AXI vordefinierte Ein-/Ausgänge	
Einheit	–	
Bereich	0	= keine Funktion
	positive Eingangsnummer	= Reaktion auf positive Flanke an entsprechendem Eingang #
	negative Eingangsnummer	= Reaktion auf negative Flanke an entsprechendem Eingang #
Init-Wert	0	

I_NEGLIMITSW - 47

Inhalt	Endschalter negativ	
Beschreibung	I_NEGLIMITSW legt fest, welcher Eingang als negativer Endschalter benutzt werden soll und ob auf eine positive oder negative Flanke reagiert werden soll. End- und Referenzschalter erlauben den Einsatz jeden beliebigen Eingangs, d.h. es werden nicht nur 1..8 unterstützt, sondern auch höhere Nummern sowie virtuelle Ein- und Ausgänge. !!! Allerdings ist es gefährlich, virtuelle Eingänge als Endschalter zu benutzen, weil dies die direkte Verbindung zwischen Endschalter und der Steuerung entfernt. Das kann zu einer Zeitverzögerung führen (oder zu anderen Fehlerszenarien), die die Fähigkeit der Steuerung begrenzt, schnell auf einen Endschalter zu reagieren.	
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS. Beliebige Eingänge und virtuelle I/Os werden ab Firmware 6.6.60 unterstützt.	
Parametergruppe	AXI vordefinierte Ein-/Ausgänge	
Einheit	–	
Bereich	0	= keine Funktion
	positive Eingangsnummer	= Reaktion auf positive Flanke an entsprechendem Eingang #
	negative Eingangsnummer	= Reaktion auf negative Flanke an entsprechendem Eingang #
Init-Wert	0	

O_BRAKE - 48

Inhalt	Ausgang für Bremse
Beschreibung	<p>O_BRAKE gibt den Ausgang an, mit dem die Bremse aktiviert werden kann.</p> <p>Der Ausgang wird nur bei einem Abbruch oder Fehler gesetzt, wenn ERRCOND (43) auf 1 oder 3 gesetzt ist.</p> <p>Wenn ein Ausgang für die Bremse definiert ist, bleibt dieser auch dann aktiviert, wenn das Programm mit ESC abgebrochen wird.</p> <p>Eine positive Zahl bedeutet, dass der Ausgang high (24 Volt) ist, wenn die Funktion aktiv ist. Eine negative Zahl bedeutet, dass der Ausgang low (0 Volt) ist, wenn die Funktion aktiv ist.</p> <p>!!! Der Ausgang für die Bremse muss immer mit einem OUT Befehl im Programm zurückgesetzt werden.</p>
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS.
Parametergruppe	AXI vordefinierte Ein-/Ausgänge
Einheit	–
Bereich	–8 ... 8
Init-Wert	0
Beispiel	<pre>ON ERROR GOSUB err_handle SET O_BRAKE -1 SET ERRCOND 1 /* Hauptprogrammschleife */ ... SUBPROG err_handle WAITI 1 ERRCLR OUT 1 1 RETURN</pre>

SYNCFAC TM - 49

Inhalt Synchronisationsfaktor Master (M:S)
bzw. im CAM-Mode Umrechnung der Einheiten qc in MU

Beschreibung Die Synchronisation wird mit einem Verhältnis von Master:Slave in qc beschrieben; SYNCFAC TM bestimmt den Synchronisationsfaktor für den Master.

SYNCFAC TM (49) und SYNCFAC TS (50) ermöglichen den Ausgleich unterschiedlicher Getriebe-faktoren bzw. die Anpassung der Drehzahl des Slaves im Verhältnis zur festgelegten Drehzahl des Masters.

$$\text{Slave-Geschwindigkeit} = \text{Master-Geschwindigkeit} \cdot \frac{\text{SYNCFAC TS (50)}}{\text{SYNCFAC TM (49)}}$$

In Verbindung mit der Kurvensynchronisation (CAM) werden die Parameter SYNCFAC TM und SYNCFAC TS zur Umrechnung der qc in MU-Einheiten benutzt. Dadurch kann der Anwender im CAM-Editor mit sinnvollen Einheiten arbeiten. Siehe Beispiel 2.

$$\frac{\text{Getriebefaktor} \cdot \text{Drehgeberauflösung} \cdot 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ MU}$$

vorausgesetzt dass:

$$\text{Getriebefaktor} = \frac{\text{Motorumdrehungen}}{\text{Umdrehungen am Abtrieb}}$$

Drehgeber = Inkrementalgeber (bei Absolutgebern entfällt der Multiplikator 4).

Skalierfaktor = Anzahl der Benutzereinheiten BE [qc], die einer Umdrehung am Antrieb entsprechen.

Portabilität Nur bei 32-Bit Steuerungen wie M1 und MACS.
Ab Version 6.4.43 ist der Faktor nicht mehr auf kleine Werte begrenzt, siehe auch [Skalierung](#) in Kapitel „Programmierung mit APOSS“.

Parametergruppe Achsparameter Synchronisierung AXS

Einheit qc

Bereich -MLONG ... MLONG

-MLONG ... -1 = Dreht die Richtung der Synchronisation (Verhältnis zum Master) um.

Init-Wert 1

Beispiel 1 Wenn der Master zweimal so schnell fahren soll wie der Slave, beträgt das Verhältnis 2:1 (SYNCFAC TM = 2; SYNCFAC TS =1).

Beispiel 2 Transportband:

Die Eingabe soll in 1/10 mm Auflösung möglich sein.

Der Antrieb ist mit dem Transportband mit einer Getriebeübersetzung von 25:11 verbunden; das heißt der Motor macht 25, das Zahnriemenrad 11 Umdrehungen.

Getriebefaktor = 25/11

Inkrementalgeber direkt am Master-Antrieb; Drehgeberauflösung = 4096

Das Zahnriemenrad hat 20 Zähne/Umdrehung, 2 Zähne entsprechen 10 mm, daher entspricht 1 Umdrehung = 100 mm Transportbandvorschub.

Skalierfaktor ist demnach 1000

$$\frac{25/11 \cdot 4096 \cdot 4}{1000} \text{ qc} = 1 \text{ MU}$$

$$\frac{25 * 4096 * 4}{1000 * 11} q_c = \frac{2048}{55} q_c = 1 \text{ MU} = \frac{\text{Syncfactor Master SYNCFACTM (49)}}{\text{Syncfactor Slave SYNCFACTS (50)}}$$

Um mit 1/10 Grad Einteilung zu arbeiten, setzen Sie die Parameter wie folgt:

$$\begin{aligned} \text{SYNCFACTM} &= 2048 \\ \text{SYNCFACTS} &= 55 \end{aligned}$$

Beispiel 3 Berechnung des Skalierfaktors bei einem Reibantrieb:

Der Abtrieb sei mit einem Reibrad (Radius 60 mm) versehen; es soll mit einer Auflösung von 1/10 mm gearbeitet werden:

Eine Umdrehung am Abtrieb berechnet sich demnach:

$$\text{Skalierfaktor} = 2 \pi r * 10 = 2 \pi * 60 * 10 = 3969,91$$

$$\text{Skalierfaktor} = 3970$$

Da durch das Aufrunden auf jeden Fall ein Fehler entsteht, muss nach jeder vollen Umdrehung ein Markerabgleich durchgeführt werden.

SYNCFACTS - 50

Inhalt	Synchronisationsfaktor Slave (M:S) bzw. im CAM-Mode Umrechnung der Einheiten qc in MU
Beschreibung	<p>Die Synchronisation wird mit einem Verhältnis von Master:Slave in qc beschrieben; SYNCFACTS bestimmt dabei den Synchronisationsfaktor für den Slave.</p> <p>SYNCFACTM (49) und SYNCFACTS (50) ermöglichen den Ausgleich unterschiedlicher Getriebe-faktoren bzw. die Anpassung der Drehzahl des Slaves im Verhältnis zur festgelegten Drehzahl des Masters.</p> $\text{Slave-Geschwindigkeit} = \text{Master-Geschwindigkeit} \cdot \frac{\text{SYNCFACTS (50)}}{\text{SYNCFACTM (49)}}$ <p>In Verbindung mit der Kurvensynchronisation (CAM) werden die Parameter SYNCFACTM und SYNCFACTS zur Umrechnung der qc in MU-Einheiten benutzt. Dadurch kann der Anwender im CAM-Editor mit sinnvollen Einheiten arbeiten. Siehe Voraussetzung für die Formel und Beispiel bei SYNCFACTM (49).</p>
Portabilität	<p>Nur bei 32-Bit Steuerungen wie M1 und MACS.</p> <p>Ab Version 6.4.43 ist der Faktor nicht mehr auf kleine Werte begrenzt, siehe auch Skalierung in Kapitel „Programmierung mit APOSS“.</p>
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	qc
Bereich	1 ... 2 * MLONG / max. Master-Geschwindigkeit wobei gilt: Einheit max. Master-Geschwindigkeit = qc/PROFTIME (29)
Init-Wert	1
Beispiel	Siehe SYNCFACTM (49)

SYNCTYPE - 51

Inhalt	Art der Synchronisation
Beschreibung	<p>Die Art, wie die Synchronisation durchgeführt wird, kann geändert werden:</p> <p>Im Standardfall (SYNCTYPE = 0), wird die Positionsdifferenz zwischen Master und Slave unter Beachtung der programmierten ACC, DEC und VEL Begrenzungen des Slaves ausgeglichen.</p> <p>Eine Konfiguration des Synchronisationsverhaltens kann über verschiedene Bits definiert werden:</p> <p>Bit 0 – nicht benutzt.</p> <p>Bit 1 (= + 2): Über das Bit 1 (= +2) von SYNCTYPE kann konfiguriert werden, dass bei Kurvenscheiben (SYNCC, SYNCCMM, SYNCCMS) ohne Berücksichtigung der aktuellen Einstellung von ACC, DEC direkt dem Master entsprechend dem CAM-Profil gefolgt wird.</p> <p>Bit 2 (= +4): Über das Bit 2 von SYNCTYPE kann eine negative Masterfahrt zwischengespeichert werden. Wenn die Masterfahrt negativ ist (das Vorzeichen von SYNCFACTM wird ausgewertet), wird diese Bewegung nicht zu SYNCx weitergeleitet, aber zwischengespeichert. Wenn der Master dann wieder wendet und in die positive Richtung fährt, wird die gleiche Anzahl Quadcounts verbraucht, bevor die positive Bewegung fortgesetzt wird. Dies erlaubt es sehr kleine negative Bewegungen des Masters zu unterdrücken ohne Inkremente zu verlieren.</p> <p>Die Anzahl der zwischengespeicherten Inkremente kann mit SYSVAR ausgelesen werden.</p> <p style="text-align: center;">PFG_SYNCMNEGBUFFER (4315) (0x2500 subindex 220)</p> <p>!!! Hinweis: Die früher verfügbare Auswahl „Look ahead“ ist nicht mehr verfügbar. Eine entsprechende Funktionalität kann über die Konfiguration des Parameters SYNCFFVEL erzielt werden.</p>
Portabilität	Standardparameter seit Controller Version 6.1.14; Änderung der Parameterwerte ab Firmware 6.7.65 (abwärtskompatibel); bisheriger Parameterwert „1 = Look Ahead“ siehe SYNCFFVEL
Parametergruppe	Achsparameter Synchronisation AXS (Noch nicht im Parameter-Dialogfenster realisiert.)
Einheit	–
Bereich	0 ... 6
	Bit 0 – nicht benutzt (Die früher verfügbare Auswahl „Look ahead“ Funktionalität erfolgt über SYNCFFVEL).
	Bit 1
	0 = Standard 1 = CAM-Slave folgt der Kurve direkt ohne ACC, DEC, VEL zu berücksichtigen
	Bit 2
	0 = Standard 1 = Negative Masterbewegungen werden zwischengespeichert und später aufgebraucht.
Init-Wert	0

SYNCKMARKM - 52

Inhalt	Markeranzahl Master
Beschreibung	SYNCKMARKM und SYNCKMARKS müssen entsprechend dem Verhältnis der Anzahl der Marker-signale des Master zum Slave gesetzt werden. Ein Verhältnis von 1:1 bedeutet, dass sich jeder Slave-Marker auf jeden Master-Marker abstimmt. Ein Verhältnis von 2:1 bedeutet, dass sich jeder Slave-Marker auf jeden zweiten Master-Marker abstimmt.
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	–
Bereich	0 ... 10000
Init-Wert	1

SYNCKMARKS - 53

Inhalt	Markeranzahl Slave
Beschreibung	SYNCKMARKM und SYNCKMARKS müssen entsprechend dem Verhältnis der Anzahl der Marker-signale des Master zum Slave gesetzt werden. Ein Verhältnis von 1:1 bedeutet, dass sich jeder Slave-Marker auf jeden Master-Marker abstimmt. Ein Verhältnis von 2:1 bedeutet, dass sich jeder Slave-Marker auf jeden zweiten Master-Marker abstimmt.
Portabilität	nur bei 32-Bit Steuerungen, wie M1 und MACS
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	–
Bereich	0 ... 10000
Init-Wert	1
Beispiel	Der Master-Marker ist ein externes Signal, das meldet, wenn ein Transportgut ankommt; der dazugehörige Slave-Marker ist der Indeximpuls vom Motor. Wenn der Motor immer 3 Umdrehungen benötigt bis ein Gut ankommt, dann bedeutet das, dass auch immer 3 Indeximpulse vergehen müssen bis ein Marker kommt. Daraus ergibt sich ein Verhältnis von 3:1; es wird nur jeder dritte Slave-Puls ausgewertet.

SYNCPOSOFFS - 54

Inhalt	Positionsoffset bei Positionssynchronisation
Beschreibung	<p>Setzt den Offset für die Positionssynchronisation (SYNCP). Dieser Offset ist auch bei einer Positionssynchronisation mit Markerkorrektur gültig (SYNCM).</p> <p>Der <i>Positionsoffset</i> kann während der Synchronisation jederzeit per Befehl online verändert werden.</p> <p>!!! Der Offset für die Positionssynchronisation wird sofort ausgeführt, wenn der Befehl SYNCP folgt.</p> <p>Beim Start von SYNCM dagegen wird auf die erste Auswertung der Markerpulse gewartet. Erst dann wird der Offset angewandt.</p> <p>Zur Vermeidung von Kompatibilitätsproblemen sollten Sie mit SYNCMSTART (62) das <i>Startverhalten</i> von SYNCM festlegen.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	qc
Bereich	-MLONG /SYNCFACFS(50) ... MLONG/SYNCFACFS(50)
Init-Wert	0

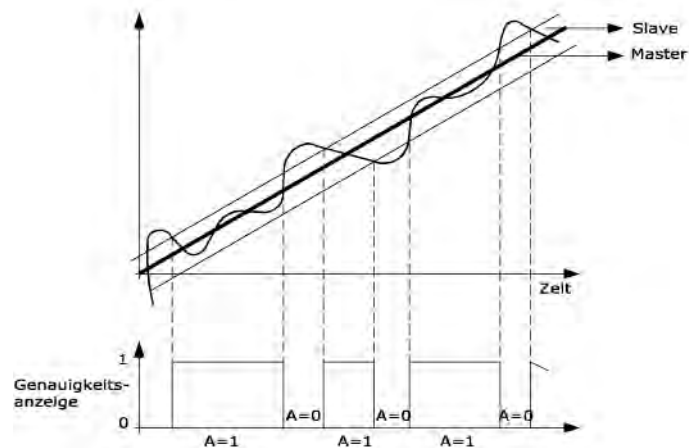
SYNCACCURACY - 55

Inhalt Größe des Genauigkeitsfensters für Positionssynchronisation

Beschreibung Der Parameter gibt an, wie groß die Differenz zwischen aktueller Master- und Slave-Position bei einer Positionssynchronisation (SYNCP und SYNCM) sein darf, damit die geforderte Genauigkeit (ACCURACY) noch erfüllt ist. SYNCERR dagegen liefert den tatsächlichen Synchronisationsfehler des Slaves in Benutzereinheiten.

Ob SYNCACCURACY erfüllt wird, kann mit SYNCSTAT im Programm abgefragt werden.

SYNCACCURACY ist wichtig für die Markersynchronisation um READY melden zu können, da andernfalls vorher n-mal SYNCERR abgefragt und verglichen werden müsste.



Genauigkeitsfenster gesetzt durch SYNCACCURACY.

