

APOSS[®] win

Development Environment and Programming Language

User Manual



What is APOSS?

APOSS is a programming language that offers an optimized syntax for effective programming of drive solutions. The APOSS language structure is based on established high level languages like Basic and C, but extends them with high performance motion control commands. Even very complex positioning and synchronization functions can be initiated with APOSS using simple commands and will run completely autonomously as background tasks. Unlike typical PLC systems, interrupt functions allow fast, real-time reactions that are independent from the program main loop cycle time.

The APOSS Integrated Development Environment (APOSS-IDE or also simply APOSS) is a comprehensive software development tool for programming, testing, and debugging motion control solutions by zub machine control AG. The APOSS Programming Language is an integral part of the APOSS-IDE. The APOSS-IDE combines the following components in a high-performance integrated environment:

- APOSS Programming Language
- Editor with Syntax-Highlighting
- Context-sensitive Online-Help
- Program and Parameter Handling
- Online-Debugging of Process data and Variables
- Smart-Oscilloscope with real-time support
- Graphical CAM profile editor

The APOSS-IDE is supported on PC systems running Microsoft Windows XP, Windows Vista, or Windows 7.

The APOSS-IDE can be freely downloaded by visiting www.zub.ch.

What's New in APOSS 6.10 NOTE: The new programming features (compiler extensions, commands, parameters, etc.) of APOSS version 6.10 can only be used for controllers that contain firmware version 6.8.01 and above. Code compiled for these controllers will compile to code version 8.

!!! This code is not backwards compatible and cannot be run on controllers containing older versions of firmware. Older controllers can continue to use previously existing programming features.

APOSS-IDE The APOSS-IDE has been enhanced in several ways intended to improve its ease-of-use. A complete list of changes can be found in the file "ApossChanges.pdf" that is located in the APOSS install directory. The following are the more important general changes:

- The APOSS-IDE supports a new **Sidebar** window feature that makes access to multiple files and multiple controllers much easier.
- The **"Find in Files"** feature has been enhanced and a new **Findbar** window has been added to improve support for multiple files.
- Files previously open in the APOSS-IDE will be reopened the next time that the APOSS-IDE is started. As well, controllers that were previously connected will be reconnected. These features can be enabled and disabled using the **Settings → Options** menu command.
- A **"Save All Source"** toolbar button  has been added that will save all currently open program files. It is no longer necessary to save files one at a time. (See also **Save and Save as**.)
- The **"Controller → Parameters → Global, Axis, and Name"** menu commands have been replaced with a single **Controller → Parameters → Edit** command that provides access to all parameters.
In addition, parameter changes are NOT saved for any controller until the user presses the "OK" button. Previously, parameters were saved when a new controller was selected even if the user subsequently pressed the "Cancel" button.

The following Oscilloscope changes have been made:

- The **Settings → Oscilloscope** menu command has been enhanced to provide more user control: User-selectable columns in the Curve List, line thickness, grid color.
- Cursors are now dragged using the left mouse button. If the right mouse button is used, then both cursors will be dragged at the same time.
- For digital signal curves, the **"Cursor Difference (B-A)"** column can now be set to display the number of transitions between the cursors.
- The "Shift-Up" and "Shift-Down" arrow keys can now be used to move curves within the Curve List.
- If the Single Shot Oscilloscope is being used with a non-cyclic group, then runs will now be automatically stopped once data collection is complete.

The following CAM Editor changes have been made:

- A new "Create array assignment on clipboard" toolbar button has been added. This can be used to create a set of array assignment statement for the currently selected CAM curve. The statements are placed on the Windows clipboard and can then be pasted into an APOSS program.

The following Simulator changes have been made:

- The Simulator can now be auto-started if it is not already running. It can also be automatically closed when a connection is lost (e.g. when APOSS is closed). These can be set using the **Settings → Interface** menu command and selecting the **Simulator** tab.

- The Simulator now saves its configuration information (e.g. SDO readout configuration).
- Multiple Simulators can now be supported simultaneously using different IP port numbers.

The following Batch Mode changes have been made:

- The Windows command line syntax for Batch Mode has been enhanced so that it supports a controller ID list.

