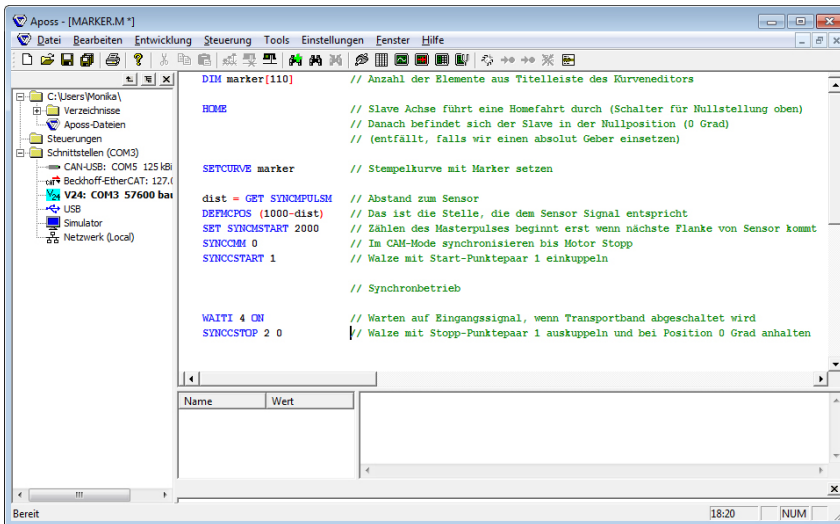


APOSS® win

Alle APOSS-Befehle in der Übersicht

Achsen-Positionier- und Synchronisier-Befehle für die Antriebstechnik



```

DIM marker[110] // Anzahl der Elemente aus Titelleiste des Kurveneditors

HOME // Slave Achse führt eine Homefahrt durch (Schalter für Nullstellung oben)
// Danach befindet sich der Slave in der Nullposition (0 Grad)
// (entfällt, falls wir einen absolut Geber einsetzen)

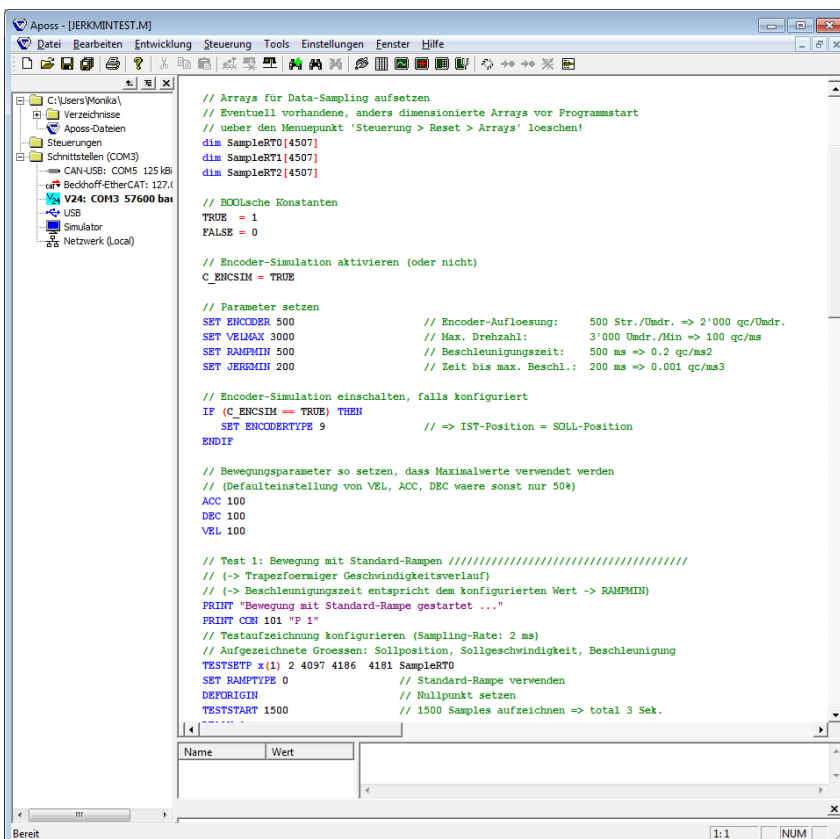
SETCURVE marker // Stempelkurve mit Marker setzen

dist = GET SYNCMPULSM // Abstand zum Sensor
DEFMCPDS (1000-dist) // Das ist die Stelle, die dem Sensor Signal entspricht
SET SYNCSTART 2000 // Zählen des Masterpulses beginnt erst wenn nächste Flanke von Sensor kommt
SYNCDOM 0 // Im CAN-Mode synchronisieren bis Motor Stopp
SYNCCSTART 1 // Walze mit Start-Punktpaar 1 einkuppeln

// Synchronbetrieb

WAITI 4 ON // Warten auf Eingangssignal, wenn Transportband abgeschaltet wird
SYNCCSTOP 2 0 // Walze mit Stopp-Punktpaar 1 auskuppeln und bei Position 0 Grad anhalten
    
```