Die dunkle Linie zeigt die Positionen, denen Master und Slave folgen. SYNCACCURACY setzt das Fenster auf 100 und somit wird das ACCURACY-Flag gesetzt, wenn der Slave innerhalb des Fensters ist.

!!! Die neue Größe des Sync-Fensters wird erst nach einem neuen SYNC-Befehl aktiviert. Dies betrifft nicht nur die Änderung der Größe sondern auch die Aktivierung selbst.

Portabilität Nur bei 32-Bit Steuerungen wie M1 und MACS.

Parametergruppe Achsparameter Synchronisierung AXS

Einheit qc
bzw. im CAM-Mode: BE

Bereich –MLONG ... MLONG

0 ... MLONG Ein positives Vorzeichen liefert den absoluten Wert an SYNCERR.

–MLONG ... –1 Ein negatives Vorzeichen liefert den Synchronisationsfehler an SYNCERR mit Vorzeichen. Daraus lässt sich dann erkennen, ob die Synchronisation voraus- oder nachläuft.

Init-Wert 1000

SYNCREADY - 56

Inhalt	Markeranzahl für READY
Beschreibung	<p>Gibt an wie oft bei einer Markersynchronisation (SYNCM und SYNCCMM) eine Bewertung der Synchronisation mit ACCURACY durchgeführt sein muss, bis READY erfüllt ist.</p> <p>Dabei wird bei jeder Korrektur ACCURACY geprüft. Wenn ACCURACY erfüllt ist, wird 1 addiert, bis die vorgegebene Markeranzahl erreicht ist.</p> <p>Die Synchronisation wird immer erst nach n Markerimpulsen beim Master SYNCMARKM (52) bewertet.</p> <p>ACCURACY und READY können Sie mit SYNCSTAT abfragen.</p>
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	–
Bereich	0 ... 10000
Init-Wert	1

SYNCFAULT - 57

Inhalt	Markeranzahl für Fault
Beschreibung	Gibt an, wie oft bei einer Markersynchronisation (SYNCM und SYNCCMM) nicht ACCURACY auftauchen darf, bis FAULT eintritt. Dieser Zustand kann im Programm mit SYNCSTAT abgefragt werden.
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	–
Bereich	0 ... 10000
Init-Wert	10

SYNCPULSM - 58

Inhalt	Markerabstand Master
Beschreibung	<p>SYNCPULSM gibt an wie viele qc (Master) zwischen zwei Master-Markern liegen bzw. im CAM-Mode den Abstand zwischen Sensor und Arbeitsposition in MU.</p> <p>Wenn man den Drehgeber-Indexpuls als Markersignal benutzt, beträgt der Abstand zwischen zwei Markern die Auflösung [qc] des Drehgebers.</p> <p>Wenn externe Markersignale benutzt werden, können Sie den Markerabstand mit dem Programm „Marker count“ (siehe Datei → Beispiel) messen, falls er nicht bekannt ist.</p> <p>SYNCPULSM gilt nur für Synchronisationen mit Markerkorrektur (SYNCPM und SYNCCMM).</p> <p>Bei einer CAM-Synchronisation wird statt des Abstands zwischen zwei Master-Markern der Abstand des Sensors zur Arbeitsposition in MU angegeben. (Der Abstand ergibt sich automatisch durch die Mastertaktlänge (Mt).)</p> <p>Wenn der Parameter größer als eine Master-Taktlänge (Mt) ist, wird automatisch ein Marker-FIFO-Register für die Handhabung der Markerkorrektur gebildet.</p> $\text{SYNCPULSM} * \text{SYNCPMARKM} * \text{SYNCPFACTS} * n < 2 * \text{MLONG}$ <p>wobei</p> <p>n = Nummer des Markers, der zwischen Master und Slave sein kann, wenn der Slave versucht, den Master beim Start einzuholen.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2.
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	qc im CAM-Mode: MU
Bereich	$0 \dots \frac{2 * \text{MLONG}}{\text{SYNCPMARKM} (52) * \text{SYNCPFACTS} (50) * n}$ <p>wobei n = siehe oben</p>
Init-Wert	500

SYNCMPULSS - 59

Inhalt	Markerabstand Slave
Beschreibung	SYNCMPULSS gibt an wie viele qc (Slave) zwischen zwei Slave-Markern liegen, bzw. im CAM-Mode den Abstand des Sensors zur Arbeitsposition in BE. SYNCMPULSS gilt nur für Synchronisationen mit Markerkorrektur (SYNCM und SYNC CMS). $\text{SYNCMPULSS} * 9 < 2 * \text{MLONG}$
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2
Parametergruppe	Achsparameter Synchronisierung AXS CAM-Editor: Registerkarte Synchron.
Einheit	qc im CAM-Mode: BE
Bereich	0 ... $2 * \text{MLONG} / 9$
Init-Wert	500

SYNCMTYPM - 60

Inhalt	Markertyp Master
Beschreibung	<p>Definiert den Signaltyp für den Master-Marker: Indexpuls des Drehgebers oder externer Marker.</p> <p>SYNCMTYPM gilt nur für Synchronisationen mit Markerkorrektur (SYNCM und SYNCCMM) oder wenn Sie den Befehl MIPOS im Programm verwenden wollen.</p> <p>Ein erweiterter Markertyp realisiert ein logisches UND für den Markereingang und einen zweiten Eingang, z.B. um jeden zweiten Marker mit einer SPS zu synchronisieren. Dies beeinflusst den Befehl MIPOS. MIPOS liefert nur dann einen neuen Wert, wenn die Bedingung erfüllt ist.</p> <p><u>Digital Latch Eingänge (MACS4)</u></p> <p>Die Zehnerstellen werden benutzt, um die digitalen Eingänge für das Latch auszuwählen.</p> <p>SET SYNCMTYPM x(1) 42 // digitalen Eingang 4 mit steigender Flanke als Master-Latch-Eingang für Achse 1 benutzen</p> <p>Falls kein Eingang gesetzt ist, dann ist die Defaultzuweisung denkbar einfach. Das heißt, Encoder 1 benutzt Eingang 1, Encoder 2 benutzt Eingang 2 ... Encoder 6 benutzt Eingang 6.</p> <p>Wenn die Master-Encodersimulation (MENCODERTYPE = 6) mit einer MACS3 benutzt wird, ist es möglich SYNCMTYPM extern zu setzen um externe Marker zuzulassen. Natürlich ist dies nicht so genau, weil nur jede ms erfasst wird, ob ein Ereignis aufgetreten ist oder nicht.</p>
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS2. Parameterwert > 3 ab Controller Version 6.1.14; digitale Latch-Eingänge (MACS4) ab Version 6.5.18
Parametergruppe	AXS Sync Marker
Querverweise	MIPOS, SYNCMTYPS (61)
Einheit	–
Bereich	0 ... 3 ±100 0 = Indexpuls (Drehgeber Z positive Flanke) 1 = Indexpuls (Drehgeber Z negative Flanke) 2 = externer Marker (positive Flanke) 3 = externer Marker (negative Flanke) Zehnerstellen = nm n = digitale Eingangsnummer (1 ... 9) m = Master-Markertyp (0 ... 3) 100 ... 103 = wie 0...3, jedoch nur gültig wenn Eingang 1 high 200 ... 203 = wie 0...3, jedoch nur gültig wenn Eingang 2 high ... -100...-103 = wie 0...3, jedoch nur gültig wenn Eingang 1 low -200...-203 = wie 0...3, jedoch nur gültig wenn Eingang 2 low ...
Init-Wert	0

SYNCMTYPS - 61

Inhalt	Markertyp Slave
Beschreibung	<p>Definiert den Signaltyp für den Slave-Marker: Indexpuls des Drehgebers oder externer Marker.</p> <p>SYNCMTYPS gilt nur für Synchronisationen mit Markerkorrektur (SYNCM und SYNCCMS) oder wenn Sie den Befehl IPOS im Programm verwenden wollen.</p> <p><u>Digitale Latch Eingänge (MACS4)</u></p> <p>Die Zehnerstellen werden benutzt, um die digitalen Eingänge für das Latch auszuwählen.</p> <p>Falls kein Eingang gesetzt ist, dann ist die Defaultzuweisung denkbar einfach. Das heißt, Encoder 1 benutzt Eingang 1, Encoder 2 benutzt Eingang 2 ... Encoder 6 benutzt Eingang 6.</p> <p>SYNCMTYPS unterstützt die folgenden Werte, wenn ENCODERTYPE = 9</p> <p>0– virtuelle Marker werden mit einem Abstand von ENCODERFREQ erzeugt.</p> <p>n2– Bei positiver Flanke von Eingang n wird die aktuelle Position als Markerposition genommen.</p> <p>n3– Bei negativer Flanke von Eingang n wird die aktuelle Position als Markerposition genommen.</p> <p>Die Genauigkeit ist natürlich auf 1 ms begrenzt, weil dies die interne Update-Rate ist.</p>
Portabilität	<p>Nur bei 32-Bit Steuerungen wie M1 und MACS2.</p> <p>Digitale Latch Eingänge (MACS4) sind ab Version 6.5.18 möglich.</p>
Parametergruppe	AXS Sync Marker
Querverweise	IPOS, SYNCMTYPM (60)
Einheit	–
Bereich	0 ... 3
	0 = Drehgeber Z positive Flanke (Indexpuls)
	1 = Drehgeber Z negative Flanke (Indexpuls)
	2 = externer Marker positive Flanke
	3 = externer Marker negative Flanke
	Zehnerstellen = nm
	n = digitale Eingangsnummer (1 ... 9)
	m = Master-Markertyp (0 ... 3)
Init-Wert	0

SYNCMSTART - 62

Inhalt	Startverhalten für Synchronisationen mit Markerkorrektur
Beschreibung	<p>SYNCMSTART gibt an, ob beim Starten der Synchronisation auf den jeweils voreilenden, nachfolgenden oder auf den dichtesten Markerimpuls des Masters aufsynchronisiert werden soll.</p> <p>SYNCMSTART gilt nur für Synchronisationen mit Markerkorrektur (SYNCM and SYNCCMM).</p> <p>Wenn man SYNCM mit SYNCMSTART 7 (oder 1007) startet wird zuerst geprüft ob die Markerpositionen bereits bekannt sind. (Dies wird über ein Indexflag geprüft, welches bei einem Encoder-Wechsel mit SET SYNCMTYPE zurückgesetzt wird).</p> <p>Falls die Slave-Position nicht bekannt ist und nicht gefahren wird, wird mit der ruckbegrenzten Rampe (falls RAMTYPE 2) gestartet und versucht auf Master-Geschwindigkeit zu kommen. Sobald der Slave gesehen wird, wird versucht ein Poly5 zu rechnen und zu starten. Falls das nicht klappt, wird weiter gefahren bis die Mastergeschwindigkeit erreicht ist und dann ein SYNCMSTART 5 durchgeführt.</p> <p>Falls die Master- und Slave-Positionen bekannt sind, wird sofort versucht ein Poly5 zu rechnen. Da dieses aber nicht zu jedem Zeitpunkt gestartet werden kann (es muss ein bestimmtes Verhältnis zwischen Master- und Slave-Strecke erfüllt sein) kommt es vor, dass noch etwas gewartet wird. (Entweder im Stillstand, falls vorher gestanden war, oder mit kontinuierlicher Geschwindigkeit, falls vorher gefahren wurde.) Sobald das Verhältnis stimmt, wird das Polynom abgefahren.</p> <p>Sollte sich während einer Polynomfahrt die Geschwindigkeit des Masters um mehr als 10 % ändern seit das Polynom gestartet wurde, wird das Polynom abgebrochen und mit SYNCMSTART 5 bzw. 1005 fortgefahren. Dasselbe geschieht, wenn die Master-Geschwindigkeit zum Startzeitpunkt null oder nahezu null ist.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2. Parameter 7 und 1007 werden ab Firmware 6.7.14 unterstützt.
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	–
Bereich	0 ... 7 und 1000 ... 1007, 2000
	0 = Führender Marker: Der Slave-Marker, der dem ersten Master-Marker (nach SYNCM) folgt, wird mit dem ersten Master-Marker abgeglichen.
	1 = Folgender Marker: Der erste Slave-Marker (nach SYNCM) wird mit dem folgenden Master-Marker abgeglichen.
	2 = Dichtester Marker: Nach Erreichen der Master-Geschwindigkeit werden die nächsten zwei Marker abgeglichen. (Korrektur durch Aufholen oder Abbremsen).
	3 = Nach Erreichen der Master-Geschwindigkeit wird der nächste Slave-Marker mit dem davor liegenden Master-Marker abgeglichen. (Korrektur durch Aufholen).
	4 = Nach Erreichen der Master-Geschwindigkeit wird der nächste Slave-Marker mit dem nachfolgenden Master-Marker abgeglichen. (Korrektur durch Abbremsen).
	5 = Nach Erreichen der Master-Geschwindigkeit wird der nächste Slave-Marker mit dem Master-Marker abgeglichen, der am dichtesten folgt. (Korrektur durch Aufholen oder Abbremsen, je nach kürzestem Abstand)
	6 = Nach dem Befehl SYNCM werden die ersten zwei Marker genommen und auf diese aufsynchronisiert.
	7 = Start mit einem Poly5 um den Master exakt auf einer Markerposition zu erreichen.

Parameter-Referenz ♦ Achsparameter im Detail

1000 ... 1007 = wie oben, aber ein bestehender Offset (SYNCPOSOFFS) wird erst bei der nächsten Markerkorrektur angewendet.

2000 = im CAM-Mode: Das Zählen (Beobachten) der Masterpulse in MU beginnt mit dem Master-Marker.

!!! Der Parameter 2000 wirkt nur bei Kurvensynchronisationen (CAM-Modus).

Init-Wert 0

REVERS - 63

Inhalt	Definiert das Reversierungsverhalten der geregelten Achse, ... die Nutzung von bipolarem oder unipolarem Sollwert und ... die ID für Bus-Teilnehmer (DRIVETYPE ungleich 0).
Beschreibung	<p>Der Parameter REVERS legt drei Charakteristika von geregelten Achsen fest.</p> <p>Erstens <u>charakterisiert</u> der Parameter REVERS das <u>Verhalten beim Rückwärtsfahren</u>, d.h. Fahren in negativer Richtung. Die Zulässigkeit der Rückwärtsfahrt kann dabei wie folgt definiert sein:</p> <ul style="list-style-type: none">- Rückwärtsfahrt ist prinzipiell erlaubt (= Default).- Rückwärtsfahrt ist nur erlaubt, wenn der Master rückwärts fährt.- Rückwärtsfahrt ist grundsätzlich gesperrt. <p>Die entsprechende Einschränkung ist immer gültig, d.h. bei der Antriebssynchronisation (SYNCP, SYNCV, SYNCM,, SYNCC etc.), den Positionierbefehlen (POSA, POSR), dem Drehzahlbefehl (CVEL) und auch bei einer Testfahrt mit dem Oszilloskop. (Tipp: Wenn Sie das automatische Reversieren bei der Testfahrt verhindern wollen, stellen Sie den Parameter auf 1 oder 2 ein.)</p> <p>Zweitens <u>charakterisiert</u> der Parameter REVERS die <u>Art der analogen Sollwert-Ausgabe</u>. Insbesondere einige Frequenzrichter haben nur einen unipolaren Sollwert-Eingang (d.h. 0...10 V statt ± 10 V). Über den Parameter REVERS kann hierfür definiert werden, dass das analoge Sollwert-Ausgangssignal der Steuerung im Bereich 0...10V liegt und die Drehrichtungsinformation (für das Reversieren) über den digitalen Ausgang 1 erfolgt. Ein theoretisch „negativer“ Sollwert (für die Rückwärtsfahrt) wird dann in ein positives analoges Signal (0...10 V) am Sollwert-Ausgang umgesetzt und der digitale Ausgang 1 auf 24 V als Reversierungsinformation für den Frequenzrichter gesetzt. Bei Vorwärtsfahrt und somit auch theoretisch positivem Sollwert wird dieser unverändert am analogen Ausgang ausgegeben und der digitale Ausgang 1 auf 0 V gesetzt.</p> <p>Als <u>dritte Charakteristik</u> der geregelten Achse legt der Parameter REVERS bei Sollwert-Ausgaben über den Bus (d.h. Parameter DRIVETYPE) ungleich 0) die <u>Bus-ID des Teilnehmers</u> fest. Der Parameterwert berechnet sich hierbei aus der Bus-ID (unabhängig von dem Bus-Anschluss) multipliziert mit 100 plus eventuellen Einschränkungen der Rückwärtsfahrt. Falls mit Bus-Teilnehmern gearbeitet wird, sollte der Parameter REVERS gesetzt werden bevor der Parameter DRIVETYPE ungleich 0 gesetzt wird.</p> <p>Der abschließende Parameterwert von REVERS berechnet sich aus der Addition der einzelnen Werte der Charakteristika.</p> <p>REVERS <u>negative Werte</u> -10 .. -90:</p> <p>Um FU zu steuern, benötigen die Antriebe zwei unterschiedliche aktivierte Signale für die positive und negative Richtung. Dafür werden zwei Ausgänge benutzt, beginnend mit REVERS/10 und dem nächsten Ausgang der Reihe nach (REVERS/10 + 1).</p> <p>Wenn der Sollwert positiv ist, wird der Ausgang REVERS/10 auf 1 gesetzt und der andere auf 0. Wenn der Sollwert negativ ist, wird der Sollwert invertiert und die Ausgänge werden umgekehrt gesetzt. Falls der Motor aus ist, werden beide Ausgänge zurückgesetzt.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2. Parameterwert > 12 ab Controller Version 6.1.14; negative Werte ab Firmware 6.5.16.
Parametergruppe	AXR Pos. Regelung
Einheit	-
Bereich	0 ... 2, 10 ... 12, 100 ... 12702
	Für die Bestimmung des Parameterwerts von REVERS müssen die <u>Werte der einzelnen Charakteristika addiert</u> werden:

1. Charakteristik: Reversierungsverhalten

- 0 = Reversieren immer erlaubt
- 1 = Reversieren nur erlaubt, wenn der Master rückwärts fährt
- 2 = Reversieren gesperrt

2. Charakteristik: Art des Sollwerts

- 0 = Bipolares Sollwertsignal ($\pm 10V$)
- 10 * (Nr. des digitalen Richtungsausgangs) = Unipolares Sollwertsignal (0..10V) + digitales Richtungssignal am definierten Ausgang

3. Charakteristik: Bus-ID des Teilnehmers

- 0 = Sollwertausgabe am lokalen Ausgang (DRIVETYPE = 0)
- Bus-ID * 100 = Bus-Teilnehmer (DRIVETYPE ungleich 0)

Die Bus-Teilnehmer müssen bei Verwendung von Steuerungen mit mehreren, getrennten CAN-Bus-Kreisen (z.B. MACS3) zwingend an dem so genannten Slave-Bus angeschlossen sein. Der Einbezug der Bus-Offset-Kennung im Parameterwert von REVERS entfällt durch diese feste Zuordnung.

Beispiele Beispiel 1:

- Reversieren immer erlaubt
 - Bipolarer Sollwert (+/- 10V)
 - Sollwert-Ausgabe am lokalen Ausgang
(d.h. Parameter DRIVETYPE gleich 0)
- => REVERS = 0 + 0 + 0 = 0

Beispiel 2:

- Reversieren nur erlaubt, wenn der Master rückwärts fährt
 - Unipolares Sollwert-Signal (0..10V) plus Richtungssignal an Ausgang 1
 - Sollwert-Ausgabe am lokalen Ausgang
(d.h. Parameter DRIVETYPE gleich 0)
- => REVERS = 1 + (10 * 1) + 0 = 11

Beispiel 3:

- Reversieren immer erlaubt
 - Unipolares Sollwert-Signal (0..10V) plus Richtungssignal an Ausgang 2
 - Sollwert-Ausgabe am lokalen Ausgang
(d.h. Parameter DRIVETYPE gleich 0)
- => REVERS = 0 + (10 * 2) + 0 = 20

Beispiel 4:

- Reversieren immer erlaubt
 - Bipolarer Sollwert
 - Sollwert-Ausgabe via Bus an Teilnehmer mit ID 1
(d.h. Parameter DRIVETYPE ungleich 0)
- => REVERS = 0 + 0 + (1 * 100) = 100

Beispiel 5:

- Reversieren gesperrt
 - Bipolarer Sollwert
 - Sollwert-Ausgabe via Bus an Teilnehmer mit ID 96
(d.h. Parameter DRIVETYPE ungleich 0)
- => REVERS = 2 + 0 + (96 * 100) = 9602

Init-Wert 0

O_AXMOVE - 64

Inhalt	Ausgang für Fahrbefehl in Aktion
Beschreibung	<p>Mit diesem Parameter setzen Sie die Ausgangsnummer, die durch die O_AXMOVE Funktion kontrolliert werden muss. Der Ausgang ist immer aktiviert, sobald ein Fahrbefehl aktiv ist, unabhängig in welchem Modus (Positions-, Geschwindigkeits- oder Synchronisationsbefehl).</p> <p>Diese Funktion eignet sich nicht für die Motorüberwachung, denn der Motor könnte stillstehen, obwohl die Steuerung in Bewegung ist.</p> <p>Eine positive Zahl bedeutet, dass der Ausgang high (24 Volt) ist, wenn die Funktion aktiv ist. Eine negative Zahl bedeutet, dass der Ausgang low (0 Volt) ist, wenn die Funktion aktiv ist.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2.
Parametergruppe	AXI vordefinierte Ein-/Ausgänge
Einheit	–
Bereich	–8 ... 8
Init-Wert	0

SYNCVFTIME - 65

Inhalt	Geschwindigkeitsfilter										
Syntax	SET SYNCVFTIME x(n) wert n = Achsnummer wert = Filterzeit Konstante										
Beschreibung	<p>Dieser Parameter konfiguriert den Geschwindigkeitsfilter, der für die Geschwindigkeits-synchronisation verwendet wird. Da bei einer Geschwindigkeitssynchronisation nur mit der jeweils aktuellen Master-Geschwindigkeit gearbeitet wird und diese sehr kleine Werte annehmen kann (z.B. 2 qc/ms), wirkt sich eine kleine Schwankung der Geschwindigkeit bereits dramatisch aus. Um dies zu glätten, wird die folgende Filterfunktion verwendet:</p> $\text{Cmdvel} = \text{Old_Cmdvel} + (\text{Actvel} - \text{Old_Cmdvel}) * \text{st}/\text{tau_filt}$ <p>Hierbei gilt:</p> <table><tr><td>Cmdvel</td><td>= Sollgeschwindigkeit</td></tr><tr><td>Old_Cmdvel</td><td>= Letzte Sollgeschwindigkeit</td></tr><tr><td>Actvel</td><td>= Aktuelle Geschwindigkeit des Masters</td></tr><tr><td>[st]</td><td>= Abtastzeit (fest 1 ms)</td></tr><tr><td>tau_filt</td><td>= Filterzeit Konstante</td></tr></table> <p>Dabei wird der Wert für τ_filt standardgemäß aus einer Tabelle genommen, in Abhängigkeit von der Drehgeberauflösung des Masters. Dieser Wert kann durch den Parameter SYNCVFTIME überschrieben werden und wird immer dann verwendet, wenn SYNCVFTIME ungleich Null ist.</p> <p>Wird der Geschwindigkeitsfilter mit einer negativen Zahl definiert, gilt der entsprechende Wert auch für eine Winkel-/Positionssynchronisation SYNCP und für eine mit Markerkorrektur SYNCM.</p> <p>Es wird in diesem Fall ebenso gefiltert wie oben beschrieben, zusätzlich jedoch der gemachte Fehler aufsummiert. Diese Fehlersumme wird jeweils zu $1000/(\tau*10)$ in die Berechnung mit einbezogen, so dass über längere Zeiträume keine Positionsabweichung entstehen kann.</p> <p>Der von SYNCERR zurück gelieferte Wert enthält immer den gemachten Fehler, so dass dieser auch bei der Bewertung der Synchronität einfließt. Im Fall einer Markerkorrektur wird der Korrekturwert langsamer und mit demselben Faktor wie die Fehlersummen ausgeglichen.</p> <p>Setzt man zum Beispiel einen Filterfaktor von -100000 (100 ms) wird eine Markerkorrektur innerhalb von 1 Sekunde ($100 \text{ ms} * 10$) ausgeglichen. Diese ermöglicht eine „Zähmung“ der Synchronisation ohne die Beschleunigung einzuschränken.</p>	Cmdvel	= Sollgeschwindigkeit	Old_Cmdvel	= Letzte Sollgeschwindigkeit	Actvel	= Aktuelle Geschwindigkeit des Masters	[st]	= Abtastzeit (fest 1 ms)	tau_filt	= Filterzeit Konstante
Cmdvel	= Sollgeschwindigkeit										
Old_Cmdvel	= Letzte Sollgeschwindigkeit										
Actvel	= Aktuelle Geschwindigkeit des Masters										
[st]	= Abtastzeit (fest 1 ms)										
tau_filt	= Filterzeit Konstante										
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS2										
Parametergruppe	Achsparameter Synchronisierung AXS										
Querverweis	SYNCVFLIMIT										
Einheit	τ_filt (μs)										
Bereich	-MLONG ... MLONG -999 ... 999 = Standardtabelle										

Parameter-Referenz ♦ Achsparameter im Detail

Standardtabelle	Drehgeber Auflösung	τ_{filt} (μs)
	250	39500
	256	38600
	500	19500
	512	19000
	1000	9500
	1024	9300
	2000	4500
	2048	4400
	2500	3500
	4096	1900
	5000	1400

Init-Wert 0

d.h. es gilt o.g. Standardtabelle

SYNCVELREL - 66

Inhalt	Relative Geschwindigkeitsbegrenzung Slave
Syntax	SET SYNCVELREL x(n) wert n = Achsnummer wert = Prozentwert
Beschreibung	<p>Dieser Parameter gibt an, um wie viel Prozent der Folgeantrieb von der Geschwindigkeit des Masters abweichen darf, während er versucht die Synchronisation wieder herzustellen. Zum Beispiel bei einer Änderung von SYNCPOSOFFS (54) oder beim Start der Synchronisation oder bei der Korrektur der Abweichung bei der Markerauswertung. Dabei gilt Folgendes:</p> <p>Muss der Slave aufholen, fährt er mit der maximal erlaubten Drehzahl, wobei dies entweder die mit VEL eingestellte Drehzahl ist oder die durch</p> $\text{MAVEL} + \text{MAVEL} * \text{SYNCVELREL}/100$ <p>berechnete, je nachdem welche von beiden kleiner ist. (MAVEL ist aktuelle Master-Geschwindigkeit).</p> <p>Muss der Slave abbremsen und auf den Master warten, fährt er mindestens mit der Drehzahl</p> $\text{MAVEL} - \text{MAVEL} * \text{SYNCVELREL}/100.$ <p>Das heißt, wenn SYNCVELREL zum Beispiel 50 ist, wird der Folgeantrieb nicht langsamer fahren als MAVEL/2.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2.
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	%
Bereich	0 ... 100 0 = Funktion ist ausgeschaltet, d.h. keine Beschränkung
Init-Wert	0

MENCODERTYPE - 67

Inhalt	Definiert die Signalquelle des Master-Gebers.
Beschreibung	<p>MENCODERTYPE legt die Signalquelle des Master-Gebers fest. Die Information kann über die folgenden „Kanäle“ bezogen werden und muss entsprechend definiert werden:</p> <ul style="list-style-type: none">- Lokaler Encoder-Eingang der Steuerung- CANopen-Drehgeber via Prozessdatenwort (PDO)- CANopen-Leistungsendstufe via Prozessdatenwort (PDO)- Encodertyp via Prozess Datenwort (PDO) auf dem EtherCAT-Bus <p>Als Drehgeber können Inkremental- und Absolutgeber eingesetzt werden.</p> <p>Negative MENCODERTYPEen legen fest, dass der Master-Encoder der Sollwert einer anderen Achse im System sei. Dies eignet sich besonders, wenn eine Achse der Master ist und die zweite Achse der Slave. Wenn dann der Sollwert statt des Istwertes benutzt wird, erhalten Sie laufruhigere Fahrbewegungen.</p> <p>Der verbundene Encoder hat nicht die identischen Werte, aber er zählt von der vorherigen Position weiter. Dies erlaubt es auch DEFMORIGIN für den Master oder DEFORIGIN für die Ausgangsachse zu nutzen.</p> <p>SYSVAR PFG_COMMANDPOS (4311) / PFG_OLDCOMMANDPOS (4312) müssen verwendet werden, um die oben beschriebene Verbesserung zu realisieren. Sie liefern zwar die Ist- und die letzte Sollposition (wie 4096), aber sie liefern die exakten Werte für jede ms, sogar wenn PROFTIME größer als 1000 ist.</p>
Besonderheiten bei Absolutgebern	Mit Absolutgebern können die Befehle DEFMORIGIN und MIPOS nicht benutzt werden.
Mastersimulation	<p>Mit MENCODERTYPE = 6 aktiviert eine Mastersimulation, bei der zum Beispiel die Master-Position über den Bus gelesen werden kann. Die Geschwindigkeit in Quadcounts pro Sample-Time wird hierbei durch einen Schreibzugriff auf die Systemvariablen SYSVAR[4105] eingestellt. Bei einem Lesezugriff auf die Systemvariablen SYSVAR[4105] wird die simulierte Master-Position zurückgeliefert.</p> <p>Wenn die Master-Encodersimulation mit einer MACS3 benutzt wird, ist es möglich SYNCMTYPM extern zu setzen um externe Marker zuzulassen. Natürlich ist dies nicht so genau, weil nur jede ms erfasst wird, ob ein Ereignis aufgetreten ist oder nicht.</p>
Device-Typen	<p>Bei Drehgeber-Informationen, die über den Bus bezogen werden, erfolgt die Parameterwert-Berechnung anhand der, im Parameter DRIVETYPE definierten Device-Typen nach folgender Formel:</p> $\text{SET MENCODERTYPE nn}$ <p>wobei $nn = \text{guard} * (\text{busoffset} + \text{Device-Typ} * 1000 + \text{id})$</p> <p>Die Drehgebertypen mit Device-Typ 1 bis 49 werden genauso behandelt wie Inkrementalgeber, bei denen die relative Änderung des über den Bus übertragenen Positionswert (PDO) zur letzten Position addiert wird.</p> <p>Die Drehgebertypen mit Device-Typ 50 werden behandelt wie Absolutgeber, bei denen ein Überlauf durch den Drehgeber auftritt (CANopen Drehgeber-Definition).</p>

Ein weiterer Encodertyp bekommt seine aktuelle Position und Indexposition von einem PDO auf dem EtherCAT Bus:

```
SET MENCODERTYPE nn  
wobei nn = Busoffset + Device-Typ * 1000 + PDO no. + Long Offset
```

Beispiel:

```
SET MENCODERTYPE 251000 // benutzt PDO[0] als EtherCAT Encoder
```

