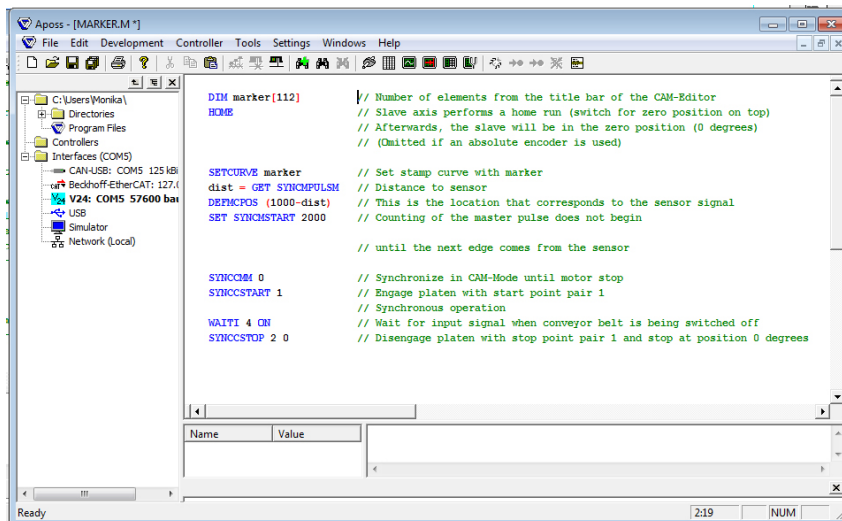


APOSS® win

An brief Overview of all APOSS Commands

Axis Positioning and Synchronization Commands for Drive Engineering



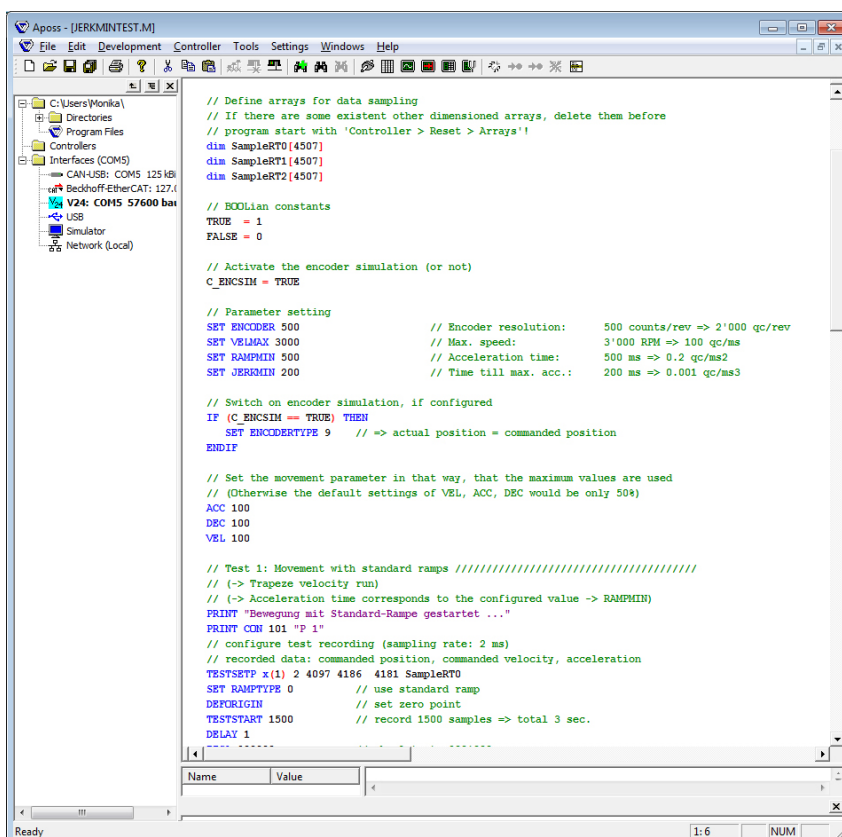
```

DIM marker[112] // Number of elements from the title bar of the CAM-Editor
HOME           // Slave axis performs a home run (switch for zero position on top)
              // Afterwards, the slave will be in the zero position (0 degrees)
              // (Omitted if an absolute encoder is used)

SETCURVE marker // Set stamp curve with marker
dist = GET SYNCMPULSM // Distance to sensor
DEFMCPPOS (1000-dist) // This is the location that corresponds to the sensor signal
SET SYNCMSTART 2000 // Counting of the master pulse does not begin
                  // until the next edge comes from the sensor

SYNCOM 0 // Synchronize in CAM-Mode until motor stop
SYNCSTART 1 // Engage platen with start point pair 1
           // Synchronous operation
WAITI 4 CN // Wait for input signal when conveyor belt is being switched off
SYNCSTOP 2 0 // Disengage platen with stop point pair 1 and stop at position 0 degrees
    
```

