APOSS Benutzeroberfläche

Das APOSS-Fenster 3

Tastatur- und Mausfunktionen 5

- Funktionstasten 5
- Esc-Taste 5
- Maus-Funktionen 5
- Tastatur-Funktionen 6
- Funktion Rückgängig 8
- Tabulatoren 8
- Makro Aufzeichnen und Abrufen 8

Menü Datei 9

- Neu 9
- Beispiel 9
- Speichern, Speichern als 9
- Programmende 9

Menü Bearbeiten 10

- Einfügen Array-Zuweisung 10
- Suchen und Ersetzen 10
- Suchen in Dateien 11
- Lesezeichen 13
- Tab umwandeln 13
- Alle Makros löschen 13
- Tastatur 13

Menü Entwicklung 14

- Ausführen F5 14
- Abbrechen [Esc] und Fortsetzen 15
- Meldungen -> Log-Datei 15
- Debug Befehle 15
- Überwachung hinzufügen / starten / stoppen 16
- Syntaxprüfung F4 17
- In Datei kompilieren 17
- Abbruch alle 17
- Start alle 17
- Download alle 17
- Alle Programme löschen 18
- Steuerung auswählen 18
- Schnittstelle schließen / Alle Schnittstellen schließen 19
- Befehlshilfe 19

Menü Steuerung 20

- Programme 20
- Parameter 24
- Speicher 26
- Reset 26
- Fehler Historie 26
- Diagnose Bericht 28
- Diagnostik 28
- Firmware Upload 28

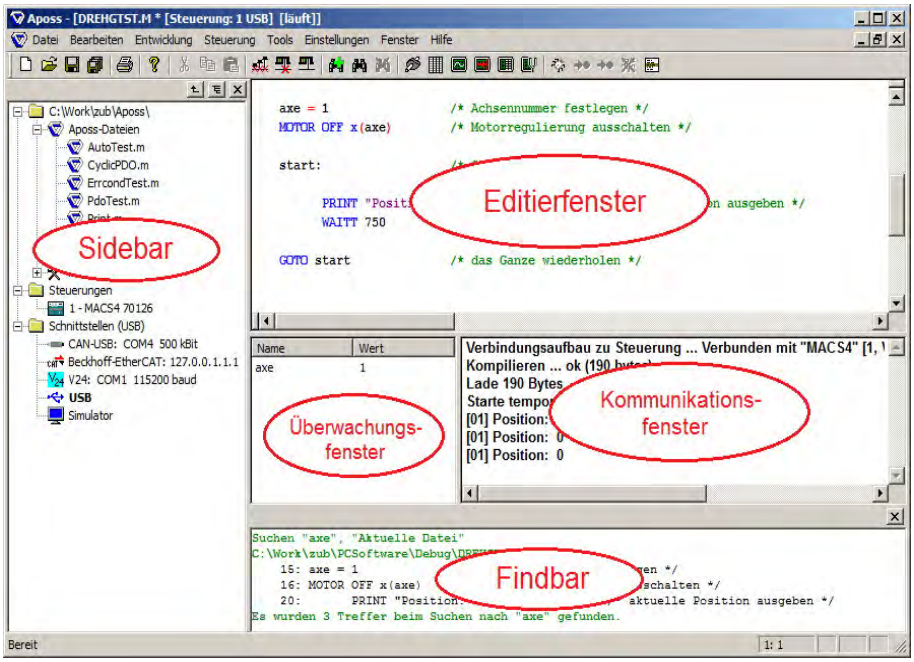
Menü Tools 28

Menü Einstellungen 29
Compiler 29
Schnittstellen Parameter 30
Sprache 30
Editor Einstellungen 31
Optionen 31
Oszilloskop 33
Menü Fenster 34
APOSS Sidebar 34
APOSS Findbar 35
Menü Hilfe 36
Menü Download 37
Download Firmware 37
Download Programme 38
Programme debuggen 40
Debugger starten 40
Debugger stoppen 41
Einzelschritt 41
Breakpoints anwenden 41
Darstellung und Änderung von Variablen 42
Darstellung der ausführenden Zeile 42

Sie sollten mit der Windows-Oberfläche und der Windows-Terminologie vertraut sein, denn diese Gebrauchsanweisung erklärt nicht die Grundlagen, aber alle Besonderheiten der APOSS-Benutzeroberfläche.

Das APOSS-Fenster

Das APOSS-Fenster erlaubt den gleichzeitigen Zugriff zu einem APOSS-Programm und einer Steuerung. Sie können mehrere APOSS-Fenster öffnen und jedes mit einem anderen APOSS-Programm und Steuerung verbinden.