Behandlung der automatisch erzeugten CAN-Objekte	Für die Informationsübertragung der Istposition über den Bus werden die benötigten CAN-Objekte (PDOs, GUARD-, SYNC-Objekt) automatisch angelegt und der Teilnehmer initialisiert (NMT0). Diese im Hintergrund ausgeführte Objekt-Erzeugung entspricht im Wesentlichen dem CANINI-Befehl. Im Gegensatz zur applikationsseitigen Objekt-Erzeugung mit dem CANINI Befehl, bleiben die mit MENCODERTYPE automatisch erzeugten Objekte jedoch auch bei einem erneuten applikationsseitigen CANINI unverändert erhalten. Ebenfalls werden die mit MENCODERTYPE automatisch erzeugten Objekte auch durch ein CANDEL -1 nicht gestoppt. Die automatisch erzeugten Objekte können innerhalb der Applikation nur durch einen erneuten, abweichenden SET MENCODERTYPE Befehl gelöscht bzw. umkonfiguriert werden. Auch beim Programmabbruch (mittels ESC) bleiben die Objekte noch bis zum nächsten Programmstart erhalten. Erst beim Neustart eines Programms werden die Objekte gelöscht und entsprechend der Einstellung im permanenten Parametersatz oder entsprechend der Definition im Programm neu angelegt.
Guarding	Bei konfiguriertem Guarding (d.h. DRIVETYPE > 0), wird der Bus-Teilnehmer überwacht und eine fehlende Rückmeldung auf das GUARD-Objekt (z.B. aufgrund des Ausfalls des Teilnehmers) führt zu dem Fehler 88. Es liegt somit in der Verantwortung des Anwenders, darauf zu achten, dass zum Zeitpunkt der Definition des Parameters MENCODERTYPE gegebenenfalls der Parameter REVERS (entsprechend der Bus-ID) korrekt gesetzt ist, der Bus-Teilnehmer korrekt konfiguriert (Baudrate, Bus-ID), angeschlossen, eingeschaltet und für die Kommunikation bereit ist.
!!!	Falls bei einem MENCODERTYPE > 0 der Bus-Teilnehmer ausfällt, muss von der Applikation (in der Fehlerbehandlungsroutine) die Überwachung durch das Setzen des negativen Werts MENCODERTYPE abgeschaltet werden, bis der Fehler auf Seiten des Teilnehmers behoben ist. Ansonsten tritt nach jedem Löschen des Steuerungsfehlerzustands 88 (mit ERRCLR) sofort wieder die identische Fehlermeldung auf.
PDO-Mapping für CANopen-Antriebe	Bei der Verwendung von CANopen-Leistungsendstufen für die Rückmeldung der Drehgeber-Information muss das so genannte PDO-Mapping, d.h. welche Objekte der Leistungsendstufe in die übertragenen PDOs verlinkt sind, in der Applikation vor dem Setzen des Parameters MENCODERTYPE ausgeführt werden. Die Wahl der am besten geeigneten Objekte hängt von der CANopen-Leistungsendstufe ab. Die zub machine control AG gibt hierzu gerne Beispiel-Sourcecode passend zu den unten genannten Device-Typen ab. Nennen Sie uns hierzu bitte die eingesetzte Leistungsendstufe per e-Mail (info@zub.ch).
SYNC Periodizität	Es ist zu beachten, dass die Periodizität des SYNC-Telegramms bis zur Firmware-Version 6.5.16 durch den kleinsten Parameterwert von TIMER definiert war, der bei einer Achse eingestellt wurde. Bei aktuellen Firmware-Versionen (6.5.16 oder höher) erfolgt die Einstellung der SYNC-Rate über den globalen Parameter CANSYNCTIMER. Dieser Parameter muss auf einen Wert gesetzt werden, der auf den langsamsten CAN-Bus Teilnehmer abgestimmt ist.
Positionswerte-Erfassung anhand von Marker-Signalen	Bei Markersynchronisationen oder Positionierungen, die von der Echtzeiterfassung von Positionswerten anhand von Marker-Signalen (siehe Befehl MIPOS, Parameter SYNCMTYPM) abhängig sind, wird empfohlen die entsprechenden Encoder-Signale auf die lokalen Encoder-Eingänge der Steuerung zu legen. Nur in diesem Fall ist das (Hardware-)Latching der Positionswerte mit der maximalen Genauigkeit von 1 qc möglich.

!!! Bei Encoder-Informationen, die über den Bus übertragen werden, verringert sich die Genauigkeit der Markererfassung in Abhängigkeit von der Zykluszeit des Bus (Parameter TIMER, CANSYNCTIMER) und der Drehzahl des Antriebs.

Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2. Master-Simulation über CAN-Bus ab Controller Version 6.1.14; absolute passive Encoder (SSI) ab Version 6.7.05 (MACS4); negative MENCODERTYPEen ab Version 6.7.03.
Parametergruppe	Achsparameter Encoder AXE
Querverweise	ENCODERTYPE, DRIVETYPE, REVERS, MIPOS, CANINI, TIMER, CANSYNCTIMER
Einheit	–
Bereich	–MLONG ... MLONG 0 = Inkrementalgeber 1 = Absolutgeber, Standard ca. 262 kHz 2 = Absolutgeber, ca. 105 kHz 5 = Sinus / Cosinus Encoder 6 = Master-Encodersimulation: Die Geschwindigkeit wird über die SYSVAR[4105] in Quadcounts pro Sample-time angegeben. In der Standardkonfiguration mit 1 ms Sample-time ist die Geschwindigkeit also in [qc/ms] anzugeben. !!! Da die SYSVAR[4105] ebenfalls für die Master-Position verwendet wird, liefern Lese- und Schreibzugriffe auf diese SYSVAR unterschiedliche Ergebnisse. Bei einem Schreibzugriff (SYSVAR[4105] = ...) wird die Geschwindigkeit für die Master-Simulation gesetzt. Bei einem Lesezugriff (... = SYSVAR[4105]) wird die Masterposition [qc] zurückgeliefert. 10 = Inkrementalgeber am lokalen Encoder-Anschluss 1 20 = Inkrementalgeber am lokalen Encoder-Anschluss 2 30 = Inkrementalgeber am lokalen Encoder-Anschluss 3 > 1000 = nn = Drehgeber-Information via Bus übertragen nn = guard * (busoffset + Device-Typ * 1000 + id) guard = -1, +1 (ohne / mit Guarding) busoffset = 100000, 0 (Slave-Bus, Master-Bus) Device-Typ (siehe auch unter DRIVETYPE) 0 = Standard Antrieb (analoger Ausgang) 1 = CAN-Antrieb des Typs Lenze (Standard) 2 = CANopen-Servoverstärker nach DS402 3 = CANopen-Servoverstärker maxon EPOS 4 = CANopen-Servoverstärker zub DSA 5 = CAN-Antrieb des Typ Lenze (Sollwert-Limit) 50 = CANopen-Drehgeber id = 1...127 = Bus-ID, mit Device-Typ 1 ...50 und Datenaustausch über PDO 2 id = 257...511 = Bus ID 1...127 mit Device-Typ 1, aber Datenaustausch über PDO 3

> 100000 = nn = Encoder Information via EtherCAT erhalten

nn = Busoffset + Device-Typ * 1000 + PDO nr. + Long offset

Device-Typ =

51 = Encoder Information via PDO an EtherCAT

Beispiel:

```
SET ENCODERTYPE 251100 // pdo[1] als EtherCAT Encoder benutzen
```

```
// 200000 = EtherCAT Bus
```

```
// 51(000) = EtherCAT Encoder
```

```
// 100er = PDO nr (z.B. 200 = pdo 2)
```

```
// 1er = Offset in PDO in Longs (0 = 1. Long im PDO benutzen)
```

Der erste Long auf diesem Platz wird für die Position benutzt, der zweite für die Index-Position.

z.B. SET ENCODERTYPE 251105

```
// PDO1 Long mit Offset 5 (PDO[6]) für Position benutzen,
```

```
// und Long mit Offset 6 (PDO[7]) für die Index-Position
```

Init-Wert 0

Programmbeispiel Datei → **Beispiel:** syncc_msim.m

SYNCMWINM - 68

Inhalt	Toleranzfenster für das Auftreten der Master-Marker (Marker Monitoring)
Beschreibung	<p>Das Master-Marker Toleranzfenster SYNCMWINM gibt an, wie groß die erlaubte Toleranz für das Auftreten der Marker ist.</p> <p>Mit der Werkseinstellung 0 wird das Fenster nicht überwacht, das heißt es wird immer auf den nächsten Marker synchronisiert, auch wenn dieser einen wesentlich größeren Abstand hat.</p> <p>Mit jeder anderen Einstellung werden nur Marker akzeptiert, die innerhalb des Fensters liegen. Wenn innerhalb des Toleranzfensters kein Marker kommt, wird das entsprechende Flag (SYNCSTAT) gesetzt und keine Markerkorrektur durchgeführt. Es wird auch der entsprechende andere Marker ignoriert und erst beim nächsten Mal wieder korrigiert – also kein Aufholen zum nächsten Marker.</p> <p>Nach dem Start von SYNCM oder SYNCCSTART beginnt die Überwachung erst nachdem der erste Marker gefunden ist.</p> <p>!!! Änderungen des Parameters werden sofort aktiv – nicht erst nach dem nächsten SYNCM Befehl.</p>
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	qc bzw. im CAM-Mode: MU
Bereich	0 ... MLONG bzw. max. Markerabstand SYNCMPULSM (58) 0 = Das Fensters wird nicht überwacht. 1 ... MLONG = Es wird nur ein Marker akzeptiert, der innerhalb des Fensters liegt. Wenn innerhalb des Toleranzfensters kein Marker kommt, wird das entsprechende Flag (SYNCSTAT) gesetzt und keine Markerkorrektur durchgeführt. Dieses Flag kann mit einem Interrupt (ON STATBIT) zurückgesetzt werden.
Init-Wert	0
Beispiel	Markerabstand SYNCMPULSM = 30000 Toleranzfenster SYNCMWINM = 1000 Es wird nur der Marker akzeptiert, der innerhalb des Intervalls von 29000 bis 31000 liegt.

SYNCMWINS - 69

Inhalt	Toleranzfenster für das Auftreten der Slave-Marker.
Beschreibung	<p>Das Slave-Marker Toleranzfenster SYNCMWINS gibt an, wie groß die erlaubte Toleranz für das Auftreten der Marker ist.</p> <p>Mit der Werkseinstellung 0 wird das Fenster nicht überwacht, das heißt es wird immer auf den nächsten Marker synchronisiert, auch wenn dieser einen wesentlich größeren Abstand hat.</p> <p>Mit jeder anderen Einstellung werden nur Marker akzeptiert, die innerhalb des Fensters liegen. Wenn innerhalb des Toleranzfensters kein Marker kommt, wird das entsprechende Flag (SYNCSTAT) gesetzt und keine Markerkorrektur durchgeführt. Es wird auch der entsprechende andere Marker ignoriert und erst beim nächsten Mal wieder korrigiert – also nicht zum nächsten Marker aufgeholt.</p> <p>Nach dem Start von SYNCM oder SYNCCSTART beginnt die Überwachung erst nachdem der erste Marker gefunden ist.</p>
!!!	Änderungen des Parameters werden sofort aktiv – nicht erst nach dem nächsten SYNCM Befehl.
Parametergruppe	Achsparameter Synchronisierung AXS
Einheit	qc bzw. im CAM-Mode: BE
Bereich	0 ... MLONG bzw. max. Markerabstand SYNCMPULSS (59)
	0 = Das Fensters wird nicht überwacht.
	1 ... MLONG = Es wird nur ein Marker akzeptiert, der innerhalb des Fensters liegt. Wenn innerhalb des Toleranzfensters kein Marker kommt, wird das entsprechende Flag (SYNCSTAT) gesetzt und keine Markerkorrektur durchgeführt. Dieses Flag kann mit einem Interrupt (ON STATBIT) zurückgesetzt werden.
Init-Wert	0

ESCCOND - 70

Inhalt	Verhalten bei Programmabbruch
Beschreibung	<p>Mit diesem Parameter legen Sie fest, wie die Steuerung bei Programmabbruch durch Esc reagieren soll. Esc bricht nur das Programm ab, alle Ausgänge bleiben im aktuellen Status (Werkseinstellung).</p> <p>Die beiden Alternativen ermöglichen es, Bedingungen für die Ausgänge festzulegen. Wenn zum Beispiel eine Pumpe angesteuert wird, würde nach Esc bei „1“ die Pumpe auch abgeschaltet werden, während sie bei „2“ weiter läuft.</p>
Parametergruppe	AXI vordefinierte Ein-/Ausgänge
Einheit	–
Bereich	0 ... 2
	0 = Geregelter Stopp Der Motor wird mit maximaler Verzögerung gestoppt, die Ausgang Bremse wird aktiviert (falls definiert), die Master-Simulation wird gestoppt. Die Ausgänge bleiben im aktuellen Status.
	1 = Geregelter Stopp + Ausgänge = 0 Wie 0, aber alle Ausgänge werden auf "0" gesetzt. <u>Ausnahme</u> : Der Ausgang Bremse wird – falls definiert – immer aktiviert.
	2 = Geregelter Stopp + Ausgänge = 1 Wie "0", aber alle Ausgänge werden auf "1" gesetzt; <u>Ausnahme</u> : Der Ausgang Bremse wird – falls definiert – immer aktiviert.
Init-Wert	0

ENCODERCLOCK - 73

Inhalt	Interne oder externe Takterzeugung für SSI Encoder.
Beschreibung	Es kann ein absoluter passiver Encoder (SSI) eingesetzt werden. Wenn ENCODERCLOCK auf 0 gesetzt ist, wird kein Takt erzeugt und der passive Modus ist aktiv.
Portabilität	Parameter ist ab Version 6.7.05 (MACS4) verfügbar.
Parametergruppe	AXE Gebersystem
Einheit	–
Bereich	0 = externe Takterzeugung für SSI Encoder 1 = interne Takterzeugung für SSI Encoder
Init-Wert	1

ENCODERFREQ - 74

Inhalt	Definiert die Frequenz für SSI Encoder.
Beschreibung	Es kann ein absoluter passiver Encoder (SSI) eingesetzt werden. ENCODERFREQ setzt die Datenrate des SSI Encoders.
Portabilität	Parameter ist ab Version 6.7.05 (MACS4) verfügbar.
Parametergruppe	AXE Gebersystem
Einheit	kHz
Bereich	39000 - 5000000
Init-Wert	262000

MENCODERCLOCK - 77

Inhalt	Interne oder externe Takterzeugung für SSI Master-Encoder.
Beschreibung	Es kann ein absoluter passiver Master-Encoder (SSI) eingesetzt werden. Wenn ENCODERCLOCK auf 0 gesetzt ist, wird kein Takt erzeugt und der passive Modus ist aktiv.
Portabilität	Parameter ist ab Version 6.7.05 (MACS4) verfügbar.
Parametergruppe	AXE Gebersystem
Einheit	–
Bereich	0 = externe Takterzeugung für SSI Master-Encoder 1 = interne Takterzeugung für SSI Master-Encoder
Init-Wert	1

VELKPROP - 90

Inhalt	Proportionalfaktor der PI Drehzahlregelung.
Beschreibung	Der <i>Proportionalfaktor</i> VELKPROP der <i>Drehzahlregelung</i> gibt den linearen Korrekturfaktor an, mit dem die Abweichung zwischen der aktuellen Soll- und Istdrehzahl bewertet und eine entsprechende Korrektur der Motordrehzahl vorgenommen wird. !!! Dieser Parameter ist nur relevant, falls eine MACS-Steuerung mit integrierten Leistungsstufen verwendet wird und der Parameter AMPMODE auf eine Betriebsart mit aktiviertem Drehzahlregler gesetzt ist. Tipp: Zur schnellen und einfachen Konfiguration, ohne Berücksichtigung von Drehzahlregler-Parametern, kann mit dem Parameter AMPMODE eine Betriebsart ohne Drehzahlregelung ausgewählt werden. Für viele Anwendungen und Motortypen bietet eine solche Konfiguration ebenfalls eine ausreichend gute Reglerqualität und Positions- sowie Drehzahlgenauigkeit. Anmerkung: Der Parameter VELKPROP ist unabhängig von der Einstellung des Parameters ENCODER. Bei einer Änderung der Geberauflösung kann ein optimierter VELKPROP unverändert beibehalten werden. Es müssen bei Änderungen der Geberauflösung lediglich die Parameter KPROP, KDER, KINT des Positionsreglers angepasst werden.
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.4.63 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweise	VELKINT, VELKILIM, AMPMODE
Einheit	-
Bereich	-32767 ... 32767
Init-Wert	200

VELKINT - 91

Inhalt	Integralfaktor der PI Drehzahlregelung.
Beschreibung	<p>Der Integralfaktor VELKINT der Drehzahlregelung ist der Gewichtungsfaktor, mit dem im Zeitpunkt n die Summe aller Drehzahlfehler bewertet wird.</p> <p>Der Integralanteil der PI-Drehzahlregelung bewirkt ein entsprechend zeitlich anwachsendes, korrigierendes Motordrehmoment. Durch den Integralanteil wird ein statischer Drehzahlfehler zu Null ausgeregelt. Ein zu großer Integralfaktor führt jedoch zu einem „nervösen“ Antrieb.</p> <p>!!! Dieser Parameter ist nur relevant, wenn eine MACS-Steuerung mit integrierten Leistungsstufen verwendet wird und der Parameter AMPMODE auf eine Betriebsart mit aktiviertem Drehzahlregler gesetzt ist.</p> <p>Tipp: Zur schnellen und einfachen Konfiguration ohne Berücksichtigung von Drehzahlregler-Parametern, kann mit dem Parameter AMPMODE eine Betriebsart ohne Drehzahlregelung ausgewählt werden. Für viele Anwendungen und Motortypen bietet eine solche Konfiguration ebenfalls eine ausreichend gute Reglerqualität und Positions- sowie Drehzahlgenauigkeit.</p> <p>Anmerkung: Der Parameter VELKPROP ist unabhängig von der Einstellung des Parameters ENCODER. Bei einer Änderung der Geberauflösung kann ein optimierter VELKPROP unverändert beibehalten werden. Es müssen bei Änderungen der Geberauflösung lediglich die Parameter KPROP, KDER, KINT des Positionsreglers angepasst werden.</p>
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.4.63 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweise	VELKPROP, VELKILIM, AMPMODE
Einheit	-
Bereich	-32767 ... 32767
Init-Wert	5