Sample:
Synchronise a belt with marker



```

// Define arrays for data sampling
// If there are some existent other dimensioned arrays, delete them before
// program start with 'Controller > Reset > Arrays!'
dim SamplerT0[4507]
dim SamplerT1[4507]
dim SamplerT2[4507]

// BOOLian constants
TRUE = 1
FALSE = 0

// Activate the encoder simulation (or not)
C_ENCDSIM = TRUE

// Parameter setting
SET ENCODER 500 // Encoder resolution: 500 counts/rev => 2'000 qc/rev
SET VELMAX 3000 // Max. speed: 3'000 RPM => 100 qc/ms
SET RAMPHIN 500 // Acceleration time: 500 ms => 0.2 qc/ms2
SET JERKMIN 200 // Time till max. acc.: 200 ms => 0.001 qc/ms3

// Switch on encoder simulation, if configured
IF (C_ENCDSIM == TRUE) THEN
  SET ENCODERTYPE 9 // => actual position = commanded position
ENDIF

// Set the movement parameter in that way, that the maximum values are used
// (Otherwise the default settings of VEL, ACC, DEC would be only 50%)
ACC 100
DEC 100
VEL 100

// Test 1: Movement with standard ramps ////////////////////////////////////////////////////////////////////
// (-> Trapeze velocity run)
// (-> Acceleration time corresponds to the configured value -> RAMPHIN)
PRINT "Bewegung mit Standard-Rampe gestartet ..."
PRINT CN 101 "P 1"

// configure test recording (sampling rate: 2 ms)
// recorded data: commanded position, commanded velocity, acceleration
TESTSETP x(1) 2 4097 4186 4181 SamplerT0
SET RAMPTYPE 0 // use standard ramp
DEPORIGIN // set zero point
TESTSTART 1500 // record 1500 samples => total 3 sec.
DELAY 1
    
```

Sample:
Test program for recording of
movements with different ramp
types of parameter RAMPTYPE.

Table of Contents

INI - Initialize	2
FLOW - Flow control	3-4
IO - Input/Output commands	5
SYS - System information	6-7
INT - Interrupt functions	8
PAR - Commands for parameter handling	9
MOV - Simple move commands	10
CTRL - Control loop commands	11
CANB - CAN-basic commands	12
CANOM - CANopen master commands	13
CANOS - CANopen slave commands	14
ECATM - EtherCAT master commands	15
SYNC - Synchronization commands	16
CAM - Curve control commands	17
MEM - Memory commands	18
COMM - Communication commands	19
DBG - Debug commands	20
SM - State Machine Commands	21
OBS - Obsolete commands	22

INI - Initialize

Commands to initialize the axes and the control unit start up and define the zero point(s).

CANOPENSLAVE	Initializes the CANopen functionality with the set CAN number (id).
DEFCORIGIN	Sets the command position as real zero point.
DEFMORIGIN	Set the current master position as the zero point for the master.
DEFORIGIN	Sets the actual position as real zero point.
DELETE ARRAYS	Delete all arrays in the RAM.
ERRCLR	Clears error.
HOME	Moves to machine zero point.
INDEX	Moves to next index position.
MOTOR OFF	Turns off motor control.
MOTOR ON	Turns on motor control.
RSTORIGIN	Reset temporary zero point.
SAVE ARRAYS	Save arrays in the flash memory permanently.
SAVE AXPARS	Save current axis parameters in the flash memory permanently.
SAVE GLBPARS	Save current global parameters in the flash memory permanently.
SAVEPROM	Save RAM in the flash memory permanently.
SETMORIGIN	Set the current position as the zero point for the master.
SETORIGIN	Set temporary zero point.