Von oben nach unten:

Titelleiste — Die **Titelleiste** zeigt den Name des APOSS-Programms. Wenn eine Steuerung angeschlossen ist, dann sehen Sie auch die ID-Nummer und das Anschluss-Interface der Steuerung. Löst eine Steuerung einen Fehler aus, wird die Fehlernummer ebenfalls in der Titelleiste angezeigt.

Menüleiste — Die **Menüleiste** bietet die Auswahl der jeweiligen Funktionen.

Toolbar — Die **Toolbar** (**Schaltsymbolleiste)** bietet einen schnellen Zugriff auf häufig benutzte Funktionen. Die einzelnen Buttons (Symbole) werden an den entsprechenden Abschnitten im Handbuch erläutert.

Sidebar — Die **Sidebar** bietet dem Anwender einen schnellen Zugang , wenn mit mehreren Programmen, Steuerungen und verbundenen Schnittstellen gearbeitet wird; siehe APOSS Sidebar.

Editierfenster — Das **Editierfenster** im oberen Bereich zeigt das APOSS-Programm, das gerade bearbeitet wird und bietet alle üblichen Funktionen eines Text-Editors zum Bearbeiten. Hierfür stehen viele Mausfunktionen sowie Shortcuts zur Verfügung; siehe unten.

Klicken Sie mit der Maus auf einen Befehl oder eine Funktion im Programm und ein kleines Popup-Fenster zeigt die Syntax dieses Befehls bzw. dieser Funktion. In **Einstellungen → Editor** können diese Popup-Fenster deaktiviert werden.

Drücken Sie **F1** während der Mauszeiger über einem Befehl oder einer Funktion steht und die entsprechende Seite der Online-Hilfe für diesen Befehl oder diese Funktion wird dargestellt.

Verschiedene Farben erleichtern Ihnen die Unterscheidung zwischen Kommentaren, Programmteilen, Operatoren, Ziffern usw. Sie können die Farbzuordnung mit **Einstellungen → Editor** ändern.

Überwachungsfenster — Das **Überwachungsfenster** links unterhalb des Editierfensters ist besonders nützlich beim Debuggen. Damit kann sowohl die Steuerung als auch das ausführende Programm während die Steuerung läuft überwacht werden. Was überwacht werden soll, können Sie mit **Entwicklung → Überwachung hinzufügen** einstellen.

Kommunikationsfenster

Das **Kommunikationsfenster** rechts daneben zeigt sowohl die Meldungen der APOSS-IDE (z.B. Meldungen des Compilers) als auch Meldungen der Steuerung (z.B. programmierte PRINT Befehle). Den Meldungen der Steuerung wird die ID-Nummer der Steuerung vorangestellt (z.B. „[#01]“ wie im Beispiel oben).

Man kann Kommentare hinzufügen, in den und aus dem Zwischenspeicher kopieren sowie nach Strings suchen.

Wenn nach dem Kompilieren eines Programms eine Fehlermeldung im Kommunikationsfenster steht, doppelklicken Sie auf diese und der Editiercursor wird in die Zeile gestellt, die den Fehler enthält. Falls der Fehler in einer Include-Datei ist, wird diese Datei geöffnet

Findbar

Der Bereich **Findbar** zeigt die Ergebnisse, wenn in mehreren Dateien gesucht wird. Dieses Fenster wird detailliert in APOSS Findbar beschrieben.

Statuszeile

Die **Statusleiste** ganz unten zeigt links den Status, z.B. „Bereit“ und rechts die Zeilennummer sowie die Cursorposition.

Innerhalb des Programms können Sie sich anhand der Zeilennummern orientieren. Die Syntaxprüfung zum Beispiel stellt nicht nur den Cursor in die Zeile mit einem Fehler, sondern nennt auch die Zeilennummern, die Fehler enthalten im Kommunikationsfenster.

Die aktuelle Zeilennummer finden Sie in der Statuszeile, zum Beispiel 13:1. Der Cursor steht dann in der Zeile 13 vor dem ersten Zeichen.

Kontextmenüs

An manchen Programmstellen werden Kontextmenüs (Popup-Menüs) angeboten, wenn Sie auf die rechte Maustaste klicken. Zum Beispiel wird mit einem Rechtsklick in das Editierfenster das Kontextmenü „Bearbeiten“ gezeigt. Wann immer Kontextmenüs verfügbar sind, werden sie in den folgenden Abschnitten beschrieben.

Die Kontextmenüs werden automatisch wieder verlassen, wenn die ausgewählte Funktion ausgeführt wird oder wenn Sie mit der linken Maustaste an eine beliebige andere Stelle im Bildschirm klicken.