Beispiel:
Transportband synchronisieren
mit Marker



```

// Arrays für Data-Sampling aufsetzen
// Eventuell vorhandene, anders dimensionierte Arrays vor Programmstart
// ueber den Menuepunkt 'Steuerung > Reset > Arrays' loeschen!
dim SampleRTO[4507]
dim SampleRT1[4507]
dim SampleRT2[4507]

// BOOLsche Konstanten
TRUE = 1
FALSE = 0

// Encoder-Simulation aktivieren (oder nicht)
C_ENCDSIM = TRUE

// Parameter setzen
SET ENCODER 500 // Encoder-Aufloesung: 500 Str./Umdr. => 2'000 qc/Umdr.
SET VELMAX 3000 // Max. Drehzahl: 3'000 Umdr./Min => 100 qc/ms
SET RAMPMIN 500 // Beschleunigungszeit: 500 ms => 0.2 qc/ms2
SET JERMIN 200 // Zeit bis max. Beschl.: 200 ms => 0.001 qc/ms3

// Encoder-Simulation einschalten, falls konfiguriert
IF (C_ENCDSIM == TRUE) THEN
SET ENCODERTYPE 9 // => IST-Position = SOLL-Position
ENDIF

// Bewegungsparameter so setzen, dass Maximalwerte verwendet werden
// (Defaulteinstellung von VEL, ACC, DEC waere sonst nur 50%)
ACC 100
DEC 100
VEL 100

// Test 1: Bewegung mit Standard-Rampen ////////////////////////////////////////////////////////////////////
// (-> Trapezfoermiger Geschwindigkeitsverlauf)
// (-> Beschleunigungszeit entspricht dem konfigurierten Wert -> RAMPMIN)
PRINT "Bewegung mit Standard-Rampe gestartet ..."
PRINT CON 101 "P 1"

// Testaufzeichnung konfigurieren (Sampling-Rate: 2 ms)
// Aufgezeichnete Groessen: Sollposition, Sollgeschwindigkeit, Beschleunigung
TESTSETP x(1) 2 4097 4186 4181 SampleRTO
SET RAMPTYPE 0 // Standard-Rampe verwenden
DEFORIGIN // Nullpunkt setzen
TESTSTART 1500 // 1500 Samples aufzeichnen => total 3 Sek.
    
```

Beispiel:
Testprogramm zur Aufzeichnung
von Bewegungsvorgängen mit
den verschiedenen Rampentypen
des Parameters RAMPTYPE.

Befehls-Referenz

Alle verfügbaren APOSS Befehle sind in folgende funktionale Gruppen organisiert:

Initialisierung	Befehle zum Initialisieren der Achse(n) und der Steuerung sowie zum Anfahren und Definieren der/des Nullpunkte(s).
Flow Control	Befehle zur Steuerung des Programmablaufs und zum Strukturieren von Programmen.
Ein-/Ausgang	Befehle zum Setzen und Lesen von Ein- und Ausgängen.
System Informationen	Befehle zum Abfragen von Status-Informationen wie Position von Antrieb und Geschwindigkeit, Systemzeit, Fehlerstatus, usw.
Interrupt-Funktionen	Setzen und Sperren von Interrupts sowie Aufrufen von Unterprogrammen.
Parameter	Alle mit einer Parameterkennung versehenen globalen und Achsparameter können mit folgenden Befehlen gesetzt und gelesen werden.
Fahren und Positionieren	Geschwindigkeit und Beschleunigung setzen, Starten und Stoppen, Positionieren.
Control Loop	Motorregelung mit benutzerdefinierten Positionen einschalten.
CANopen Basic	Diese Bibliothek enthält alle Funktionen zur Durchführung des „Raw CAN Telegram“-Transfers. und gepufferter Nachrichten.
CANopen Master	Diese Bibliothek bietet Funktionen um den Einsatz von CANopen-Slaves wie I/O-Module oder Drehgeber oder sogar Antriebe zu unterstützen. Diese Funktionen erlauben nicht nur die Konfiguration mit SDOs sondern auch den Einsatz von Prozessdaten durch PDOs.
CANopen Slave	Anwendungs-Software wäre es sogar möglich, sich wie ein DS402 Antrieb zu verhalten.
EtherCAT Master	Befehle zum Konfigurieren und Steuern des EtherCAT-Masters, zum Lesen und Setzen von SDOs sowie SDO2, um mit Debug-Einrichtungen wie einem SDO-Monitor zu kommunizieren.
Synchronisation	Befehle zum Synchronisieren der Slaves mit dem Master oder mit der Master-Simulation.
Kurvensteuerung	Befehle für die Synchronisation im CAM-Modus (Kurvenscheibensteuerung).
Speichern	Befehle zum Definieren und Speichern von Arrays.
Kommunikation	Befehle für das Handling der RS485 und RS233 Schnittstellen sowie der Parameter der Danfoss VLTs.
Debug	Befehle zur Ausgabe von Informationen zur Definition und

	zum Starten und Stoppen von Datenaufzeichnungen.
State Machine	Alle Befehle zur Programmierung einer Zustandsmaschine (State Machine). Abschnitt in Englisch.
Obsolete	Ältere, nicht mehr unterstützte Befehle.

Initialisierung der Steuerung

Befehle zum Initialisieren der Achse(n) und der Steuerung sowie zum Anfahren und Definieren der/des Nullpunkte(s).

CANOPENSLAVE	Initialisiert CANOpen-Funktionalität mit der eingestellten CAN-Nummer (id).
DEFCORIGIN	Sollposition als Nullpunkt setzen.
DEFMORIGIN	Aktuelle Master-Position als Nullpunkt für den Master setzen.
DEFORIGIN	Istposition als Nullpunkt setzen.
ERRCLR	Fehlermeldung löschen.
HOME	Maschinennullpunkt anfahren.
INDEX	Indexposition des Drehgebers anfahren.
MOTOR	Motorregelung einschalten/ausschalten.
RSTORIGIN	Temporärnullpunkt löschen.
SETMORIGIN	Aktuelle Position als Nullpunkt für den Master setzen.
SETORIGIN	Temporärnullpunkt setzen.

Flow Control

Befehle zur Steuerung des Programmablaufs und zum Strukturieren von Programmen.

BREAK	Abbruch der Befehlsausführung innerhalb einer WHILE or REPEAT Schleife oder einer SWITCH Anweisung.
CONTINUE	Abgebrochene Positionier- und Drehzahlbefehle fortsetzen, zum Beispiel nach einem MOTOR STOP.
DELAY	Zeitverzögerung.
DELAYUS	Zeitverzögerung in Mikrosekunden.
EXIT	Vorzeitiger Programmabbruch.
GOSUB	Aufruf eines Unterprogramms.
GOTO	Sprung zu einem Programmlabel.
IF THEN	Bedingte einfache Programmverzweigung.
... ELSE ... IF ... THEN	Bedingte mehrfache Programmverzweigung
... ELSE	Alternative Programmverzweigung.
... ENDIF	Ende der Programmverzweigung.
LOOP	Definierte Schleifenwiederholung.
NOWAIT ON/OFF	Wartemodus ein-/ausschalten.

REPEAT	Bedingte Schleife Anfang.
REPEAT ... UNTIL	Bedingte Schleife Ende.
SUBMAINPROG	Beginn der Definition des Unterprogramms.
.. ENDPROG	Ende der Definition des Unterprogramms.
SUBPROG	Beginn eines Unterprogramms.
... RETURN	Ende eines Unterprogramms.
SWITCH	Bedingte Befehlsausführung...
... CASE	... Befehlsausführung in Abhängigkeit eines Variablenwerts.
... DEFAULT	... Befehlsausführung ohne zutreffenden Zweig.
... ENDSWITCH	... Ende der bedingten Befehlsausführung.
WAITAX	Warten bis Zielposition erreicht ist.
WAITI	Warten auf bestimmten Eingangszustand.
WAITNDX	Warten auf Index.
WAITP	Warten bis Position erreicht.
WHILE ... DO	While-Schleife Anfang.
... ENDWHILE	While-Schleife Ende.