Character Handling in APOSS and the Compiler

APOSS writes program (.m) files to disk using a character encoding based on the content of the program. APOSS does this to improve backwards compatibility. If a program file contains any characters (including characters in comments) that cannot be represented as common ANSI characters (e.g. Russian characters), then the file is written using UTF-8 encoding. Otherwise, the file is written using ANSI encoding. APOSS provides the **Edit → Keyboard** dialog to allow users to insert non-ANSI characters into programs.

When APOSS saves names in the controller (e.g. program names and controller names), then encoding is also based on the content of the name. If the name contains any characters that cannot be represented as common ANSI characters (e.g. Russian characters), then the name is saved using UTF-8 encoding. Otherwise, the name is saved using ANSI encoding. Note that UTF-8 encoding can be more "verbose" (i.e. require more bytes) than ANSI encoding. This can limit names to a smaller number of characters if UTF-8 encoding is required since names have a fixed byte-length in the controller. APOSS itself provides no mechanism to insert non-ANSI characters into names. However, other mechanisms can be used such as Cut/Paste and Windows keyboard drivers.

The compiler handles characters and character strings in different ways, based on the firmware version of the controller. For firmware versions older than 6.07.56, characters in programs are handled as follows:

- All characters in character strings are left "asis". If they are saved in the program (.m) file using ANSI encoding, then they are compiled into the binary program as ANSI bytes. If they are saved using UTF-8 encoding, then the UTF-8 bytes are compiled into the binary program. This allows controllers containing old firmware to provide a reasonable level of support for non-ANSI characters while still allowing existing programs to run unchanged.
- Single-quoted character assignments will always be converted automatically to hex assignments using ANSI encoding. For example "abc = 'ü'" will be converted to "abc = 0xFC". If the character has no ANSI equivalent (e.g. Russian characters), then the character will be replaced with 0xBF (i.e. '¿').
- If UTF-8 Byte Order Marks are found in the file, they will be discarded (even if the original file is ANSI encoded).

For firmware version 6.07.56 and newer, characters in programs are handled as follows:

- All characters in character strings are compiled into binary programs as UTF-8 strings, regardless of how they have been encoded in the original program file.
- Single-quoted character assignments will always be converted automatically to hex assignments using Unicode (UCS-2) encoding. For example, "abc = 'ü'" will be converted to "abc = 0x00FC".
- If UTF-8 Byte Order Marks are found in the file, they will be discarded (even if the original file is ANSI encoded).

APOSS Compiler Extension

The APOSS Compiler has been extended with many new features:

- New array **copy function** for two-dimensional arrays.
- More arrays are accessible by **SDO** (0x2140 .. 0x215F).
- The **_** (underscore) character can be used to separate groups of **hexadecimal numbers** for better readability.
- **Assignment operator** (i.e. '=') supports **copy of arrays**

Compiler version \geq 6.6.04 (= "Code Version 8"):

- New variable and array types: LONG, DOUBLE, and STRING: (see "Data handling" **Using Arrays**)
Variable arrays can be defined and used local or global.
Fully support of all array types in their two-dimensional version.
All array types can be used in DIM statements.
- Passing arrays as well as two-dimensional arrays:
In addition to the previous possibilities of the function parameter handling it is also possible to pass arrays of type DOUBLE or STRING to a **function** now.
- Passing variables to **functions** by reference.
- New STRING copy functions (see **STRING Array**):
Extended Array copy functions to copy STRING and DOUBLE arrays,
Extended STRING functions,
STRING copy command,
STRING concatenation command,
STRING length function,
PRINT command adapted to the STRING arrays.

New and Extended Commands

CANOPENSLAVE	This command is still supported but EtherCAT is not initialized, if CANOPENSLAVE is executed.
DIM	Extension: Arrays of type DOUBLE and STRING as well as variables of type STRING are supported.
DISABLE ... -INT	Extension: Parameter for a total reset of an interrupt type.
ON CANINPUT	Extension: CANINPUT value for RS485.
ON xxx GOSUB DELETE	Eliminates an interrupt.
SDOREADSEGP	Extension: STRING arrays can be used in READSDOSEG statements. This makes it possible to use PRINT commands to print the strings later on.
SDOWRITEN	Similar to SDOWRITE, but specification of the number of valid data bytes (= data size indicated).
STAT	The new bit 3 of byte 2 in the STAT command flags "Amplifier power NOT ok" for MACS4-DC3 and "Amplifier voltage below 20V" for MACS4-DC6
SYNCMARKERSTART	New Parameter value '2' to reset the marker distance and marker window size immediately without waiting for next SYNCM or SETCURVE.