Tastatur- und Mausfunktionen

Funktionstasten

Häufig benötigte Funktionen sind auf die Funktionstasten gelegt:

- **F1** Online-Hilfe
- **F2** Zum nächsten Lesezeichen springen.
- **F3** Weitersuchen (beim Suchen)
- **F4** Syntax des Programms prüfen
- **F5** Programm ausführen
- **F9** Programm zeilenweise ausführen (nur im Debug Modus)
- **F11** System-Prozessdaten in der Online-Hilfe aufrufen
- **F12** Befehlshilfe aufrufen

Die anderen Funktionstasten werden an den entsprechenden Stellen erwähnt.

Esc-Taste

In Standard Windows-Anwendungen schließt die **Esc**-Taste normalerweise das aktive Fenster. Im APOSS-Fenster jedoch öffnet die **Esc**-Taste automatisch eine Verbindung zu einer Default-Steuerung, wenn keine Steuerung angeschlossen ist.

Wenn schon eine Steuerung angeschlossen ist, beendet die **Esc**-Taste alle gerade laufenden Programme der Steuerung.

!!!

Wenn ein laufendes Programm abgebrochen wird, während sich der Antrieb dreht, wird der Antrieb mit der maximal zulässigen Geschwindigkeit abgebremst. Stoppt man einen Antrieb während der Fahrt auf diese Weise, kann das System, das mit dem Antrieb verbunden ist. beschädigt werden.

Um zu vermeiden, dass der Anwender unbeabsichtigt durch Drücken der **Esc**-Taste ein laufendes Programm abbricht, kann in **Einstellungen** → Optionen eingestellt werden, dass Abbrechen nur mit **Umschalt+Esc** möglich ist. Ist diese Option gesetzt, kann der Anwender weiterhin ein ausführendes Programm abbrechen, jedoch wird durch Drücken der **Umschalt+ Esc** verhindert, dass es unbeabsichtigt passiert.

Eine unterbrochene Verbindung wiederherstellen

Wenn eine aktive Verbindung zu einer Steuerung unterbrochen ist (zum Beispiel, wenn die Steuerung ausgeschaltet oder die Kommunikation getrennt wurde), meldet die Titelleiste des APOSS-Editierfensters, das mit dieser Steuerung verbunden war, „Verbindung unterbrochen". Aber das Editierfenster bleibt mit dieser Steuerung „verbunden". Wenn die **Esc**-Taste gedrückt wird (oder irgendein Befehl ausgeführt werden soll, der eine Kommunikation zur Steuerung erfordert) wird APOSS versuchen, die gleiche Steuerung wieder zu verbinden. Dieses Verhalten ist nur relevant, wenn mehrere Steuerungen an einem Kommunikationsstrang vorhanden sind, z.B. CAN-Bus.

In bestimmten Fällen ist es von Vorteil, wenn APOSS versucht, automatisch eine verlorene Verbindung zu einer Steuerung wiederherzustellen. Zum Beispiel kann dies nützlich sein, wenn ein unbeaufsichtigt ausführendes Programm Debugging Informationen erzeugt. „**Automatisch neu verbinden bei Verbindungsabbruch**" kann in **Einstellungen** → Optionen gesetzt werden.

Maus-Funktionen

Das Editierfenster unterstützt folgende Maus-Funktionen:

Linksklick	Cursor-Position ändern und vorherige Auswahl aufheben.
Rechtsklick	Kontextmenü **Bearbeiten** öffnen.
Links-Doppelklicken	Das Wort unter dem Cursor markieren.
Linke Maustaste drücken und ziehen	Text markieren

Alt + linke Maustaste drücken und ziehen	Textspalte markieren.
Mit linker Maustaste auf die Auswahl zeigen, drücken und ziehen.	Markierten Text verschieben.
Strg + mit linker Maustaste auf die Auswahl zeigen, drücken und ziehen.	Markierten Text kopieren.
Linksklick im linken Rand	Die ganze Zeile markieren.
Linksklick im linken Rand, drücken und ziehen	Mehrere Zeilen markieren.
Scrollrad drehen	Fenster vertikal scrollen
Klicken mit Scrollrad	Das Wort unter dem Cursor markieren.
Doppelklicken mit Scrollrad	Die ganze Zeile markieren.
Linksklick auf die Trennleiste (Splitter), drücken und ziehen	Das Fenster in mehrere Ansichten teilen oder die aktuelle Ansicht eines geteilten Fensters ausrichten.
Links-Doppelklicken auf die Trennleiste. 	Das Fenster in zwei Hälften teilen oder, wenn bereits geteilt, die Teilung wieder aufheben.

Tastatur-Funktionen

Das Editierfenster unterstützt folgende Funktionen mit Shortcuts. Beachten Sie, dass viele dieser Funktionen nur mit Tastatur-Shortcuts verfügbar sind.