Ein-/Ausgangs-Befehle

IN	Eingänge bitweise lesen (einzeln).
INAD	Analogeingänge und Prozessdatenobjekte (PDO) von CAN-Objekten lesen.
INB	Eingänge byteweise lesen (8 Stück).
OUT	Digitale Ausgänge bitweise setzen (einzeln).
OUTAN	Analogausgang setzen (einer Achse zugeordnet).
OUTB	Digitale Ausgänge byteweise setzen (8 Stück).
OUTDA	(Frei verfügbaren) Analogausgang setzen und einen Ausgang, der als Antrieb initialisiert wurde setzen.

System Informationen

APOS	Istposition lesen.
APOSDIFF	Overflow-Handling von Inkrementalgebern in Anwendungen.
AVEL	Aktuelle Geschwindigkeit der Achse abfragen.
AXEND	Status der Programmausführung abfragen.
CPOS	Sollposition lesen.
CPOSDIFF	Overflow-Handling von Inkrementalgebern in Anwendungen.
ERRAX	Nummer der Achse, die den Fehler auslöst.

ERRNO	Fehlernummer lesen.
_GETVEL	Abtastzeit für AVEL und MAVEL ändern.
IPOS	Letzte Index- bzw. Markerposition des Slaves abfragen.
IPOSDIFF	Overflow-Handling von Inkrementalgebern in Anwendungen.
JERKFINVEL	Berechnet die Endgeschwindigkeit für einen ruckbegrenzten Stopp mit maximaler Beschleunigung/Verzögerung.
JERKSTOPDIST	Berechnet die notwendige Distanz für einen ruckbegrenzten Stopp mit maximaler Verzögerung.
MAPOS	Istposition des Masters abfragen.
MAPOSDIFF	Overflow-Handling von Inkrementalgebern in Anwendungen.
MAVEL	Aktuelle Geschwindigkeit des Masters abfragen.
MIPOS	Letzte Index- bzw. Markerposition des Masters abfragen.
MIPOSDIFF	Overflow-Handling von Inkrementalgebern in Anwendungen.
PID	PID-Berechnung durchführen.
STAT	Status der Achse lesen.
STATCLR	Löscht die Index-Flags von Slave und Master.
SYNCERR	Aktuellen Synchronisationsfehler einer Achse abfragen.
SYSVAR	Systemvariable (Pseudo-Array) liest Systemwerte.
TIME	Systemzeit auslesen.
TIMEHW	Internen Zähler auslesen.
TRACKERR	Aktuellen Schleppabstand einer Achse abfragen.
USRSTAT	Setzt Benutzer-Status.

Interrupt-Funktionen

DISABLE ...	Sperrt die Ausführung eines Interrupt-Typs oder löscht alle Definition(en) für einen kompletten Interrupt-Typ vollständig (Reset).
ENABLE ...	Gibt gesperrte Interrupts wieder frei.
ON CANINPUT id GOSUB	Unterprogramm aufrufen, wenn ein CAN-Telegramm vom Typ 'id' ankommt.
ON CANMSG GOSUB	Eintreffen einer gepufferten CAN-Nachricht.
ON COMBIT ... GOSUB	Unterprogramm aufrufen, wenn Bit n des Kommunikationspuffers gesetzt ist.
ON DELETE ... GOSUB	Löscht einen Positions-Interrupt: ON posint GOSUB.
ON DELETE SETOUT	Löscht alle Interrupts, die einen Ausgang setzen oder

	zurücksetzen.
ON ERROR GOSUB	Unterprogramm bei Fehler aufrufen.
ON .. GOSUB DELETE	Löscht einen einzelnen Interrupt
ON INT ... GOSUB	Unterprogramm bei Flanke eines Eingangs aufrufen.
ON KEYPRESSED GOSUB	Unterprogramm aufrufen, wenn ein Zeichen über eine serielle Schnittstelle kommt.
ON PARAM ... GOSUB	Unterprogramm aufrufen, wenn sich ein Parameter ändert.
ON PERIOD ... GOSUB	Unterprogramm in regelmäßigen Zeitabständen aufrufen.
ON posint ... GOSUB	Unterprogramm aufrufen, wenn ein Positions-Interrupt auftritt: <ul style="list-style-type: none">• ON APOS wenn die Slave-Position xxx passiert ist.• ON IPOS wenn der Abstand zwischen der letzten Markerposition und der Istposition erreicht ist.• ON MAPOS wenn die Master-Position xxx [qc] passiert ist.• ON MCPOS wenn die Master-Position xxx [MU] passiert ist.• ON MIPOS wenn der Abstand zwischen zwei Markern erreicht ist.
ON posint ... GOSUBOT	Definiert einmalige Positions-Interrupts.
ON posint SETOUT (TOIN)	Simuliert ein Nockenschaltwerk (CAM-Box) (alle Typen von Positions-Interrupts).
ON posint ... SETOUTOT	Definiert einmalige Positions-Interrupts.
ON STATBIT	Unterprogramm aufrufen, wenn Bit n des Statuswortes gesetzt ist.
ON TIME	Unterprogramm nach einmaligem Zeitablauf aufrufen.