CURKPROP - 92

Inhalt	Proportionalfaktor der PI Strom-/Drehmomentregelung
Beschreibung	<p>Der Proportionalfaktor CURKPROP der Strom-/Drehmomentregelung gibt den linearen Korrekturfaktor an, mit dem die Abweichung zwischen dem aktuellen Soll- und Ist-Motorstrom bewertet und eine entsprechende Korrektur des Motorstroms vorgenommen wird.</p> <p>!!! Dieser Parameter ist nur relevant, falls eine MACS-Steuerung mit integrierten Leistungsendstufen verwendet wird und der Parameter AMPMODE auf eine Betriebsart mit aktiviertem Strom-/Drehmomentregelung gesetzt ist.</p> <p>Tipp: Eine Grundeinstellung der Parameter des Strom-/Drehmomentreglers kann berechnet werden!</p> <p>Erforderliche Daten für die Berechnungsformel:</p> <ul style="list-style-type: none">- Beta: Dynamik der Regelung 0,1 ... 0,3 (0,1 = träge / 0,3 = schnell)- Ts: Abtastperiode [ms] 0,000125 (für MACS4-DC3 und MACS4-DC6)- U: Versorgungsspannung der Leistungsendstufe [V] 12V ... 48V- kc: Verstärkung der Leistungsendstufe 6554 für MACS4-DC6 3495 für MACS4-DC3- L: Induktivität der Motorwicklung [H] siehe Motordatenblatt. Einheit beachten!- R: Widerstand der Motorwicklung [Ohm]; siehe Motordatenblatt <p>Feste, MACS-interne Faktoren:</p> <ul style="list-style-type: none">- $ku = 2^{-12} * 2^{-15} = 0.000000007$ <p>Zwischenberechnete Werte:</p> <ul style="list-style-type: none">- $w = \text{Beta} * \pi / (2 * Ts)$ <p>Berechnungsformel:</p> <ul style="list-style-type: none">- $\text{CURKPROP} = ((2 * w) - (R / L)) * L / (U * kc * ku)$- $\text{CURKINT} = ((w^2 * L) / (kc * ku * U)) * (Ts / 4)$
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.4.63 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweise	CURKINT, CURKILIM, AMPMODE
Einheit	-
Bereich	-32767 ... 32767
Init-Wert	200

CURKINT - 93

Inhalt	Integralfaktor der PI Strom-/Drehmomentregelung.
Beschreibung	<p>Der Integralfaktor CURKINT der Strom-/Drehmomentregelung ist der Gewichtungsfaktor, mit dem im Zeitpunkt n die Summe aller Stromfehler bewertet wird.</p> <p>Der Integralanteil der PI-Stromregelung bewirkt einen entsprechend zeitlich anwachsenden, korrigierenden Motorstrom. Durch den Integralanteil wird ein statischer Stromfehler zu null ausgeglet. Ein zu großer Integralfaktor führt jedoch zu einem „nervösen“ Antrieb mit starken Stromschwankungen.</p> <p>!!! Dieser Parameter ist nur relevant, falls eine MACS-Steuerung mit integrierten Leistungsstufen verwendet wird und der Parameter AMPMODE auf eine Betriebsart mit aktiviertem Strom-/Drehmomentregelung gesetzt ist.</p> <p>Tipp: Eine Grundeinstellung der Parameter des Strom-/Drehmomentreglers kann berechnet werden!</p> <p>Erforderliche Daten für die Berechnungsformel:</p> <ul style="list-style-type: none">- Beta: Dynamik der Regelung 0,1 ... 0,3 (0,1 = träge / 0,3 = schnell)- Ts: Abtastperiode [ms] 0,000125 (für MACS4-DC3 und MACS4-DC6)- U: Versorgungsspannung der Leistungsstufe [V] 12V ... 48V- kc: Verstärkung der Leistungsstufe 6554 für MACS4-DC6 3495 für MACS4-DC3- L: Induktivität der Motorwicklung [H] siehe Motordatenblatt. Einheit beachten!- R: Widerstand der Motorwicklung [Ohm] siehe Motordatenblatt <p>Feste, MACS-interne Faktoren:</p> <ul style="list-style-type: none">- $ku = 2^{-12} * 2^{-15} = 0.000000007$ <p>Zwischenberechnete Werte:</p> <ul style="list-style-type: none">- $w = \text{Beta} * \pi / (2 * Ts)$ <p>Berechnungsformel:</p> <ul style="list-style-type: none">- $\text{CURKPROP} = ((2 * w) - (R / L)) * L / (U * kc * ku)$- $\text{CURKINT} = ((w^2 * L) / (kc * ku * U)) * (Ts / 4)$
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.4.63 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweise	CURKPROP, CURKILIM, AMPMODE
Einheit	-
Bereich	-32767 ... 32767
Init-Wert	100

AMPMAXCUR - 94

Inhalt	Begrenzung des maximal zulässigen Motorstroms bei MACS4-Steuerungen mit integrierten Leistungsendstufen.
Beschreibung	<p>Der Parameter AMPMAXCUR begrenzt den maximal zur Verfügung stehenden Motorstrom. Der Motorstrom kann den konfigurierten Wert nicht übersteigen.</p> <p>!!! Beim Einsatz von Servomotoren wird häufig bewusst ein Maximalstrom zugelassen, der deutlich größer als der eigentliche Motornennstrom ist. Dies ermöglicht Anwendungen mit einer hohen Dynamik in der Beschleunigungs- und Bremsphase. Es muss aber beachtet werden, dass der effektive Motorstrom (=RMS), der sich in Abhängigkeit vom Fahrprofil und von Zykluszeiten ergibt, den Motornennstrom nicht übersteigt. Liegt der effektive Motorstrom über dem Nennstrom, kann der Motor beschädigt werden oder komplett ausfallen! Es liegt in der Verantwortung des Anlagenbauers einen Motor auszuwählen oder eine Anlagenkonfiguration vorzunehmen, die den Anforderung entspricht und zu keinen Defekten einzelner Komponenten führt.</p> <p>Ein begrenzter Motorstrom führt zu einem begrenzten Motordrehmoment. Falls dieses aufgrund der Lastverhältnisse nicht mehr zur Einhaltung des programmierten Bewegungsablaufs ausreicht, treten so genannte Schleppfehler-Meldungen auf.</p> <p>Die Motorstrombegrenzung ist nur aktiviert, falls eine MACS-Steuerung mit integrierten Leistungsendstufen verwendet wird und der Parameter AMPMODE auf eine Betriebsart mit aktiviertem Strom-/Drehmomentregelung gesetzt ist.</p>
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.4.63 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweis	AMPMODE
Einheit	mA
Bereich	0 ... 60000 (Der Maximalwert hängt vom MACS4-Typ und der Endstufe ab.)
Init-Wert	2000

AMPCOMMTYPE - 95

Inhalt	Konfiguration des Motortyps, der an MACS-Steuerungen mit integrierten Leistungsendstufen angeschlossen ist.
Beschreibung	<p>Der Parameter AMPCOMMTYPE konfiguriert den Typ und die Kommutierung des Motors, der an die MACS-Steuerung mit integrierten Leistungsendstufen angeschlossen ist:</p> <p>0 = Kein Motor angeschlossen 1 = Bürstenbehafteter Motor 2 = Bürstenloser Motor mit 120° Hallsensoren 3 = Bürstenloser Motor mit 60° Hallsensoren 4 = Schrittmotor mit feldorientierter Ansteuerung 5 = Schrittmotor mit Mikroschritt-Ansteuerung 6 = Gekoppelter DC-Servobetrieb (2 Endstufen für einen bürstenbehafteten Motor)</p> <p>Die Verfügbarkeit und die Einsatzmöglichkeit der unterschiedlichen AMPCOMMTYPE-Werte hängt stark von dem eingesetzten Steuerungstyp ab: AMPCOMMTYPEs 0, 1, 2, 3 sind bei der MACS4-DC3 möglich AMPCOMMTYPEs 0 und 1 sind bei der MACS4-DC6 möglich Die Schrittmotorsteuerung sowie der gekoppelte DC-Servobetrieb sind derzeit OEM-Versionen der MACS-Steuerungen vorbehalten.</p>
Portabilität	Dieser Parameter ist nur für MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version ab 6.4.63 vorhanden.
Parametergruppe	AXA Verstärker
Querverweis	AMPMODE
Einheit	-
Bereich	0 ... 6
Init-Wert	0

AMPPWMFREQ - 96

Inhalt	PWM-Taktfrequenz von MACS4 Steuerungen mit integrierten Leistungsendstufen.
Beschreibung	<p>Der Parameter AMPPWMFREQ konfiguriert die Taktfrequenz der PWM-Endstufe der integrierten Leistungsendstufe.</p> <p>Eine höhere Frequenz verhindert ein Motorpfeifen im hörbaren Bereich und reduziert die Eigenerwärmung von Motoren mit niedriger Induktivität. Ein kleiner Nachteil höherer Frequenzen ist eine geringfügig höhere Eigenerwärmung der Endstufe aufgrund von Umschaltverlusten in den internen Treiberbausteinen.</p>
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.4.63 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweis	AMPMODE
Einheit	Hz
Bereich	<p>10000 ... 50000</p> <p>Die maximale PWM Frequenz und die einstellbaren Werte hängen von der eingesetzten MACS4 Hardware ab!</p> <p>Beispiel: MACS4-DC3 und MACS4-DC6</p> <p>Maximale PWM-Frequenz ist 48000 Hz. Zulässige Werte sind jeweils ein Vielfaches von 8000 Hz, also 16000, 24000, 32000, 40000 oder 48000 kHz. Falls hiervon abweichende Werte für den Parameter eingegeben werden, kann hieraus ein unbefriedigendes Reglerverhalten resultieren.</p>
Init-Wert	24000

AMPMODE - 97

Inhalt	Auswahl der Reglerstruktur/-kaskadierung, die bei MACS-Steuerungen mit integrierten Endstufen verwendet wird.
Beschreibung	<p>Der Parameter AMPMODE konfiguriert, welche Regler aktiviert sind und wie die Reglerstruktur bei MACS-Steuerungen mit integrierten Endstufen ausgeführt werden soll:</p> <p>0 = Positionsregler -> Drehzahlregler -> Stromregler -> PWM 1 = Positionsregler -> Drehzahlregler -> PWM 2 = Positionsregler -> Stromregler -> PWM 3 = Positionsregler -> PWM</p> <p>Wird in einer Anwendung nur eine Drehzahl- und/oder Strom-/Drehmomentregelung benötigt, muss der Parameter AMPMODE auf einen Typ gesetzt werden, der die benötigten Regler beinhaltet. Das Anwendungsprogramm kann dann über einen MOTOR OFF- und OUTAN-Befehl den Positionsregelkreis deaktivieren und den (über OUTAN definierten) Sollwert direkt zu dem Regelkreis unterhalb des Positionsreglers senden.</p> <p>Der MOTOR OFF-Befehl deaktiviert prinzipiell alle Regelkreise der entsprechenden Achse. Der OUTAN-Befehl aktiviert die Regelkreise entsprechend der AMPMODE Reglerstruktur unterhalb der Positionsregelung.</p> <p>Wenn AMPMODE auf 0 gesetzt ist und MOTOR OFF und OUTAN durch die Applikation ausgeführt wird, ... wird der Drehzahl- und der Strom-/Drehmomentregler aktiviert ... wird der OUTAN Wert als Sollwertvorgabe für den Drehzahlregler verwendet</p> <p>Wenn AMPMODE auf 1 gesetzt ist und MOTOR OFF und OUTAN durch die Applikation ausgeführt wird, ... wird nur der Drehzahlregler aktiviert ... wird der OUTAN Wert als Sollwertvorgabe für den Drehzahlregler verwendet</p> <p>Wenn AMPMODE auf 2 gesetzt ist und MOTOR OFF und OUTAN durch die Applikation ausgeführt wird, ... wird nur der Strom-/Drehmomentregler aktiviert ... wird der OUTAN Wert als Sollwertvorgabe für den Strom-/Drehmomentregler verwendet.</p> <p>!!! Es ist zu beachten, dass bei Reglerstrukturen ohne Stromregler (d.h. AMPMODE = 1 oder 3) keine Strombegrenzung (AMPMAXCUR) aktiviert ist. Die Strombegrenzung ist nur wirksam bei Reglerstrukturen mit aktiviertem Stromregler.</p> <p>Eine Änderung bei der Auswahl der Reglerstruktur kann ebenfalls Auswirkungen auf die Reglerqualität und -stabilität haben. In den meisten Fällen ist nach einer Änderung der Reglerstruktur auch eine erneute Optimierung der Parameter Positions- und gegebenenfalls des Drehzahlregelkreises notwendig.</p> <p>Der AMPMODE = 3 ist für die schnelle und einfache Inbetriebnahme eines Antriebssystems die geeignete Wahl. Bei dieser Reglerstruktur müssen lediglich die Parameter des Positionsregelkreises (KPROP, KDER, KINT) optimiert werden.</p>
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.4.63 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweise	KPROP, KDER, KINT, KILIM – VELKPROP, VELKINT, VELKILIM – CURKPROP, CURKINT, CURKILIM, AMPMAXCUR, AMPCOMMTYPE – MOTOR OFF, OUTAN
Einheit	-
Bereich	0 ... 3
Init-Wert	0

JERKMIN - 98, 100, 101, 102

Inhalt	JERKMIN	ist die Zeitspanne [ms], in der ... die definierte Maximalbeschleunigung erreicht werden soll, bzw.
	JERKMIN2	... die maximale Beschleunigung auf 0 Beschleunigung reduziert werden soll,
	JERKMIN3	... von 0 die maximale Verzögerung erreicht werden soll.
	JERKMIN4	... von der maximalen Verzögerung auf 0 Verzögerung reduziert werden soll.
Beschreibung	Es stehen vier verschiedene JERKMIN Varianten zur Verfügung, siehe „Ruckbegrenzung“ im Kapitel „Motion Control Praxis“.	
	Der bei Parameter RAMPTYPE (32) = 2 benutzte maximale Ruck wird durch JERKMIN mit folgender Formel berechnet:	
	Maximale Beschleunigung	= Maximale Geschwindigkeit / RAMPMIN
	Maximaler Ruck	= Maximale Beschleunigung / JERKMIN
	Beachten Sie, dass RAMPMIN und JERKMIN Zeitangaben in Millisekunden sind.	
	Kalkulationsbeispiel:	
	VELMAX	= 3000 (U/min)
	ENCODER	= 500 Pulse/Umdrehung
	RAMPMIN	= 500 ms
	JERKMIN	= 200 ms
	Das resultiert in:	
	VELMAX	= $3000 * 500 * 4 / 60 = 100.000 \text{ qc/s} = 100 \text{ qc/ms}$
	MaxAcc	= $200.000 \text{ qc/s}^2 = 0,2 \text{ qc/ms}^2$
	MaxJerk	= $1.000.000 \text{ qc/s}^3 = 0,001 \text{ qc/ms}^3$
	!!! Eine geänderte JERKMIN Einstellung wird nach dem nächsten Fahrbefehl (POSA, POSR, CVEL oder MOTOR STOP) ausgeführt und in der Beschleunigungsberechnung berücksichtigt. Im NOWAIT ON Modus ist es somit möglich, während einer Bewegung durch nochmaligen Aufruf des POSA Befehls den Bewegungsvorgang online (das heißt ohne Stopp) an neue Definitionen anzupassen.	
	!!! Wenn bei JERKMIN2, JERKMIN3 und/oder JERKMIN4 „0“ gesetzt ist, wird der gleiche Wert wie bei JERKMIN verwendet.	
Portabilität	Nur mit MACS3 und Firmware Version ab 6.4.63 sowie APOSS Entwicklungsoberfläche ab Version 6.7.40 mit der integrierten Compiler-Version 6.3.20.	
Parametergruppe	AXV Velocity	
Einheit	ms	
Bereich	0 ... MLONG	
Init-Wert	0	