Gehe zu	
Pos1	Gehe zum Zeilenanfang
Ende	Gehe zum Zeilenende
Strg + **Pos1**	Gehe zum Programmanfang
Strg + **Ende**	Gehe zum Programmende
Strg + ←	Gehe zum Wortanfang
Strg + →	Gehe zum Wortende
Strg + **Alt** + →	Gehe zum Anfang der nächsten leeren Zeile
Strg + **Alt** + ←	Gehe zum Ende der vorhergehenden leeren Zeile
Strg + **G**	Gehe zu Zeile … (Dialog öffnen)
Strg + **B**	Gehe zu Klammerpaaren (("{([" oder "})]")
F2	Gehe zum nächsten Lesezeichen
Umschalt + **F2**	Gehe zum vorherigen Lesezeichen
Strg + **F2**	Lesezeichen in der aktuellen Zeile aus-/einschalten.
Text Auswahl	
Umschalt + ←	Auswahl nach links erweitern
Umschalt + →	Auswahl nach rechts erweitern
Umschalt + ↑	Auswahl nach oben erweitern
Umschalt + ↓	Auswahl nach unten erweitern
Strg + **Umschalt** + ←	Auswahl zum Anfang des Wortes erweitern
Strg + **Umschalt** + →	Auswahl zum Ende des Wortes erweitern
Umschalt + **Pos1**	Auswahl zum Zeilenanfang erweitern
Umschalt + **Ende**	Auswahl zum Zeilenende erweitern

Umschalt + Bild↑	Auswahl um eine Seite nach oben erweitern
Umschalt + Bild↓	Auswahl um eine Seite nach unten erweitern
Strg + Umschalt + Pos1	Auswahl zum Programmanfang erweitern
Strg + Umschalt + Ende	Auswahl zum Programmende erweitern
Strg + Alt + F8	Markiert die Zeile, in der der Cursor steht.
Ausschneiden / Kopieren / Einfügen	
Strg + C	Auswahl in Zwischenspeicher kopieren
Strg + Einfg	Auswahl in Zwischenspeicher kopieren
Umschalt + Entf	Auswahl löschen und in Zwischenspeicher kopieren
Strg + X	Auswahl löschen und in Zwischenspeicher kopieren
Strg + Y	Markierte Zeile löschen und in Zwischenspeicher kopieren
Strg + V	Einfügen aus dem Zwischenspeicher
Umschalt + Einfg	Einfügen aus dem Zwischenspeicher
Strg + Alt + K	Alle Zeilen von der vorhergehenden Leerzeile bis zur nächsten Leerzeile löschen und in den Zwischenspeicher kopieren.
Suchen / Ersetzen	
Alt + F3	Suchen
Strg + F	Suchen
F3	Weitersuchen.
Umschalt + F3	Weitersuchen nach oben.
Strg + F3	Nächstes Wort wie unter dem Cursor suchen.
Strg + Umschalt + F3	Vorheriges Wort wie unter dem Cursor suchen.
Strg + R	Suchen und ersetzen.
Ändern	
Einfg	Zwischen „Einfügen“ und „Überschreiben“ hin- und herschalten.
Strg + Z	Rückgängig: Letzte Änderung
Alt + Rücktaste	Rückgängig: Letzte Änderung
Tab (mit markierten Zeilen)	Rückt die ausgewählten Zeilen ein.
Umschalt + Tab	Rückt die ausgewählten Zeilen wieder aus.
Strg + Backspace	Löscht bis zum Beginn des Wortes.
Strg + Entf	Löscht bis zum Ende des Wortes.
Strg + U	Auswahl in Kleinbuchstaben ändern.
Strg + Umschalt + U	Auswahl in Großbuchstaben ändern.
Strg + Umschalt + N	Neue Zeile oberhalb einfügen.
Strg + Alt + R	Wiederholt den nächsten Befehl mehrere Male (öffnet den Dialog).
Strg + Umschalt + R	Startet die Makro-Aufzeichnung.

Fenster scrollen	
Strg + ↑	Fenster nach oben scrollen
Strg + ↓	Fenster nach unten scrollen
Strg + Bild↑	Fenster nach links scrollen
Strg + Bild↓	Fenster nach rechts scrollen

!!! Manche der Tastatur-Funktionen hängen von spezifischen Einstellungen Ihrer Tastatur ab. Bei unerwünschten Effekten fragen Sie bitte Ihren Systemadministrator.

Funktion Rückgängig Mit **Alt** + **Rücktaste**, **Strg** + **Z** oder **Bearbeiten → Rückgängig** können Sie die letzte Aktion im Editierfenster rückgängig machen.