Befehle für die Handhabung der Parameter

Alle mit einer Parameterkennung versehenen globalen und Achsparameter können mit folgenden Befehlen gesetzt und gelesen werden.

GET	Parameterwert lesen (Achsparemeter und globale Parameter).
LINKPDO	Systemvariable mit PDO verknüpfen und in die internen Parameter kopieren.
LINKSDO	Internes Objekt in PDO kopieren.
SET	Parameterwert setzen (Achsparemeter und globale Parameter).

Befehle zum Fahren und Positionieren

ACC	Beschleunigung setzen.
DEC	Verzögerung (negative Beschleunigung) setzen.

VEL	Geschwindigkeit für relative und absolute Bewegungen und max. zulässige Geschwindigkeit für Synchronisationen setzen.
POSA	Positioniere Achsen absolut.
POSR	Positioniere Achsen relativ.
LINA	Positioniere Achsen zeitsynchron absolut.
LINR	Positioniere Achsen zeitsynchron.
CSTART	Starten des Drehzahlmodus.
CSTOP	Stoppen des Antriebs im Drehzahlmodus.
CVEL	Geschwindigkeit für die Drehzahlregelung setzen.
MOTOR STOP	Stoppen des Antriebs.

Control Loop Befehle

MOTOR USERREFPOS	Motorregelung mit benutzerdefiniertem Positions-Sollwert einschalten.
MOTOR USERREFVEL	Motorregelung mit benutzerdefiniertem Geschwindigkeits-Sollwert einschalten.

CANopen Basic Befehle

CANDEL	Löscht alle oder einzelne CAN-Objekte.
CANIN	Liest ein Objekt.
CANOPENSLAVE	Initialisiert CANopen-Funktionalität mit der eingestellten CAN-Nummer (id).
CANOUT	Sendet ein Objekt (aktiv).
DEFCANIN	Definiert ein Empfangsobjekt.
DEFCANOUT	Definiert ein Sendeobjekt im CAN Controller.
INGLB	Liest eine Gb-CAN-Nachricht.
INMSG	Liest eine gepufferte CAN-Nachricht (Polling).
MSGVAL	Liefert den Long-Wert der letzten Nachricht.
ON CANINPUT	Ruft ein Unterprogramm auf, wenn ein CAN-Telegramm vom Typ 'id' eintrifft.
OUTMSG	Sendet eine gepufferte CAN-Nachricht (Polling).
USRSTAT	Setzt Benutzer-Status.

CANopen Master Befehle

Diese Bibliothek bietet Funktionen um den Einsatz von CANopen-Slaves wie I/O-Module oder Drehgeber oder sogar Antriebe zu unterstützen. Diese Funktionen erlauben nicht nur die Konfiguration mit SDOs sondern auch den Einsatz von Prozessdaten durch PDOs.

CANIN (>100)	Sammelt alle Transmit-PDOs von digitalen Eingangs-Modulen oder den CAN-Antriebsstatus durch ein einziges
--------------	--

	CAN-Telegramm.
CANINI	Initialisiert die notwendigen Objekte (PDOs) für den Datenaustausch mit CANopen-Teilnehmern oder aktiviert die erweiterten Funktionen CANINI und CANIN.
SDOREAD	Liest SDO eines angeschlossenen CANopen Gerätes.
SDOREADSEG	Segmentiertes Lesen von SDOs (ungepackt).
SDOREADSEGP	Segmentiertes Lesen von SDOs (gepackt).
SDOSTATE	Ergebnis einer aktiven Kommunikation prüfen.
SDOWRITE	Setzt SDO eines angeschlossenen CANopen-Gerätes.
SDOWRITEN	Setzt SDO mit definierter Anzahl gültiger Datenbytes (= Data size indicated) eines angeschlossenen CANopen Gerätes.
IN	Eingänge bitweise lesen (einzeln).
INB	Eingänge byteweise lesen (8 Stück).
OUT	Digitale Ausgänge bitweise setzen (einzeln).
OUTB	Digitale Ausgänge byteweise setzen (8 Stück).
SET ENCODERTYPE	SET Parameter ENCODERTYPE.
SET DRIVETYPE	SET Parameter DRIVETYPE.