FLOW - Flow control

flowCommands for controlling the program flow, and structuring the programs:

BREAK	Break of command execution inside a WHILE or REPEAT loop, or a SWITCHstatement.
CONTINUE	Continue the positioning from point of interruption, e.g. following a motor-stop.
DELAY	Time delay.
DELAYUS	Time delay in Microseconds.
DIM	Declaration of a global array.
EXIT	Desired premature program termination.
GOSUB	Calling up a subroutine.
GOTO	Jumping within a program.
IF THEN	Conditional program execution.
... ELSE ... IF ... THEN	... with conditional alternative branching.
... ELSE	... with alternative branching.
... ENDIF	End of the conditional program execution.
LOOP	Repeats loop.
MOTOR STOP	Motor-stop with a programmed delay (with ramp).
NOWAIT ON/OFF	On/off switch for waiting for the command execution.
REPEAT	Beginning of repeat loop.
REPEAT ... UNTIL	Conditional loop, with an end criteria.
SUBMAINPROG	Start of the subroutine definition area.
... ENDPROG	End of the subroutine definition area.
SUBPROG	Begins a subroutine.
... RETURN	Ends a subroutine.
SWITCH	Conditional command execution, ...
... CASE	... command execution depending on the value of a variable.
... DEFAULT	... command execution without a fitting branch.
ENDSWITCH	... end of conditional command execution.
SYSVAR	System variable (pseudo array) reads system values.
WAITAX	Wait until target position is reached.
WAITI	Wait for input.
WAITNDX	Wait until the next index position is reached.
WAITP	Wait until a certain position is reached.
WAITT	Time delay in milliseconds.

<code>WHILE ... DO</code>	Conditional loop with commencement criteria.
<code>... ENDWHILE</code>	End of the loop.
<code>#INCLUDE</code>	Compiler directive: embedding further data.

IO - Input/Output commands

- IN** Reads input bit-by-bit (individually).
- INAD** Reads analog input and process data objects (PDO) of CAN objects.
- INB** Reads input by bytes (units of 8).
- OUT** Sets digital outputs bit-by-bit (single).
- OUTAN** Sets analog output (allocated to an axis).
- OUTB** Sets digital outputs by bytes (units of 8).
- OUTDA** Sets analog output (freely available) and an output, which is initialized as a drive.

SYS - System information

Commands for querying system status information such as drive position and velocity, system clock, error status, etc..
Commands for configuring and using arrays for recording test results.

APOS	Reads actual position.
APOSDIFF	Overflow handling of incremental encoders in applications.
AVEL	Queries actual velocity of an axis.
AXEND	Reads info on status of program execution.
CPOS	Reads set position.
CPOSDIFF	Overflow handling of incremental encoders in applications.
ERRAX	Number of the axis which triggers the error.
ERRNO	Reads error number.
IPOS	Queries last index or marker position of the slave.
IPOSDIFF	Overflow handling of incremental encoders in applications.
JERKFINVEL	Calculates the final velocity for a jerk-limited stop with maximum acceleration/deceleration.
JERKSTOPDIST	Calculates the necessary distance for a jerk-limited stop with maximum deceleration.
MAPOS	Queries actual position of the master.
MAPOSDIFF	Overflow handling of incremental encoders in applications.
MAVEL	Queries actual velocity of the master.
MIPOS	Queries last index or marker position of the master.
MIPOSDIFF	Overflow handling of incremental encoders in applications.
PID	Executes PID calculation.
PRINT	Output (display) texts and variables.
PRINTDEV	Stops information output.
STAT	Reads axis status.
STATCLEAR	Clears the slave and master index flags.
SYNCERR	Queries actual synchronization error of a slave axis.
TESTSETDEST	Defines the memory section for saving recorded data.
TESTSETINDEX	Specifies system variables for data recording.
TESTSETP	Specifies recording data for test run.
TESTSETTIME	Defines the sampling period for data recording.
TESTSETTYPE	Defines "one time" or "cyclic" recording.
TESTSTART	Starts the data recording.
TESTSTOP	Stops the data recording.