!!! **Datei → Speichern** und **→ Speichern als** löschen den Undo-Speicher.

Tabulatoren Benutzen Sie Tabulatoren und Einzüge um Ihr Programm visuell zu strukturieren. Die Tab-Inkremente können in **Einstellungen → Editor** individuell definiert werden.

Makro Aufzeichnen und Abrufen Häufig benutzte Befehle oder Befehlsketten können als Makro aufgezeichnet werden. Diesen Makros können Sie Funktionstasten zuordnen und dann die Befehle beliebig wiederholen. Bis zu 10 verschiedene Makros können definiert werden. Sie werden gesichert und wieder geladen, sobald APOSS-IDE das nächste Mal gestartet wird.

Drücken Sie **Strg** + **Umschalt** + **R** und das kleine Makro-Aufzeichnungsfenster zeigt den Start der Aufzeichnung. Dann geben Sie ein, was Sie aufzeichnen wollen: Das können normale Tastatureingaben, Shortcuts und Menü-Befehle sein.

Beenden Sie die Aufzeichnung indem Sie auf das schwarze Schließen-Symbol im „Makro-Aufzeichnen"-Dialog klicken. Damit wird folgendes Dialogfenster gezeigt:

Drücken Sie die Tasten für das Shortcut, das Sie für das Makro benutzen wollen (z.B. **Strg**+**Umschalt**+**M**). Jedes Shortcut kann bis zu 2 Zeichen lang sein. Dann wählen Sie eine der beiden „Sichern als" Alternativen.

!!! Sie können zum Festlegen des Makros fast alle Tasten oder Funktionstasten verwenden. Benutzen Sie aber nicht Funktionstasten oder Shortcuts, die bereits eine Funktion haben, denn dann wäre die ursprüngliche Funktion verloren und in einigen Fällen würde dies zu unvorhersehbaren Ergebnissen führen. Wenn Sie zum Beispiel die ↑-Taste belegen, können Sie diese nie mehr zum Bewegen des Cursors einsetzen!
Die **Alt**-Taste dagegen (z.B. **Alt**+**A**, **Alt**+**X** usw.) eignet sich sehr gut für Shortcuts, da die meisten Kombinationen nicht mit vordefinierten Shortcuts in Konflikt geraten.

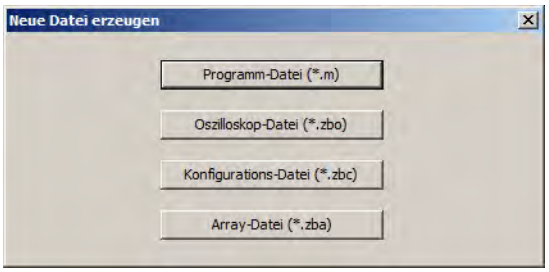
Mit **Bearbeiten → Alle Makros löschen** können alle derzeit definierten gelöscht werden.

Menü Datei

Das Menü **Datei** enthält alle notwendigen Befehle zum Erstellen, Öffnen, Speichern und Drucken eines Programms. **Datei → Öffnen, Schließen, Speichern, Speichern als, Drucken** und **Drucker einrichten** werden wie üblich benutzt. Die anderen sind nachfolgend beschrieben.

Neu

Um ein neues Programm zu schreiben, klicken Sie in der Menüleiste auf **Datei** und dann auf **→ Neu** oder nur auf die Schaltfläche 🗎. Legen Sie eine neue **→ Programm-Datei (*.m)** an:



Ein APOSS-Fenster wird mit der Bezeichnung „Unbenannt" geöffnet und Sie können beginnen, Ihr Programm zu schreiben.

Beispiel

APOSS enthält viele Programmbeispiele. Jedes dieser Programmbeispiele kann vom Anwender beliebig benutzt, verändert oder in andere Programme eingebunden werden. Mit **Datei → Beispiel** kann jedes dieser Programmbeispiele editiert werden. Im Kapitel *„Programmieren mit APOSS"* finden Sie eine Übersicht der verfügbaren Beispiele.

Speichern, Speichern als

APOSS-Programme haben immer die Datei-Erweiterung „.m".

Mit dem Toolbar-Button **→ Alle Quellcodedateien speichern** werden alle offenen Programmdateien gespeichert. Es ist nicht notwendig, eine Datei nach der anderen einzeln zu speichern.

Programmende

Die APOSS Anwendung kann mit Klicken auf **Programmende** oder auf das Schließen-Symbol rechts oben im Fenster beendet werden. Falls Sie eine neue Datei oder eine Änderung noch nicht gespeichert haben, erhalten Sie mit einer letzten Abfrage die Chance, dies zu tun.