CANopen Slave Befehle

LINKPDO	Systemvariable mit RxPDO verknüpfen und in die internen Parameter kopieren oder Teil eines Arrays in das PDO verknüpfen.
LINKSDO	TxPDO mit interner Systemvariable verknüpfen oder Teil eines Arrays in das PDO verknüpfen.
PDO[]	Pseudo-Array für den direkten Zugriff auf die CANopen-PDOs.
SYSVAR[0x01...]	Systemvariable (Pseudo-Array) liest Systemwerte; siehe SDO-Dictionary .

EtherCAT Master Befehle

ECATMASTERCOMMAND	Steuert den EtherCAT Master.
ECATMASTERCONFIG	Konfiguriert den EtherCAT Master.
ECATMASTERINFO	Liest Informationen aus dem EtherCAT Master.
PDO[]	Pseudo-Array für den direkten Zugriff auf die CANopen-PDOs.
SDOREAD	Liest SDO eines angeschlossenen CANopen Gerätes.
SDOREADSEG	Segmentiertes Lesen von SDOs (ungepackt).
SDOREADSEGP	Segmentiertes Lesen von SDOs (gepackt).

SDOSTATE	Ergebnis einer aktiven Kommunikation prüfen.
SDOWRITE	Setzt SDO eines angeschlossenen CANopen-Gerätes.
SDOWRITEN	Setzt SDO mit definierter Anzahl gültiger Datenbytes (= Data size indicated) eines angeschlossenen CANopen Gerätes.

Synchronisationsbefehle

Befehle zum Synchronisieren der Slaves mit dem Master oder mit der Master-Simulation.

DEFMORIGIN	Aktuelle Master-Position als Nullpunkt für den Master setzen.
DEFSYNCORIGIN	Definiert das Verhältnis von Master zu Slave für den nächsten SYNCP oder SYNCM Befehl.
MOVESYNCORIGIN	Synchronisationsursprung relativ verschieben.
PULSACC	Beschleunigung für den virtuellen Master setzen.
PULSVEL	Geschwindigkeit für den virtuellen Master setzen.
SETMORIGIN	Aktuelle Position als Nullpunkt für den Master setzen.
SYNCM	Winkel-/Positionssynchronisation mit Markerkorrektur.
SYNCMARKERSTART	Setzt einen Marker oder die Markerbehandlung zurück.
SYNCP	Winkel-/Positionssynchronisation.
SYNCSTAT	Flag für Synchronisationsstatus abfragen.
SYNCSTATCLR	Zurücksetzen der Flags MERR und MHIT.
SYNCV	Geschwindigkeitssynchronisation.
ACC	Beschleunigung setzen.
DEC	Verzögerung (negative Beschleunigung) setzen.
VEL	Geschwindigkeit für relative und absolute Bewegungen und max. zulässige Geschwindigkeit für Synchronisationen setzen.

Befehle zur Kurvensteuerung

Befehle für die Synchronisation im CAM-Modus (Kurvenscheibensteuerung).

CURVEPOS	Slave-Position, die der aktuellen Master-Position der Kurve entspricht, abfragen.
DEFMCPOS	Anfangsposition des Masters definieren.
POSA CURVEPOS	Slave auf die, der Master-Position entsprechenden Kurvenposition fahren.
SETCURVE	CAM-Kurve setzen.
SYNCC	Synchronisation im CAM-Modus.
SYNCCMM	Synchronisation im CAM-Modus mit Markerkorrektur des Masters.

SYNCCMS	Synchronisation im CAM-Modus mit Markerkorrektur des Slaves.
SYNCCSTART	Slave zur Synchronisation im CAM-Modus starten.
SYNCCSTOP	Slave nach der CAM-Synchronisation anhalten.
ACC	Beschleunigung setzen.
DEC	Verzögerung (negative Beschleunigung) setzen.
VEL	Geschwindigkeit für relative und absolute Bewegungen und max. zulässige Geschwindigkeit für Synchronisationen setzen.