VELKILIM - 103

Inhalt	Grenzwert für die Integralsumme der PI Drehzahlregelung.
Beschreibung	Das <i>Integrationslimit</i> VELKILIM der <i>Drehzahlregelung</i> begrenzt die Integralsumme, um zu verhindern, dass durch einen hohen aufaddierten Fehler eine zu starke Gegenregelung erfolgt, die das System zum Schwingen bringen würde.
!!!	Dieser Parameter ist nur relevant, wenn eine MACS-Steuerung mit integrierten Leistungsstufen verwendet wird und der Parameter AMPMODE auf eine Betriebsart mit aktiviertem Drehzahlregler gesetzt ist.
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.6.18 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweise	VELKPROP, VELKINT, AMPMODE
Einheit	-
Bereich	0 ... 32767
Init-Wert	1000

CURKILIM - 104

Inhalt	Grenzwert für die Integralsumme der PI Strom-/Drehmomentregelung.
Beschreibung	Das <i>Integrationslimit</i> CURKILIM der <i>Strom-/Drehmomentregelung</i> begrenzt die Integralsumme, um zu verhindern, dass durch einen hohen aufaddierten Fehler eine zu starke Gegenregelung erfolgt, die das System zum Schwingen bringen würde.
!!!	Dieser Parameter ist nur relevant, wenn eine MACS-Steuerung mit integrierten Leistungsstufen verwendet wird und der Parameter AMPMODE auf eine Betriebsart mit aktiviertem Strom-/Drehmomentregelung gesetzt ist.
Portabilität	Dieser Parameter ist nur bei MACS-Steuerungen mit integrierten Endstufen und einer Firmware-Version 6.6.18 oder höher vorhanden.
Parametergruppe	AXA Verstärker
Querverweise	CURKPROP, CURKINT, AMPMODE
Einheit	-
Bereich	0 ... 32767
Init-Wert	1000

KILIMTIME - 105

Inhalt	Zeitspanne, über welche der Integralgrenzwert des Positionsreglers auf den definierten KILIM-Wert erhöht bzw. auf null reduziert wird.
Beschreibung	<p>Der Integralanteil der PID Positionsregelung kann permanent aktiv sein oder nur während einer Fahrt oder nur im Stillstand. Der Wert des Parameters KILIMTIME legt das gewünschte Verhalten fest.</p> <p>Die Werkseinstellung 0 für KILIMTIME bedeutet, dass der Integralanteil der PID Positionsregelung permanent, entsprechend den Einstellungen von KINT und KILIM, aktiv ist.</p> <p>Ein Wert größer als 0 für KILIMTIME bedeutet, dass der Integralanteil der PID Positionsregelung nur während der Motorbewegung aktiv ist. Beim Motorstillstand wird der Integralanteil auf null reduziert. Beim Start der Bewegung wird das zulässige Integralmaximum innerhalb der durch KILIMTIME definierten Zeitspanne von 0 auf den definierten Grenzwert KILIM angehoben. Sobald der Motor wieder stoppt, wird das Integralmaximum innerhalb der Zeitspanne KILIMTIME auf null reduziert und der Integralanteil ist somit abgeschaltet. Ein solches Verhalten des Integralanteils erlaubt bei Synchronisationsanwendungen während der Bewegung einen sehr kleinen Synchronisationsfehler, bei gleichzeitig hoher Elastizität in der Ruhelage. Insbesondere bei Förderketten kann dies gewünscht sein.</p> <p>Ein Wert von kleiner als 0 für KILIMTIME bedeutet, dass der Integralanteil der PID Positionsregelung nur im Stillstand aktiviert ist. Der Absolutwert von KILIMTIME definiert hierbei die Zeitspanne, in der beim Motorstart das zulässige Integrationsmaximum auf null reduziert wird bzw. in der beim erneuten Stopp das zulässige Integralmaximum auf den konfigurierten Grenzwert KILIM angehoben wird. Dieses Verhalten des Integralanteils ist insbesondere hilfreich um ein nervöses Motorverhalten oder Aufschaukeln während der Bewegung zu verhindern und gleichzeitig eine sehr exakte Endpositionierung zu ermöglichen.</p>
Portabilität	MACS Firmware 6.6.71 oder höher
Parametergruppe	AXR - Parameter für den Positionsregler
Querverweise	KPROP, KDER, KINT, KILIM
Einheit	ms
Bereich	-32767 ... 32767
Init-Wert	0
	0 = Integralanteil ist permanent aktiv
	< 0 = Integralanteil ist nur bei Motorstillstand aktiv
	> 0 = Integralanteil ist nur während der Motorbewegung aktiv

SYNCSFTIME - 106

Inhalt	Filterzeit [ms] für die Slave Marker Korrektur.
Beschreibung	<p>Filterzeit für Slave-Encoder (wie SYNCMFTIME für Master).</p> <p>Wenn SYNCSFTIME gesetzt ist, wird die MMarkerDist ein wenig anders berechnet. Wenn die angesetzte Zeit zu einer Anzahl von Markern führt, die kleiner ist als 100, werden mindestens 100 genommen um den durchschnittlichen Markerabstand (Filter) zu berechnen. Dies betrifft nur die Berechnung des Markerabstands. Für die anderen Filterberechnungen (Abweichung) wird weiterhin die Anzahl der Marker verwendet, die durch SYNCSFTIME und aktueller Master-Geschwindigkeit berechnet wurden.</p> <p>Siehe auch Abbildung Marker Correction im Kapitel „Technische Referenz“.</p>
Portabilität	Der Parameter ist ab Firmware 6.6.18 verfügbar.
Parametergruppe	AXS
Querverweise	SYNCMFTIME
Einheit	ms
Bereich	0 ... MLONG 0 = keine Filterung
Init-Wert	0

ENCCONTROL - 107

Inhalt	Konfiguration, wie die Position nach einem Wechsel der Drehgeberquelle ausgewertet werden soll.
Beschreibung	<p>Bei einem „Soft Encoder Wechsel“ (Bit 0) wird, wenn der Encoder gewechselt wird, die Position des neuen Encoders ignoriert und es werden nur die relativen Änderungen zur Istposition hinzugefügt. Dies erlaubt den Wechsel zwischen verschiedenen Drehgebern (d.h. verschiedene Master) ohne Sprünge zu produzieren. Wenn diese Funktion nicht ausgewählt ist, wird beim Wechsel der Encoder der exakte Wert des neuen Encoders genommen.</p> <p>Soft Nullstellung (Bit 1) heißt, dass ein DEFORIGIN, SETORIGIN, DEFCORIGIN oder RSTORIGIN nicht wirklich die interne Drehgeberposition ändert, sondern es wird nur der virtuelle Encoder für diese Achse benutzt. Zum Beispiel können mit dieser Funktion zwei Achsen den gleichen Encoder benutzen (eine Achse als Slave-, die andere als Master-Encoder). Wenn diese Funktion aktiviert ist, können die ersten Achsen ein DEFORIGIN für den Slave-Encoder ausführen, ohne einen Sprung für den Master von Achse 2 zu erzeugen. Die tatsächliche Drehgeberposition bleibt unverändert.</p> <p>Beispiel:</p> <pre>SET ENCODERTYPE 10 / 20 // Achse 1 benutzt den ersten Encoder als Slave, // Achse 2 benutzt den zweiten Encoder SET MENCODERTYPE 30 / 10 // Achse 2 Master ist aktuell der Encoder von Achse 1 SET ENCCONTROL 2 // Soft Nullstellung für Achse 1 Slave-Encoder PRINT "apos x1 ",apos x(1), " mapos x(2) ",mapos x(2), "encoder(2) ",sysvar[42] // PRINT wird etwa so aussehen // " apos x(1) 25000 mapos x(2) 25000 encoder(2) 25000 DEFORIGIN x(1) PRINT "apos x1 ",apos x(1), " mapos x(2) ",mapos x(2), "encoder(2) ",sysvar[42] // Print sieht etwa so aus " apos x(1) 0 mapos x(2) 25000 encoder(2) 25000</pre> <p>Wenn MENCCONTROL von x(2) auf 2 gesetzt war, dann könnte auch ein DEFMORIGIN x(2) ohne Probleme ausgeführt werden.</p>
Portabilität	Parameter ab Firmware 6.6.18.
Parametergruppe	AXE Gebersystem
Einheit	–
Bereich	0 = Direkt zu den Positionsdaten der neuen Quelle wechseln. Bit 0 = 1 = Soft-Wechsel, d.h. Position der alten Quelle + relative Daten der neuen Quelle Bit 1 = 1 = Soft-Nullstellung (DEFORIGIN ... setzt den Encoder nicht zurück)
Init-Wert	0

MENCCONTROL - 108

Inhalt	Konfiguration, wie die Master-Position nach einem Wechsel der Drehgeberquelle aufgeholt werden soll.
Beschreibung	<p>Ein „Soft-Encoder-Wechsel“ eignet sich besonders, wenn Encoder während der Fahrt gewechselt werden sollen. Wenn so ein Wechsel ohne diesen Parameter durchgeführt wird, würde die Einstellung des neuen Encoders typischerweise einen Schleppfehler verursachen, weil die Drehgeberwerte nicht die gleichen sind. Wenn dagegen ein Soft-Encoder-Wechsel gewählt wird, wird nicht vollständig auf den Wert des neuen Encoders gewechselt, sondern der alte Wert behalten und es werden nur die Unterschiede des neuen Encoders addiert. Daher können die Encoder „während der Fahrt“ gewechselt werden.</p> <p>Eine Soft Nullstellung eignet sich zum Beispiel besonders dann, wenn der Encoder von einer weiteren Achse für einen anderen Zweck benutzt wird und Sie den Drehgeberwert bei einem DEFMORIGIN nicht tatsächlich wechseln wollen. Wenn die Soft Nullstellung eingestellt ist, dann können Sie ein DEFMORIGIN benutzen und die neue zurückgemeldete Istposition ist danach null. Tatsächlich hat der Encoder aber immer noch seinen alten Wert (d.h. der Wert den Sie mit SYSVAR 0x0122022A oder ähnlich auslesen können).</p>
Portabilität	Ab Firmware 6.6.18
Parametergruppe	AXE
Einheit	–
Bereich	0 = Direkt zu den Positionsdaten der neuen Quelle wechseln. Bit 0 =1 = Soft-Wechsel, d.h. Position der alten Quelle + relative Daten der neuen Quelle Bit 1 =1 = Soft-Nullstellung
Init-Wert	0

SYNCFFVEL - 109

Inhalt	Geschwindigkeits-Feedforward [per mill von VCMD] für den Synchronisations-Modus.
Beschreibung	<p>SYNCFFVEL unterstützt geschwindigkeitsabhängiges Feedforward in Synchronmodi (SYNCP / SYNCM / SYNCC). Dieser Parameter ist entweder 0 (deaktiviert = Default) oder hat den Wert, der den Feedforward in 1/1000 der Sollgeschwindigkeit angibt. Das heißt, ein Wert von 1000 addiert einen Feedforward von der VCMD (Sollgeschwindigkeit) zur MPCMD (Sollposition der Synchronisation). Um den Wert der benutzt werden soll zu bestimmen, kann die SYSVAR NORMTRACKERR benutzt werden. Wenn Sie herausfinden, dass der normale NORMTRACKERR 100 Prozent ist, dann sollten Sie die SYNCFFVEL auf 1000 setzen.</p> <p>Dadurch kann SYNCERR minimiert werden. Mit so einem Feedforward kann SYNCERR fast ausgeschlossen werden, ohne die Nachteile durch Nutzen eines Interrupt-Teils im PID.</p> <p>Zurzeit gibt es noch den Nachteil eines Rucks, wenn Sie SYNCP während der Fahrt starten. Dies tritt auf, wenn die Geschwindigkeit durch ein SYNCC oder ein CVEL erreicht ist und dann SYNCP gestartet wird.</p> <p>Neue SYSVAR REG_NORMTRACKERR (4124):</p> <p>Diese SYSVAR liefert den Fahrwegfehler im Verhältnis zur Sollgeschwindigkeit in Prozent. Zum Beispiel sagt ein Wert von 120 aus, dass der Fahrwegfehler 1,2-mal VCMD ist. Dieser Wert ist relativ konstant, wenn sich die Bedingungen nicht ändern (Last, Reibung, ...) und typischerweise unabhängig von der Geschwindigkeit.</p>
Portabilität	Ab Firmware 6.6.24
Parametergruppe	AXS Synchronisation
Querverweise	SYNCP, SYNCM, SYNCC, SYSVAR 4124 siehe SDO dictionary, axis process data
Einheit	–
Bereich	0 ... MLONG 0 = ausgeschaltet
Init-Wert	0

SYNCVFLIMIT - 110

Inhalt	Synchronisationsfehler-Fenster [qc] für die automatische Deaktivierung von SYNCVFTIME.
Beschreibung	<p>SYNVFTIME wird nicht aktiviert, wenn SYNCERR den Wert erreicht, der durch SYNCVFLIMIT definiert ist. SYNCVFTIME wird wieder aktiviert, wenn SYNCERR kleiner als 1/5 des Wertes von SYNCVFLIMIT wird.</p> <p>Dieser Parameter hilft große Synchronisationsfehler zu vermeiden, wenn der Master beschleunigt oder bremst und ein hoher Wert für SYNCVFTIME benutzt wird.</p> <p>Dieser Parameter erlaubt das Ausschalten des SYNCVFTIME Filters, wenn der Fehler größer als SYNCVFLIMIT wird. Sobald der Filterfehler (PFG_MFILTERERROR) den Wert von SYNCVFLIMIT [qc] erreicht, wird der Filter langsam deaktiviert. Wenn der Fehler kleiner wird als SYNCVFLIMIT/5, wird der Filter wieder aktiviert. Bei SYNCM wird der interne Filter bei jedem SYNCVFTIME auf null zurückgesetzt, weil er nur ansteigt und nie abnimmt. In diesem Fall wird nach SYNCVFTIME wieder geprüft ob der Fehler klein genug ist, um den Filter wieder zu aktivieren.</p> <p>Der Filter wird nicht sofort deaktiviert oder aktiviert, sondern er wird langsam 1 ms pro ms erhöht oder verringert. Und er wird nicht vollständig deaktiviert, sondern es werden mindestens 5 ms übrig gelassen. Bei SYNCP wird der ursprüngliche Filterwert noch für die Berechnung des aktuellen Filterfehlers (PFG_MFILTERERROR) benutzt, um zu entscheiden, ob der Filter wieder aktiviert werden sollte. Bei SYNCM (keine Filterkorrektur) wird der Filterfehler auf null zurückgesetzt und mindestens SYNCVFTIME gewartet, um zu entscheiden, ob der Filterfehler unter dem Limit bleibt oder nicht. Falls er unter dem Limit bleibt wird der Filter wieder aktiviert.</p> <p>Der Filterfehler kann mit der SYSVAR PFG_MFILTERERROR beobachtet werden.</p>
Portabilität	Ab Firmware 6.6.24
Parametergruppe	AXS Synchronisation
Querverweise	SYNVFTIME,
Einheit	qc
Bereich	0 ... MLONG
Init-Wert	0

POSERRTIME - 111

Inhalt	Zeitfenster [ms] für das Auslösen eines Positionsfehlerstatus.
Beschreibung	<p>Zu große Schleppfehler (POSERR) lösen den Fehlerstatus aus, wenn sie länger existieren als POSERRTIME.</p> <p>Der Default dieses Parameters ist 0. Wenn dieser Parameter nicht 0 ist, wird ein Schleppfehler nur dann erzeugt, wenn der Wert des Schleppfehlers (POSERR) für eine Zeit länger als POSERRTIME erreicht wird. Intern wird der Schleppfehler alle 20 ms für eine Achse geprüft. Das heißt, das Erkennen eines Schleppfehlers kann im schlimmsten Fall bis zu $\text{NoOfAxes} * 20$ ms dauern. Zum Beispiel bei 2 Achsen und einer POSERRTIM von 300 ms wird der Schleppfehler mehr als $(300 / 2 * 20) = 7$ Mal geprüft.</p> <p>Wenn das erfüllt ist, wird ein Fehler ausgegeben.</p>
Portabilität	Ab Firmware 6.632
Parametergruppe	AXR Pos. Regelung
Querverweise	POSERR
Einheit	ms
Bereich	0 ... 10000
	0 = es ist kein Zeitfenster aktiviert, d.h. der Fehlerstatus wird sofort ausgelöst.
Init-Wert	0