!!!

Programmende beendet aber nicht ein laufendes Programm in der Steuerung. Ein Programm können Sie nur mit **Esc** oder mit **Entwicklung → Abbrechen** bzw. beenden. Dazu muss auch die Datei, die mit der Steuerung verbunden ist, geöffnet sein bzw. wieder geöffnet werden.

!!!

Das Trennen der Verbindung von sehr viel älteren Steuerungen (zum Beispiel bei **Datei → Programmende**), kann dazu führen, dass die Steuerung die Programmausführung stoppt, wenn von der Steuerung PRINT Befehle an den PC geschickt werden. Dies passiert, wenn der interne Print-Puffer der Steuerung voll ist. Dies ist kein Problem für alle neueren Steuerungen. Diese können die Programmausführung fortsetzen, auch wenn der Print-Puffer voll ist; die Print-Meldungen werden einfach verworfen, sobald der Puffer voll ist.

Menü Bearbeiten

Das Menü **Bearbeiten** bietet die zum Programmieren notwendigen Editierhilfen, von denen Sie die meisten auch über Tasten und Tastenkombinationen erreichen können.

Bearbeiten → Rückgänge, Ausschneiden, Kopieren und **Einfügen** werden wie üblich benutzt. Die anderen sind nachfolgend beschrieben.

Einfügen Array-Zuweisung

Mit dieser Funktion können automatisch Array-Zuweisungs-Befehle in ein APOSS Programm eingefügt werden, basierend auf Array-Werte-Token im Windows-Zwischenspeicher. Wenn zum Beispiel der Array-Name auf „list" und der Start-Index auf "1", gesetzt ist und der Zwischenspeicher die 4 Token „1 3 5 7" enthält, wird das Folgendes in das Programm eingefügt:

```
list[1] = 1
list[2] = 3
list[3] = 5
list[4] = 7
```

Beachten Sie, dass der Start-Index ein beliebiger String sein kann. Zum Beispiel könnte er auf „offset+1" gesetzt sein; das würde die Indices [offset+1], [offset+2], [offset+3] usw. erzeugen.

Der Wert-Token im Zwischenspeicher kann aus jeder anderen Windows Anwendung mit den üblichen Kopierfunktionen eingefügt werden. Zum Beispiel könnte eine Spalte mit Werten aus einer Tabellenanwendung kopiert werden.

Suchen und Ersetzen

Klicken Sie auf **Bearbeiten → Suchen** oder drücken Sie **Strg**+**F** und geben Sie im folgenden Dialogfeld den gesuchten Begriff ein. Mit **F3** können Sie dann von einer Fundstelle zur nächsten springen.

Klicken Sie auf **→ Alle Markieren** statt auf **Suchen** und es werden sofort alle Fundstellen am linken Rand mit einem blauen Dreieck markiert. Dann kann man mit **F2** die Funkstellen nacheinander aufsuchen.

Die Regulären Ausdrücke in Suchen und Ersetzen sind mit folgenden Syntaxregeln realisiert:

Wildcards	?	für beliebiges Zeichen
	.	für beliebiges Zeichen
	+	für ein oder mehrere Suchbegriffe
	*	für kein oder mehrere Zeichen
Zeichengruppe	Zeichen in eckigen Klammern werden als Gruppe gesucht; der Bereich wird mit Bindestrich angegeben, z.B. alle Zeichen von a bis c: [a-c].	
Logisches ODER	Unterausdrücke werden mit Hilfe des Pipeline-Symbols | mit ODER verknüpft.	
Unterausdrücke in Anführungszeichen	Ein regulärer Ausdruck sollte in Anführungszeichen gesetzt werden und wird als eine Einheit behandelt.	
Code-Umschaltzeichen	Abläufe wie \t, etc. werden durch ein äquivalentes einzelnes Zeichen ersetzt. \\ stellt den Backslash dar.	

Beispiele:

10	„10" suchen
10+	Suchen nach „1" gefolgt von mindestens einer „0" (z.B. 10, 100, 1000, etc.).
10*	Suchen nach „1" gefolgt von keiner oder mehreren „0" (z.B. 1, 10, 100, 1000, etc.).
vel[xyz]	Suchen nach „velx", „vely" oder „velz".

vel[1-3]	Suchen nach „vel1", „vel2" oder „vel3".
(vel)|(acc)	Suchen nach „vel" oder „acc".
vel[\t]*=	Suchen nach „vel", gefolgt von einer beliebigen Anzahl von Leerzeichen oder Tabs, gefolgt von „=" (d.h. Suchen nach Anweisungen der Variablen „vel").