Befehle zum Speichern

CAMDIM	Definiert ein globales CAM Array.
DELETE ARRAYS	Alle mit dem DIM-Befehl definierten Arrays im RAM löschen.
DIM	Definiert ein globales Array.
DIMARRAY	Übergibt ein DIM-Array an Funktionen durch Referenzierung der Array-Nummer des DIM-Arrays statt des Array-Namens.
MEMORYDUMP	Die MEMORYDUMP Funktion kopiert Daten vom flüchtigen Speicher der Steuerung auf die SD Karte.
MEMORYDUMPSTATE	Dieser Befehl kann verwendet werden, um den Status der MEMORYDUMP Funktion abzufragen.
SAVE ARRAYS	Arrays im Flash-Speicher permanent sichern.
SAVE AXPARS	Aktuelle Achsparameter im Flash-Speicher permanent sichern.
SAVE GLBPARS	Aktuelle globale Parameter im Flash-Speicher permanent sichern.
SAVEPROM	RAM-Speicher im Flash-Speicher permanent sichern.

Kommunikationsbefehle

DCPINIT	Initiiert den MACS4 RS485 und MACS5 RS233 Schnittstellen-Handler.
DCPREADCOMM	DCP Befehl für die MACS4 RS485 und MACS5 RS233 Schnittstelle: Prüft eingehendes Telegramm.
DCPWRITECOMM	DCP Befehl für die MACS4 RS485 und MACS5 RS233 Schnittstelle: Sendet Telegramm.
INKEY	Einlesen eines Zeichens (RS232).
VLTCOMMAND	Ermöglicht das Senden von Steuerwort (ControlWord) und Sollwerten zu Danfoss VLT via RS485.
VLTCOMSTATE	Ergebnis der letzten Parameter-Transaktion abfragen.
VLTREAD	Liest Parameter von einem Danfoss VLT via RS485.

VLTSTATUS Liest Status und aktuelle Drehzahl von Danfoss VLTs über RS485.

VLTWRITE Schreibt Parameter nach Danfoss VLTs über RS485.

Debug-Befehle

PRINT Ausgabe von Informationen.

PRINTDEV Stoppt die Ausgabe von Informationen.

PRINTF Ausgabe von formatierten Informationen

SPRINTF Ausgabe von formatierten Informationen

TESTSETDEST Definiert den Speicherbereich in dem eine Datenaufzeichnung abgelegt wird.

TESTSETINDEX Systemvariablen für die Datenaufzeichnung festlegen.

TESTSETTIME Konfiguriert die Abtastperiode für die Datenaufzeichnung.

TESTSETTYPE Definiert, ob eine einmalige oder zyklische Datenaufzeichnung ausgeführt werden soll.

TESTSTART Datenaufzeichnung starten.

TESTSTOP Datenaufzeichnung beenden.

State Machine Befehle

SmClear Clear the event queue.

SmConfig Define the state machine configuration parameters.

SmEvent Define an event and its parameters.

SmInput Request an event when a digital input changes.

SmMachine Define a state machine variable.

SmParam Request an event when a parameter value changes.

SmPeriod Start a periodic timer.

SmPosition Request an event when the motor passes a specified position.

SmPositionOnce Request an event when the motor passes a specified position for the first time.

SmPost Create an event and post it to the end of the event queue.

SmPublish Create an event and publish for all state machines.

SmRun Start state machines running.

SmState Define a state machine state, including its event handlers and substates.

SmStop Stop running state machines.

SmSubscribe Subscribe to receive published events.

SmTimer Start a "once only" timer.

SmUnsubscribe	Request to stop receiving published events.
SmUrgent	Create an event and post it to the head of the event queue.

Obsolete Befehle

#DEBUG	Befehl bis Firmware Version < 5.22
SWAPMENC	Zuweisung des Drehgebers
TESTSETP	Aufzeichnungsdaten für Testfahrt festlegen.
WAITT	Zeitverzögerung.
GETSYS/SETSYS	Speichern und Ausgeben benutzerdefinierter SYSVARS für Steuerungen bis MACS4. Ab MACS5-Steuerungen sind die notwendigen Funktionen in SDO 0x2203 implementiert.