TIME	Reads internal controller time.
TIMEUS	Reads internal microsecond timer.
TRACKERR	Queries actual position error of an axis.
USRSTAT	Sets user status.
VLTCOMMAND.	Allows control word and references to be sent to Danfoss VLT via RS485.
VLTCOMSTATE	Polls for the result of the last parameter transaction.
VLTREAD	Reads parameters from a Danfoss VLT via RS485.
VLTSTATUS	Reads status and actual speed of Danfoss VLT over RS485.
VLTWRITE	Writes parameters to a Danfoss VLT over RS485.
_GETVEL	Changes sample time for AVEL and MAVEL .

INT - Interrupt functions

DISABLE ...	Locks the execution of interrupts.
ENABLE ...	Enables locked interrupts.
ON CANINPUT id GOSUB	Calls up a subprogram when a CAN telegram type 'id' arrives.
ON CANMSG GOSUB	Verifies arrival of a buffered CAN message.
ON COMBIT ... GOSUB	Calls up a subprogram when Bit n of the communication buffer is set.
ON DELETE .. GOSUB	Deletes a position-interrupt: ON posint GOSUB.
ON DELETE SETOUT	Deletes all interrupts which set or reset an output.
ON ERROR GOSUB	Calls subroutine in the event of an error.
ON .. GOSUB DELETE	Eliminates a single interrupt.
ON INT ... GOSUB	Calls subroutine depending on input signal.
ON KEYPRESSED GOSUB	Interrupts when a character appears via the serial interface when any key is pressed.
ON PARAM ... GOSUB	Calls up a subprogram when a parameter is altered.
ON PERIOD GOSUB	Calls subroutine at regular intervals.
ON posint ... GOSUB	Calls up a subprogram when a position interrupt occurs: <ul style="list-style-type: none">• ON APOS when the slave position xxx is passed.• ON IPOS when the distance between the last marker position and the actual position is reached.• ON MAPOS when the master position xxx [mu] is passed.• ON MIPOS when the distance between two markers is reached.
ON posint SETOUT (TOIN)	Simulates a cam box (all types of position interrupts).
ON STATBIT ... GOSUB	Calls up a subprogram when bit n of the status word is set.
ON TIME ... GOSUB	Calls subroutine after single timing.

PAR - Commands for parameter handling

All global and axis parameters with a parameter code can be set and read with the following commands:

- GET** Reads parameter values (axis and global parameter).
- LINKPDO** Link the system variable with PDO and copy in the internal parameters.
- LINKSDO** Copying an internal object in the PDO.
- SET** Sets parameter values (axis and global parameter).

MOV - Simple move commands

- ACC** Sets acceleration.
- DEC** Sets deceleration.
- VEL** Sets velocity for relative and absolute motions and set maximum allowed velocity for synchronizing.
- POSA** Positions axis absolutely.
- POSR** Positioning relative to the actual position.
- LINA** Synchronously positions axis absolutely.
- LINR** Synchronized relative positioning of several axes.
- CSTART** Starts the continuous movement in RPM mode.
- CSTOP** Stops the drive in speed mode.
- CVEL** Sets the velocity for speed regulation.

CTRL - Control loop commands

MOTOR USERREFPOS Turns on motor control on with user defined reference position.

MOTOR USERREFVEL Turns motor control on with user defined velocity reference

CANB - CAN-basic commands

CANDEL	Erases all or single CAN objects.
CANIN	Reads an object.
CANOUT	Sends message (active).
DEFCANIN	Defines a receive object.
DEFCANOUT	Defines a transmit object in the CAN controller.
INGLB	Reads Glb CAN message.
INMSG	Reads CAN message (polling).
MSGVAL	Supplies the long value of the last message.
OUTMSG	Sends CAN message (polling).

CANOM - CANopen master commands

CANINI	Initializes the necessary objects (PDOs) for data exchange of CANopen nodes, or enables extended CANINI, CANIN function.
SDOREAD	Reads SDO of a connected CANopen device.
SDOREADSEG	Segmented read of SDOs (unpacked).
SDOREADSEGP	Segmented read of SDOs (packed).
SDOSTATE	Checks the result of an active communication.
SDOWRITE	Sets SDO of a connected CANopen device.
SDOWRITEN	Sets a SDO and specifies the number of valid data bytes (= "Data set size is indicated").