Suchen in Dateien

Die Funktion **Suchen in Dateien** bietet ein Dialogfenster, um Strings (Zeichenketten) in Dateien zu finden. Dabei kann man zwischen Dateien, die gerade in APOSS geöffnet sind und Dateien auf der Festplatte wählen. Alle gefundenen Vorkommen werden in der APOSS Findbar angezeigt. Falls die **Findbar** gerade nicht offen ist, wird sie geöffnet. Doppelklick auf eine Fundstelle öffnet die Datei und positioniert den Cursor in der Zeile.

Es gibt folgende Optionen:

Suchen nach: Geben Sie den gesuchten String ein. Dieser kann Wildcards enthalten, wenn „Regulärer Ausdruck" aktiviert ist (siehe unten). In der Dropdown-Auswahlliste können zuvor gesuchte Strings ausgewählt werden.

Groß-/Kleinschreibung: Diese Option legt fest ob bei der Suche die Groß- und Kleinschreibung beachtet werden soll oder nicht.

Ganze Wörter: Mit dieser Option werden nur vollständige „Wörter" gesucht, d. h. das Zeichen vor und nach dem String muss ein nicht zum Wort gehörendes Zeichen sein, wie ein Satz- oder Leerzeichen. Dies ist äußerst nützlich bei der Suche nach Variablen, denn es vermeidet Fundstellen, bei denen der String nur ein Teil innerhalb eines längeren Strings ist.

Regulärer Ausdruck: Die aktivierte Option erlaubt beim Suchen den Einsatz von Wildcards (Platzhaltern), die wie unten in „Regulärer Ausdruck" beschrieben, interpretiert werden.

Max. Trefferzahl: Die Suche wird abgebrochen, wenn diese Anzahl von Strings gefunden wurde. Dies verhindert eine übermäßige Anzahl von Fundstellen, wodurch die **Findbar** unbrauchbar werden könnte.

Wo: Auswahl, wo die Dateien gesucht werden sollen. Bei „Verzeichnis" geben Sie das Verzeichnis ein, das gesucht werden soll (oder browsen es). Beachten Sie, dass dieses Feld auch Drag & Drop-Dateien und -Verzeichnisse unterstützt. Wurde ein Dateiname mittels Drag & Drop erstellt, enthält das Verzeichnis die durchsuchte Datei.

In Dateitypen: Liste der Dateitypen, in denen gesucht wird. Sie steht nur zur Verfügung, wenn „Verzeichnis:" ausgewählt ist. Folgende zwei Wildcard-Zeichen sind möglich:

* findet jedes Vorkommen des Zeichens
? findet jedes einzelne Zeichen.

Mehrere Dateitypen werden durch Leerzeichen getrennt. Zum Beispiel werden mit „*.m *.mi" alle Dateien mit den Erweiterungen „.m" oder „.mi" durchsucht.

Untergeordnete Ordner einbeziehen: Wenn aktiviert, werden auch Unterverzeichnisse durchsucht. Diese Option ist nur verfügbar, wenn „Verzeichnis:" ausgewählt ist.

Regulärer Ausdruck

Folgende Standard-Wildcards werden als Platzhalter unterstützt:

.	Punkt für ein beliebiges einzelnes Zeichen
?	Der voranstehende Ausdruck (Zeichen oder Ausdruck) kann einmal vorkommen, muss es aber nicht, d. h. der Ausdruck kommt null- oder einmal vor.
*	Der voranstehende Ausdruck (Zeichen oder Ausdruck) darf beliebig oft (auch keinmal) vorkommen.
+	Der voranstehende Ausdruck (Zeichen oder Ausdruck) muss mindestens einmal vorkommen, darf aber auch mehrfach vorkommen.
^	Steht für den Zeilenanfang.
$	Steht für das Zeilenende.
\	Hebt die Metabedeutung des nächsten Zeichens auf (d. h. es ist keine Wildcard).
[abc]	Mit eckigen Klammern kann eine Zeichenauswahl definiert werden. Dies kann auch ein Bereich aus einem oder mehreren Zeichen sein, wie „[a-z]" oder „[A-Za-z]", etc. Wenn das erste Zeichen ein ^ ist, gilt Übereinstimmung mit jedem Zeichen, das NICHT in der Liste ist. Wenn das ^ Zeichen selbst gesucht werden soll, darf es nicht an erster Stelle in der Liste stehen. Wenn die eckige Klammer] gesucht werden soll, muss sie als erstes in der Liste stehen. Wenn der Bindestrich – gesucht werden soll, muss er zuletzt in der Liste stehen.
|	Findet entweder den Ausdruck auf der linken oder rechten Seite.
( )	Wird zum Abgrenzen des Ausdrucks benutzt, falls notwendig.
@	(Nicht Standard) Bei den folgenden Zeichen wird die Groß- und Kleinschreibung berücksichtigt.
~	(Nicht Standard) Bei den folgenden Zeichen wird die Groß- und Kleinschreibung NICHT berücksichtigt.
!	(Nicht Standard) Das Ausrufezeichen ist reserviert und darf nicht als erstes Zeichen im Regulären Ausdruck benutzt werden. Falls es als erstes Zeichen gebraucht wird, muss ein Literal (Buchstabensymbol) vorangestellt werden (d. h. \!).