FEEDDIST - 112

Inhalt	Faktor Benutzereinheit um CANopen-kompatible Benutzereinheiten zu erhalten.
Beschreibung	<p>Andere Faktoren: FEEDREV, POSENCOC, POSENCREV. Die Defaultwerte sind abwärtskompatibel.</p> <p>Formel zur Berechnung der Benutzereinheit:</p> $BE = \frac{FEEDREV * POSFACT_Z * POSENCOC}{FEEDDIST * POSFACT_N * POSENCREV} = qc$ <p>(Die Parameter POSFACT_Z und POSFACT_N entsprechen dem CANopen DS402 Objekt 0x6091 Getriebeübersetzung.)</p>
Portabilität	Ab Firmware 6.6.51.
Parametergruppe	AXE (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweise	FEEDREV, POSENCOC, POSENCREV, POSFACT_Z, POSFACT_N
Einheit	–
Bereich	1 ... MLONG
Init-Wert	1

FEEDREV - 113

Inhalt	Berechnung der Benutzereinheit [qc] für die CANopen-Kompatibilität.
Beschreibung	Formel zur Berechnung der Benutzereinheit: $BE * \frac{FEEDREV * POSFACT_Z * POSENCQC}{FEEDDIST * POSFACT_N * POSENCREV} = qc$ <p>(Die Parameter POSFACT_Z und POSFACT_N entsprechen dem CANopen DS402 Objekt 0x6091 Getriebeübersetzung)</p>
Portabilität	Ab Firmware 6.6.51.
Parametergruppe	AXE (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweise	POSFACT_Z, POSFACT_N
Einheit	qc
Bereich	1 ... MLONG
Init-Wert	1

POSENCQC - 114

Inhalt	Faktor Benutzereinheit für die CANopen-Kompatibilität
Beschreibung	Formel zur Berechnung der Benutzereinheit: $BE = \frac{FEEDREV * POSFACT_Z * POSENCQC}{FEEDDIST * POSFACT_N * POSENCREV} = qC$ (Die Parameter POSFACT_Z und POSFACT_N entsprechen dem CANopen DS402 Objekt 0x6091 Getriebeübersetzung)
Portabilität	Ab Firmware 6.6.51.
Parametergruppe	AXE (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweise	FEEDREV, POSENCQC, POSENCREV, POSFACT_Z, POSFACT_N
Einheit	–
Bereich	1 ... MLONG
Init-Wert	1

POSENCREV - 115

Inhalt	Faktor Benutzereinheit für die CANopen-Kompatibilität
Beschreibung	Formel zur Berechnung der Benutzereinheit: $BE \cdot \frac{FEEDREV \cdot POSFACT_Z \cdot POSENCOC}{FEEDDIST \cdot POSFACT_N \cdot POSENCREV} = qC$ <p>(Die Parameter POSFACT_Z und POSFACT_N entsprechen dem CANopen DS402 Objekt 0x6091 Getriebeübersetzung)</p>
Portabilität	Ab Firmware 6.6.51.
Parametergruppe	AXE (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweise	FEEDREV, POSENCOC, POSENCREV, POSFACT_Z, POSFACT_N
Einheit	–
Bereich	1 ... MLONG
Init-Wert	1

HOMEZEROVEL - 116

Inhalt	Skalierte Geschwindigkeit zum Fahren auf den Drehgeber-Index.
Beschreibung	<p>Geschwindigkeitsteiler (cp. VELRES), um auf den Drehgeber-Index zu fahren, als Teil des Befehls HOME oder bei CANopen Homefahrt-Prozeduren.</p> <p>Zurzeit wird der Parameter HOMEZEROVEL von der Firmware nicht benutzt. Er wird lediglich für die Anwendung CANopenStateMachine benötigt. Dort hat der Benutzer die Möglichkeit, die Geschwindigkeit für die Suche des Indexpulses ebenfalls vorzugeben (beim HOME Befehl wird immer ein fester Bruchteil von HOMELEVEL genommen). Der Benutzer setzt diesen Wert typischerweise mit SDO 0x6099 Subindex 2 in Benutzereinheiten per Sekunde. Dieser Wert wird dann automatisch umgerechnet und in VELRES im Parameter HOMEZEROVEL gespeichert.</p>
Portabilität	Ab Firmware 6.6.51.
Parametergruppe	AXH (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweise	VELRES
Einheit	typ. % (VELRES)
Bereich	1 ... MLONG
Init-Wert	10

I2TFTIME - 117

Kurzinfo Filterzeit in ms, um den I² gefilterten Wert zu bilden.

Beschreibung Alle Steuerung mit integrierten Verstärkern unterstützen auch eine Art von I²T Begrenzung. Dafür wurden die folgenden Parameter implementiert:

I2TLIMIT definiert die Begrenzung. Wird die Begrenzung überschritten, wird der Fehler 99 (F_I2TLIMIT) ausgegeben.

I2TFTIME definiert die Filterzeit in ms, die benutzt wird um den gefilterten I² Wert zu bilden.

CURFTIME definiert die Filterzeit für die Strommessung. Dies beeinflusst nicht die Stromregelung, der gelieferte Wert wird nur für die oben genannte I²T Begrenzung und von SYSVAR (siehe unten) benutzt.

AMP_I2TVALUE ist eine neue SYSVAR [0x2700 / 27] die den gefilterten Wert liefert, der für die Begrenzung benutzt wird.

Intern wird jede ms der Stromwert in mA ausgelesen und I² gebildet. Danach wird dieser Wert in 1/1000 A² skaliert. Der Wert wird dann in ein PT-Filter geschrieben und das Ergebnis wird für den Vergleich mit I2TLIMIT benutzt.

Dieser Wert gibt eine Vorstellung wie viel Energie in den Motor gesteckt wurde und dies natürlich ist proportional zur Energie, die den Motor aufheizt.

!!! Wenn dieser Parameter auf einen höheren Wert als Null gesetzt ist, wird automatisch die Überwachung der Begrenzung aktiviert; siehe I2LIMIT.

Portabilität Der Parameter ist ab Firmware 6.6.19 verfügbar.

Parametergruppe AXA (wird nicht im Parameter-Dialogfeld dargestellt)

Querverweise I2LIMIT, CURFTIME

Einheit ms

Bereich 0 ... MLONG

Init-Wert 0

I2TLIMIT - 118

Kurzinfo	Begrenzung des Parameters I2TVALUE (siehe 0x2700 Subindex 27).
Beschreibung	<p>Alle Steuerung mit integrierten Verstärkern unterstützen auch eine Art von I²T Begrenzung. Dafür wurden die folgenden Parameter implementiert:</p> <p>I2TLIMIT definiert die Begrenzung. Wird die Begrenzung überschritten, wird der Fehler 99 (F_I2TLIMIT) ausgegeben.</p> <p>I2TFTIME definiert die Filterzeit in ms, die benutzt wird um den gefilterten I² Wert zu bilden.</p> <p>CURFTIME definiert die Filterzeit für die Strommessung. Dies beeinflusst nicht die Stromregelung, der gelieferte Wert wird nur für die oben genannte I²T Begrenzung und von SYSVAR (siehe unten) benutzt.</p> <p>AMP_I2TVALUE ist eine neue SYSVAR [0x2700 / 27] die den gefilterten Wert liefert, der für die Begrenzung benutzt wird.</p> <p>Intern wird jede ms der Stromwert in mA ausgelesen und I² gebildet. Danach wird dieser Wert in 1/1000 A² skaliert. Der Wert wird dann in ein PT-Filter geschrieben und das Ergebnis wird für den Vergleich mit I2TLIMIT benutzt.</p> <p>Dieser Wert gibt eine Vorstellung wie viel Energie in den Motor gesteckt wurde und dies natürlich ist proportional zur Energie, die den Motor aufheizt.</p> <p>Beispiel:</p> <p>Angenommen, der Nennstrom eines Motors ist 2,5 Amps. Wenn der Motor nun mit Nennlast über eine längere Zeit fährt (etwa 3-mal länger als I2TFTIME), dann wird der ausgelesene Wert von AMP_I2TVALUE etwa 6250 sein, berechnet als 2,5 * 2,5 * 1000.</p> <p>So bietet dieser Wert einen Anhaltspunkt um zu beurteilen, ob der Motor überlastet werden könnte oder nicht.</p> <p>Wenn Sie diese Begrenzung zum Ausschalten des Motors benutzen, garantiert dies natürlich nicht, dass der Motor tatsächlich geschützt ist. Das Überhitzen des Motors hängt von vielen anderen Faktoren ab und kann nicht wirklich mit einer einfachen I²T Messung berechnet werden.</p>
Portabilität	Ab Firmware 6.6.19.
Parametergruppe	AXA (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweise	I2TVALUE siehe SDO object dictionary, I2TFTIME, Fehler_99
Einheit	A ² (1/1000)
Bereich	0 ... MLONG
Init-Wert	0

CURFTIME - 119

Kurzinfo	Filterzeit für die Strommessung.
Beschreibung	Die Filterzeit für die Strommessung beeinflusst nicht die Stromregelung, sondern nur den gelieferten Wert der für die I ² T Begrenzung und von der SYSVAR AMP_GETCUR benutzt wird.
Portabilität	Ab Firmware 6.6.19.
Parametergruppe	(wird nicht im Parameter-Dialogfeld dargestellt)
Querverweis	SDO object dictionary
Einheit	ms
Bereich	
Init-Wert	0

AMPENCNO - 126

Kurzinfo	Definition des Encoder-Eingangs, der als Rückführung für die Geschwindigkeitsregelung verwendet wird.
Beschreibung	<p>Die Rückführung für die Geschwindigkeitsregelung kann getrennt von der Rückführung der Positionsregelung festgelegt werden. Über insgesamt 3 Parameter wird die Rückführung des Geschwindigkeitsreglers konfiguriert:</p> <p>AMPENCNO Definition des Encoder-Eingangs</p> <p>AMPENCRES Auflösung des Encoders [qc/Umdr.]</p> <p>AMPENCRPM Maximale Geschwindigkeit</p> <p>Standardgemäß ist AMPENCNO auf -1 gesetzt und damit verhält sich das System wie bisher: Der Feedback-Encoder, welcher für die Positionsregelung verwendet wird, wird auch für die Geschwindigkeitsregelung benutzt.</p> <p>Wird AMPENCNO auf 1 gesetzt, wird der Encoder 1 als Rückführung verwendet; ein Wert von 2 verwendet Encoder 2, usw. Mit dem Parameter AMPENCRES muss die Auflösung des benutzten Encoders sowie mit AMPENCRPM die Maximalgeschwindigkeit in U/Min gesetzt werden. Der Defaultwert 0 von AMPENCRPM bedeutet dabei, dass für die Geschwindigkeitsregelung die identische Maximalgeschwindigkeit wie bei VELMAX verwendet wird.</p> <p>Mit dem Parameter POSDRCT kann die Art der Auswertung (normal oder invertiert) für die Geschwindigkeits- und Positionsrückführung getrennt festgelegt werden.</p>
Portabilität	Parameter ab Compiler Version 6.5.36.
Parametergruppe	AXB (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweis	AMPENCRES , AMPENCRPM , POSDRCT
Einheit	–
Bereich	<p>-1 wie bisher, d.h. die Feedback-Quelle der Positionsregelung wird auch für die Geschwindigkeitsregelung benutzt.</p> <p>1 Encoder 1 wird als Feedback-Quelle für die Geschwindigkeitsregelung verwendet.</p> <p>2 Encoder 2 wird als Feedback-Quelle für die Geschwindigkeitsregelung verwendet.</p> <p>usw.</p>
Init-Wert	-1

AMPENCRES - 127

Kurzinfo	Auflösung des Encoders [qc/Umdr.], der für die Geschwindigkeitsregelung verwendet wird.
Beschreibung	Falls mit AMPENCNO (> -1) die Rückführung für die Geschwindigkeits- und Positionsregelung getrennt gesetzt wurde, muss mit AMPENCRES ebenfalls die Auflösung des Encoders definiert werden. Der Wert von AMPENCRES muss (im Gegensatz zum Parameter ENCODER) in der Einheit Quadcounts/Umdrehung vorgegeben werden!
Portabilität	Parameter ab Compiler Version 6.5.36.
Parametergruppe	AXB (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweise	AMPENCNO , AMPENCRPM , POSDRCT
Einheit	qc/Umdrehung
Bereich	1 ... MLONG
Init-Wert	-1

AMPENCRPM - 128

Kurzinfo	Maximale Drehzahl [Umdr./Min] des Encoders, der für die Geschwindigkeitsregelung verwendet wird.
Beschreibung	<p>Wenn mit AMPENCNO (>-1) die Rückführung für die Geschwindigkeits- und Positionsregelung getrennt gesetzt wurde, muss mit AMPENCRPM die maximale Geschwindigkeit in [U/Min] definiert werden. Sie muss mit MAXVEL der Positionsregelung abgestimmt sein.</p> <p>Falls AMPENCRPM auf dem Defaultwert 0 gesetzt ist, wird automatisch der Wert von VELMAX ebenfalls für den Geschwindigkeitsregelkreis benutzt.</p> <p>Die Maximaldrehzahl des Geschwindigkeitsreglers korrespondiert mit der maximalen Ansteuerung (= 100% PWM Duty Cycle) der Endstufe und bestimmt somit die effektive erreichbare Maximaldrehzahl in Abhängigkeit von der Versorgungsspannung der Endstufe und der Motorcharakteristik. Der Parameterwert von AMPENCRPM (und VELMAX) muss für eine korrekte Funktion deshalb auf diese Größen abgestimmt sein.</p>
Portabilität	Parameter ab Compiler Version 6.5.36.
Parametergruppe	AXB (wird nicht im Parameter-Dialogfeld dargestellt)
Querverweis	AMPENCNO , AMPENCRES , POSDRCT
Einheit	Umdr./Min
Bereich	1 ... 65535
Init-Wert	0

Globale Parameter im Detail

CANNR - 100

Inhalt	CAN-Teilnehmernummer
Beschreibung	<p>Wenn eine CAN-Baugruppe vorhanden ist, legt der Parameter CANNR die Teilnehmernummer für die Steuerung fest.</p> <p>Die CAN-Nummer wird bei der M1 automatisch aus der mit dem Hex-Drehschalter vorgenommenen Einstellung generiert; bei allen anderen Steuerungen wird die CAN-Nummer bei der Einstellung der Schnittstelle definiert.</p> <p>Falls die CANNR auf 9999 gesetzt ist, werden keine Standard-CAN-Objekte erzeugt. Standard-CAN-Objekte werden zur Kommunikation mit dem APOSS-Programm bei den Befehlen OUTMSG, INMSG und INGLB benötigt.</p>
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2
Parametergruppe	GL globale Parameter
Einheit	–
Bereich	0 ... 2048
Init-Wert	0

CANBAUD - 101

Inhalt	CAN Baudrate
Beschreibung	<p>CANBAUD definiert die Baudrate für CAN Kommunikation (MACS3 für Master- und Slave-Bus). Die Baudrate kann mit HEX-Schalter (nur MACS3) oder mit APOSS-Befehlen gesetzt werden. Folgen Sie diesen Anweisungen, wenn Sie den HEX-Schalter benutzen (oder schlagen Sie im Hardware-Handbuch nach):</p> <ul style="list-style-type: none">– HEX-Schalter auf 00 stellen– Einschalten– 7-Segment-Anzeige zeigt „CAN“– Innerhalb von 30 Sekunden müssen Sie die Baudrate für den Master mit dem HEX-Schalter (H) und für den Slave mit dem HEX-Schalter (L) einstellen:<ul style="list-style-type: none">1 = 10 kBaud2 = 20 kBaud3 = usw. siehe unten <p>Wenn beide auf 0 stehen bleiben, wird 125 kBaud für beide ausgewählt.</p> <p>Nach 30 s ist die Baudrate akzeptiert und eingestellt und die Anzeige zeigt „Id“. Sie haben weitere 30 s Zeit, nun die CAN-ID für den Master-Bus zu wählen. Sie können dies auch später ändern und jetzt Aus- und Wiedereinschalten.</p> <p>Natürlich kann die Baudrate auch mit dem APOSS Befehl SET und dem Parameter CANBAUD gesetzt werden. Dieser Parameter hat folgenden Bereich:</p> $n = \text{MasterBaud} * 10 + \text{SlaveBaud}$ $\text{MasterBaud} = 0 \dots 8 \text{ (Baudrate 5 kBaud} \dots \text{ 1 MBaud)}$ $\text{SlaveBaud} = 0 \dots 8 \text{ (Baudrate 5 kBaud} \dots \text{ 1 MBaud)}$
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS2; Master und Slave nur mit MACS3.
Parametergruppe	GL globale Parameter
Einheit	–
Bereich	0 ... 8 <ul style="list-style-type: none">1 = 10 kBaud2 = 20 kBaud3 = 50 kBaud4 = 100 kBaud5 = 125 kBaud6 = 250 kBaud7 = 500 kBaud8 = 1000 kBaud
Init-Wert	5 = Standard CANopen-Module oder MACS2 55 = MACS3