CANOS - CANopen slave commands

`PDO[]` Pseudo array for direct access to the CANopen PDOs.

ECATM - EtherCAT master commands

ECATMASTERCOMMAND	Controls the EtherCAT master
ECATMASTERINFO	Reads information from the EtherCAT master
ECATMASTERCONFIG	Configures the EtherCAT master
SDOREAD	Reads SDO of a connected CANopen device.
SDOWRITE	Sets SDO of a connected CANopen device.

SYNC - Synchronization commands

Commands to synchronize the slave with the master or the master simulation:

DEFSYNCORIGIN	Defines master-slave relation for the next <code>SYNCP</code> or <code>SYNCM</code> command.
MOVESYNCORIGIN	Relative shifting of the origin of synchronization.
PULSACC	Sets acceleration for the virtual master.
PULSVEL	Sets velocity for virtual master.
SYNCM	Synchronization of angle/position with marker correction.
SYNCMARKERSTART	Resets a marker or resets marker handling.
SYNCP	Synchronization of angle/position.
SYNCSTAT	Queries flag for synchronization status.
SYNCSTATCLR	Resetting of the flags MERR and MHIT.
SYNCV	Synchronization of velocity.
ACC	Sets acceleration.
DEC	Sets deceleration.
VEL	Sets velocity for relative and absolute motions and set maximum allowed velocity for synchronizing.

CAM - Curve control commands

Commands for synchronization in CAM-Mode (cam control):

CURVEPOS	Retrieve slave curve position that corresponds to the current master position of the curve.
DEFMCPOS	Define initial position of the master.
POSA CURVEPOS	Move slave to the curve position corresponding to the master position.
SETCURVE	Sets CAM curve.
SYNCC	Synchronization in CAM mode.
SYNCCMM	Synchronization in CAM mode with master marker correction.
SYNCCMS	Synchronization in CAM mode with slave marker correction.
SYNCCSTART	Starts slave for synchronization in CAM mode.
SYNCCSTOP	Stops slave after the CAM synchronization.
ACC	Sets acceleration.
DEC	Sets deceleration.
VEL	Sets velocity for relative and absolute motions and set maximum allowed velocity for synchronizing.

MEM - Memory commands

SAVEPROM Save RAM in the flash memory permanently.

MEMORYDUMP

DIMARRAY

CAMDIM

COMM - Communication commands

INKEY	Reads ASCII-character from interface (RS232).
PRINTF	Formatted information output.
PRINTHEX	
DCPREADCOMM	DCP command for MACS4 RS485 or MACS5 RS232 interface: Checks for an incoming telegram.
DCPWRITECOMM	DCP command for MACS4 RS485 or MACS5 RS232 interface: Sends telegram.
DCPINIT	Initializes the RS485 (MACS4) or RS232/RS485 (MACS5) interface handler.

DBG - Debug commands

#DEBUG	Command up to firmware version < 5.22
TESTSETDEST	Defines the memory section for saving recorded data.
TESTSETINDEX	Specifies system variables for data recording.
TESTSETTIME	Defines the sampling period for data recording.
TESTSETTYPE	Defines "one time" or "cyclic" recording.
TESTSTART	Starts the data recording.
TESTSTOP	Stops the data recording.

SM - State Machine Commands

Commands related to state machines.

SmEvent	Define an event and its parameters.
SmState	Define a state machine state, including its event handlers and substates.
SmMachine	Define a state machine variable.
SmConfig	Define the state machine configuration parameters.
SmRun	Start state machines running.
SmStop	Stop running state machines.
SmPost	Create an event and post it to the end of the event queue.
SmUrgent	Create an event and post it to the head of the event queue.
SmPublish	Create an event and publish for all state machines.
SmSubscribe	Subscribe to receive published events.
SmUnsubscribe	Request to stop receiving published events.
SmClear	Clear the event queue.
SmTimer	Start a "once only" timer.
SmPeriod	Start a periodic timer.
SmParam	Request an event when a parameter value changes.
SmInput	Request an event when a digital input changes.
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.

OBS - Obsolete commands

#DEBUG	Command up to firmware version < 5.22
SWAPMENC	Encoder assignment