New and Extended Parameters

The following new axis parameters are supported or were extended:

AMPENCNO 126	Definition of the encoder input, which is in use as the feedback source of the velocity control loop.
AMPENCRES 127	Resolution [qc/turn] of the encoder in use by the velocity control loop.
AMPENCRPM 128	Maximum velocity [rpm] of the encoder in use by the velocity control loop.
DRIVETYPE	Extension: DRIVETYPE 13 for Kollmorgen AKD.
POSDRCT 28	Extension: The kind of feedback evaluation (normal or inverted) can be configured separately for the velocity and position control loop.
SYNCTYPE 51	Extension: New SYNCTYPE value to buffer negative master movements.
VMENCTYP 118	Extension: New features to handle and process the virtual master signal.

New SYSVAR Indices

Extension of the [SDO Internal System Data](#):

- SDO 0x2216, 0x2217 and 0x2218 for FPGA.
- 4316 PFG_CMASTERCLEN to read out the master curve length of the actual curve running.
- 4317 PFG_CALCCURVEPOS to calculate the curve slave position for a given master position

New DCP Commands for MACS4

The new DCP commands support the RS485 interface.

Legal Information and Conventions

© zub machine control AG, 2013

All rights reserved. Reproducing or disassembling the computer algorithms in this program is strictly forbidden. All information contained in this manual is subject to technical changes.

Version 6.10

zub machine control AG
Buzibachstrasse 31
CH-6023 Rothenburg
Switzerland

Phone +41 41 54150-40

Fax +41 41 54150-49

www.zub.ch

www.aposs.de

- Disclaimer of Warranty Every effort has been made to make this information and technical data as complete and as accurate as possible, but no warranty or fitness is implied. The information and technical data are provided on an "as is" basis. zub machine control AG shall have neither liability nor responsibility to any person or entity with respect to any loss or damage arising from the information or technical data or from the use thereof.
- Trademarks APOSS, APOSS OS, and the zub logo are registered trademarks of zub machine control AG. Microsoft, Windows XP, Windows Vista, and Windows 7 are either registered trademarks or trademarks of the Microsoft Corporation in the USA and other countries. VLT is a registered Danfoss trademark.
- Conventions** The information in this manual follows the system and uses the typographical features described below to the greatest extent possible:
- Caution Hints and cautionary remarks are highlighted with "!!!" and 
- Menus and Functions Menus and functions as well as buttons are printed in bold text, for example: **Controller → Parameter** or **→ OK**.
- Commands and Parameters Commands and parameter names are written in capitals, for example: AXEND and KPROP; Parameters are printed in bold faced italics, for example: ***Proportional Factor***.
- Keys The names of keyboard keys and function keys are printed in bold face, for example the **Ctrl** key, the **Esc** key, or the **F1** key.
- Cross References Cross references to other documents are printed between quotes, for example: see "Hardware reference".
- Cross references to other parts of the text are underlined and are also marked in color, for example **Command List**. Click on the marked link and the online help jumps to the desired section. Click on **Back** in the menu bar to return.
- Underlined and blue marked links can also show images, graphics, and explanations from the glossary, for example MLONG. Click beside the popup or press the **Esc** key to close the popup.
- Illustrations All illustrations in this document that show screen images were created using the APOSS-IDE running on a Windows XP computer. If the APOSS-IDE is running on a different Windows operating system, then the screens may be similar but not exactly the same.

Content Getting Started

APOSS User Interface

APOSS Tools

- CAM Editor
- Array Editor
- Oscilloscope

Motion Control Topics

- Optimizing PID Controller
- CAM Control
- CAM Box
- Limited Jerk

Programming with APOSS

- Introduction
- Preprocessor
- APOSS Command Groups
- Command Reference
- Overviews APOSS Program Samples

Parameter Reference

- Basics
- Parameter Overview
- Axis Parameters
- Global Parameters

Error Reference and Messages

Technical Reference

- Functions and commands for MACS4
- Array Structure of CAM Profiles
- Illustrations
- SDO Object Dictionary – please see online help
- Versions History – please see online help

We wish you the best in your work with APOSS! For questions regarding programming or operating the controller, please contact your supplier or zub machine control AG directly.