PRGPAR - 102

Inhalt	Aktivierte Programmnummer
Beschreibung	<p>Mit dem PRGPAR können Sie festlegen, welches Programm nach Ablauf eines per Autostart (Autokennung) ausgeführten Programms gestartet werden soll. Dieser Parameter kann auch von Programmen oder per Display geändert und gespeichert werden.</p> <p>Wenn Sie innerhalb eines Programms mit dem Befehl SET, diesen Programmparameter setzen, z.B. SET PRGPAR 5 wird nach Ablauf des Programms das Programm Nr. 5 gestartet. So kann man von einem APOSS-Programm aus andere Programme starten und auch mehrere Programme miteinander verketteten.</p> <p>Wenn keine Programmnummer aktiviert ist und auch kein Eingang für Programmstart I_PRGSTART (103) gesetzt ist, wird wieder das mit Autokennung versehene Programm gestartet.</p> <p>!!! Wenn kein Autostart-Programm definiert ist, kann auch kein Programm über PRGPAR (102) gestartet werden, denn dies erfordert immer ein beendetes Autostart-Programm.</p>
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS2;Parameterwert +1000 mit Controller Version 6.1.14 aufwärts.
Parametergruppe	GL globale Parameter
Einheit	–
Bereich	–1 ... 127 (optional + 1000)
	–1 = Programmnummer ist nicht aktiviert, d. h. es wird nach AutoExec kein Programm gestartet.
	0...127 = Aktivierte Programmnummer (und AutoExec) werden nach dem Einschalten gestartet.
	0...127 +1000 = Aktivierte Programmnummer wird gestartet, aber beim Einschalten ist der Motor stromlos (MOTOR OFF).
	–1 + 1000 = (999) wie –1, aber der Motor ist nach dem (Strom-)Einschalten noch ausgeschaltet.
Init-Wert	–1
Beispiel	<pre>SET PRGPAR 5 // Programmnummer 5 wird sofort nach dem Einschalten // und nach dem Ausführen der AutoExec gestartet SET PRGPAR 1005 // Programmnummer 5 wird gestartet, // aber beim Hochfahren ist der Motor aus</pre>

I_PRGSTART - 103

Inhalt	Definiert den Eingang für die Art des Programmstarts
Beschreibung	<p>Wenn der Eingang für I_PRGSTART \neq 0 ist, wird zuerst das Autostart-Programm ausgeführt und danach gewartet, bis Eingang I_PRGSTART aktiv ist. Dieser wird entsprechend der Programmwahl I_PRGCHOICE (104) ausgewertet, um die Nummer des Programms zu bestimmen, das ausgeführt werden soll.</p> <p>Wenn kein Eingang für Programmstart gesetzt ist, wird das mit Autokennung versehene Programm gestartet.</p>
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS2
Parametergruppe	GL globale Parameter
Einheit	–
Bereich	0...8
Init-Wert	0

I_PRGCHOICE - 104

Inhalt Definiert den Eingang für die Programmwahl.

Beschreibung Wenn I_PRGCHOICE > 0 gesetzt wird, gibt dieser Parameter die Eingangsnummer an, ab der die Eingänge für die **Programmwahl** verwendet werden. Dazu gehören alle bis zu I_PRGSTART (103)

Beispiel Wenn I_PRGCHOICE = 3 und I_PRGSTART = 7, werden bei der Aktivierung von Eingang 7 die Eingänge 3, 4, 5, 6 binär ausgewertet und das Ergebnis als Programmnummer verwendet.

<u>Eingang</u>	<u>Pegel</u>	<u>Binärwert</u>
3	low	0
4	high	2
5	high	2 ²
6	low	0

=> zu startendes Programm: 6

Maximal kann somit zwischen 90 Programmen, die mit den Nummern 0 bis 89 gekennzeichnet sind, ausgewählt werden.

Portabilität Nur bei 32-Bit Steuerungen, wie M1 und MACS

Parametergruppe GL globale Parameter

Einheit –

Bereich 0 ... 8

Init-Wert 0

I_BREAK - 105

Inhalt	Eingang für Abbruch	
Beschreibung	I_BREAK definiert den Eingang, der bei Aktivierung sofort das laufende Programm abbricht. Mit CONTINUE kann ein solches Programm wieder fortgesetzt werden. Dabei kann man mit einer positiven oder negativen Zahl festlegen, ob auf eine positive oder negative Flanke reagiert werden soll.	
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.	
Parametergruppe	GL globale Parameter	
Einheit	–	
Bereich	1 ... 8	= Reaktion auf positive Flanke an Eingang 1 ... 8
	0	= keine Funktion
	-1 ... -8	= Reaktion auf negative Flanke an Eingang 1 ... 8
Init-Wert	0	

I_CONTINUE - 106

Inhalt	Programm fortsetzen	
Beschreibung	I_CONTINUE legt fest, welcher Eingang zur Fortsetzung von abgebrochenen Programmen verwendet wird. Dabei kann man mit einer positiven oder negativen Zahl festlegen, ob auf eine positive oder negative Flanke reagiert werden soll.	
Portabilität	Nur bei 32-Bit Steuerungen, wie M1 und MACS.	
Parametergruppe	GL globale Parameter	
Einheit	–	
Bereich	1 ... 8	= Reaktion auf positive Flanke an Eingang 1 ... 8
	0	= keine Funktion
	-1 ... -8	= Reaktion auf negative Flanke an Eingang 1 ... 8
Init-Wert	0	

I_ERRCLR - 107

Inhalt	Fehler löschen	
Beschreibung	I_ERRCLR legt fest, welcher Eingang zum Fehler löschen verwendet wird. Dabei kann man mit einer positiven oder negativen Zahl festlegen, ob auf eine positive oder negative Flanke reagiert werden soll.	
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.	
Parametergruppe	GL globale Parameter	
Einheit	–	
Bereich	1 ... 8	= Reaktion auf positive Flanke an Eingang 1 ... 8
	0	= keine Funktion
	-1 ... -8	= Reaktion auf negative Flanke an Eingang 1 ... 8
Init-Wert	0	

O_ERROR - 108

Inhalt	Ausgang für Fehler
Beschreibung	Der mit O_ERROR definierte Ausgang wird gesetzt, wenn ein Fehler aufgetreten ist. Sobald der Fehler gelöscht ist, wird dieser Ausgang zurückgesetzt. Eine positive Zahl bedeutet, dass der Ausgang high (24 Volt) ist, wenn die Funktion aktiv ist. Eine negative Zahl bedeutet, dass der Ausgang low (0 Volt) ist, wenn die Funktion aktiv ist.
!!!	Die Einstellung des Parameters O_ERROR hat keinen Einfluss auf die Verwendung der Befehle OUT und OUTB. Mit diesen Befehlen kann auch der Zustand der Ausgänge verändert werden, die vordefinierte Funktionen besitzen.
Portabilität	Nur bei 32-Bit Steuerungen wie M1 und MACS.
Parametergruppe	GL globale Parameter
Einheit	–
Bereich	–8 ... 8
Init-Wert	0

AINFTIME - 109

Inhalt Definiert die Filterzeit für den analogen Eingang.

Beschreibung Zum Setzen der Filterzeit für den analogen Eingang (MACS3) wird dieser globale Parameter genutzt. Wenn dieser Wert > 0 ist, wird der analoge Eingang jede ms gelesen und ein PT Filter benutzt um den Eingang zu glätten. Intern wird wie folgt gerechnet:

$$\text{filtval} = (((\text{analog IN 1}) * 1000) - \text{filtval}) / \text{AINFTIME} + \text{filtval}$$

Mit dem Befehl INAD 1 wird "filtval / 1000" zurückgegeben. (So erhält man eine bessere Auflösung.).

Portabilität Ab Firmware 6.2.35.

Parametergruppe GL globale Parameter

Einheit –

Bereich 0 ... MLONG

Init-Wert 0

CANSYNCTIMER - 114

Inhalt	Bestimmt die Zykluszeit für das Versenden von SYNC-Telegrammen auf dem CAN-Bus.
Beschreibung	<p>Der globale Parameter CANSYNCTIMER bestimmt die Periodenzeit für den zyklischen Versand von SYNC-Telegrammen. SYNC-Telegrammen dienen als Trigger um von CAN-Bus Geräten zyklisch Informationen (PDO) abzurufen.</p> <p>Bei CANopen-Leistungsendstufen und Drehgebern (vgl. Parameter DRIVETYPE, ENCODERTYPE, MENCODERTYPE) erfolgt die Datenanforderung typischerweise zyklisch im Reglertakt. Die Zykluszeit für das hierfür notwendige SYNC-Telegramm wird im Parameter CANSYNCTIMER festgelegt.</p> <p>Falls die Encoder-Information über den Bus übertragen wird, ist sicherzustellen, dass in jedem Positionsreglertakt auch eine aktualisierte Positionsinformation zur Verfügung steht. Dies bedeutet, dass die Einstellung des globalen Parameters CANSYNCTIMER entsprechend dem Takt des Positionsreglers und der Sollwert-Übernahme durch die Leistungsendstufe erfolgen muss. Der Timer sollte so wie im folgenden Beispiel sein (Default ist 1 ms):</p> <pre>SET TIMER 2 / 4 // Antriebe auf entsprechende Transmit-Typen setzen (d.h. 1 und 2) SET CANSYNCTIMER 2 // alle 2 ms wird ein SYNC gesendet // DRIVETYPE und REVERS setzen</pre> <p>!!! Bis Firmware-Version < 6.5.16 wurde der kleinste achsspezifisch eingestellte Wert des Parameters TIMER für die Periodizität des SYNC-Telegramm herangezogen.</p>
Portabilität	Ab Firmware 6.5.16
Parametergruppe	GL globale Parameter
Querverweise	DRIVETYPE, ENCODERTYPE, MENCODERTYPE, TIMER
Einheit	ms
Bereich	0 ... MLONG
Init-Wert	1

VMENCTYP - 116

Inhalt	Definiert die Quelle des virtuellen Encoders und den Encodertyp der verwendet wird, falls der virtuelle Encoder deaktiviert wird.
Beschreibung	<p>Definiert auf welchen Encodertyp der virtuelle Encoder gesetzt werden soll, wenn die virtuelle Funktion mit PULSACC 0 beendet wird. Außerdem kann mit diesem Parameter auch die Sollgeschwindigkeit einer anderen Achse als Vorgabe für den virtuellen Master gewählt werden.</p> $\text{VMENCTYP} = ax * 100 + \text{enctype}$ <p>$ax = 0$ heißt, der Wert, der durch PULSVEL Befehle vorgegeben wird, wird auf dem Encoder Ausgang ausgegeben (siehe PULSVEL, PULSACC).</p> <p>$ax = 1..n$ heißt, der Wert wird durch die Sollgeschwindigkeit der entsprechenden Achse vorgegeben.</p> <p>enctype definiert den ENCODERTYPE, welcher bei der Abschaltung des virtuellen Masters verwendet wird, d.h. kann die Werte des Parameters ENCODERTYPE annehmen (0 = inkremental, 1 = absolut, 5 = sinus cosinus).</p> <p>Wenn bei VMENCTYP die Sollgeschwindigkeit einer Achse gewählt wird muss zum Starten der Funktion ebenfalls PULSACC auf den gewünschten Wert gesetzt werden (dieser wirkt dann auch gegebenenfalls begrenzend) und es muss PULSVEL gesetzt werden. Der Wert von PULSVEL ist allerdings ohne Bedeutung und kann deswegen auch 0 sein. Lediglich der Aufruf von PULSVEL ist nötig um die Funktion zu starten.</p> <p>Ab Firmware 6.771 bestehen zwei weitere Möglichkeiten den virtuellen Master zu konfigurieren:</p> <p>Wird PULSACC auf -1 gesetzt, wird das Eingangssignal für den virtuellen Master von einem DDS [Data Distribution Service] erzeugt, welcher durch den Befehl SETSYS 15 definiert ist. In diesem Fall sind die Werte von PULSACC und PULSVEL außer Gebrauch.</p> <p>Außerdem gibt es noch eine Konfigurationsmöglichkeit für Anwender, die eine eigene oder erweiterte FPGA-Software installiert haben: Auf den Parameterwert von VMENCTYP kann Offset von 100000 beaufschlagt werden. In diesem Fall wird das Signal des virtuellen Masters nicht an dem Encoder-Ausgang (z.B. MACS4 ENC4) ausgegeben, sondern gelangt an einen internen Knotenpunkt, auf welchen eine erweiterte FPGA-Software zugreifen und die Signale weiterleiten oder verarbeiten kann. Der Eingang des virtuellen Masters wird unverändert wie oben beschrieben definiert. Daher wäre es möglich dieses Feature zusammen mit einem Vielfachen von 100 (-> Sollgeschwindigkeit einer Achse) oder mit PULSACC -1 zu verwenden.</p> <p>!!! Achtung: Ein aktualisierter VMENCTYP wird nur mit dem nächsten PULSVEL Befehl verwendet..</p>
Portabilität	Parameter verfügbar ab Firmware 6.6.18; ab Firmware 6.771 neue Varianten zur Verarbeitung des virtuellen Master-Signals.
Parametergruppe	Globale Parameter GL
Querverweise	VMENCTYP, VMENCRE, PULSVEL, PULSACC
Einheit	–
Bereich	0 ... MLONG
	0 = der Wert, der durch PULSVEL Befehle vorgegeben wird, wird auf dem Encoder Ausgang ausgegeben
	1 ... = $n * 100 + \text{encodertype}$ Nutzung der Sollgeschwindigkeit der Achse n als virtueller Master und Konfiguration auf Encodertyp bei abgeschaltetem virtuellem Master
	+100000 = Abschaltung des Encoderausgangs und Nutzung von speziellem FPGA-Code zur Weiterverarbeitung des virtuellen Master-Signals
Init-Wert	0

VMENCRES - 117

Inhalt	Auflösung des virtuellen Master-Encoders.
Beschreibung	Encoder-Auflösung des virtuellen Masters, die für die Index-Abstände verwendet wird. Beispiel: Wenn man also einen 500er Encoder simulieren möchte, müsste man die Parameter wie folgt setzen: <pre>SET VMENCRES 2000 // Erzeuge einen Indexpuls alle 2000 qc PULSACC 1000 // Beschleunige innerhalb von einer Sekunde // auf eine Geschwindigkeit von 1000 qc/Sek. PULSVEL 1000 // Fahre mit einer Geschwindigkeit von 1000 qc/Sek</pre>
Portabilität	Ab Firmware 6.6.18.
Parametergruppe	Globale Parameter GL
Querverweise	VMENCTYP, VMENCMTYP
Einheit	qc
Bereich	0 ... 65535
Init-Wert	500

VMENCMTYP - 118

Inhalt	Markertyp des virtuellen Master-Encoders.
Beschreibung	Definiert den Indextype für den virtuellen Master, analog zu den Einstellungen für SYNCMTYP. Das heißt, 0 = Indexpuls (Drehgeber Z positive Flanke) 1 = Indexpuls (Drehgeber Z negative Flanke) 2 = externer Marker (positive Flanke) 3 = externer Marker (negative Flanke) Zehnerstellen = nm n = digitale Eingangsnummer (1 ... 9) m = Master-Markertyp (0 ... 3)
Portabilität	Ab Firmware 6.6.18.
Parametergruppe	Globale Parameter GL
Querverweise	VMENCTYP, VMENCRES
Einheit	–
Bereich	0 ... MLONG 0 = Nullsignal
Init-Wert	0