Beispiele

a.b	Findet jede Zeichensequenz die ein „a" und „b" getrennt durch genau ein Zeichen enthält (z. B. „axb", „a&b", etc.).
a\.b	Findet jede Zeichensequenz, die den String „a.b" enthält.
ab*c	Findet jede Zeichensequenz, die mit „a" beginnt, eine beliebige Anzahl von „b" enthält und mit „c" endet (z. B. „ac", „abc", „abbc" etc.).
^abc.*xyz$	Findet alle Zeilen, die mit „abc" beginnen und mit „xyz" enden. Beliebig viele andere Zeichen können dazwischen sein.
a[0-9]+	Findet jede Sequenz, die mit „a" beginnt, gefolgt von einer oder mehreren Ziffern (z. B. „a1", „a999" etc.).
abc|xyz	Findet jede Sequenz, die „abc" oder „xyz" enthält.
(abc|xyz)[0-9]+	Findet jede Sequenz, die „abc" oder „xyz" enthält und anschließend eine oder mehrere Ziffern (z. B. „abc1", „xyz99", etc.). Beachten Sie, dass „abc|xyz[0-9]+" „abc" finden würde und „xyz" mit nachfolgenden Ziffern, aber nicht „abc" mit nachfolgenden Ziffern.

Lesezeichen Mit Lesezeichen kann der Anwender bestimmte für ihn interessante Zeilen markieren und dann schnell zwischen diesen zu springen.

Wenn Lesezeichen im Editor gesetzt wurden, werden diese mit der Programmdatei gespeichert und mit dieser wieder geladen und gesetzt.

Nächstes Lesezeichen [F2] Wenn Lesezeichen im Editor gesetzt wurden, dann scrollt → **Nächstes Lesezeichen** oder **F2** durch das Programm und positioniert den Cursor auf die nächste Zeile, die ein Lesezeichen enthält.

Vorheriges Lesezeichen Wenn Lesezeichen im Editor gesetzt wurden, dann positioniert → **Vorheriges Lesezeichen** oder **Umschalt+F2** den Cursor auf die vorherige Zeile, die ein Lesezeichen enthält.

Lesezeichen ein/aus Diese Funktion oder **Strg+F2** schaltet das Lesezeichen in der aktuellen Zeile ein- bzw. aus. Wenn in dieser Zeile kein Lesezeichen ist, wird eines gesetzt. Falls die Zeile bereits ein Lesezeichen enthält, wird es gelöscht.

Alle Lesezeichen Löschen Klicken Sie auf **Bearbeiten → Alle Lesezeichen löschen** um alle existierenden Lesezeichen im Editor zu löschen.

Tab umwandeln Klicken Sie auf → **Tab. Umwandeln** und alle Tab-Zeichen werden während des Editierens in Leerzeichen umgewandelt.

Dazu wird der Tab-Wert benutzt, der in **Einstellungen → Editor** gesetzt ist.

Alle Makros löschen Diese Menü-Funktion löscht alle Makros im Editor und deren Shortcuts, die mit **Strg+Umschalt+R** erzeugt wurden. Siehe auch „Makro aufzeichnen und abrufen".

Tastatur Mit dieser Funktion können Sie Zeichen aus anderen Sprachen in Ihr Programm einfügen.

Menü Entwicklung

Dieses Menü bietet verschiedene Funktionen für die Phase der Entwicklung von Anwendungsprogrammen. Dies beinhaltet Funktionen zum Kompilieren, Ausführen, Abbrechen und für die Fehlersuche. Es enthält außerdem Funktionen, um Steuerungen zu verbinden und Schnittstellen zu schließen.

Viele dieser Funktionen erfordern, dass eine Steuerung verbunden ist, bevor sie ausgeführt werden können. Eine Steuerung kann entweder mit **Esc** mit der Default-Steuerung verbunden werden, oder mit **Entwicklung → Steuerung auswählen** mit einer bestimmten Steuerung, wenn mehrere Steuerungen vorhanden sind. Falls keine Steuerung verbunden ist, wenn eine Funktion ausgeführt wird, werden die meisten Funktionen automatisch eine Verbindung zur Default-Steuerung herstellen.

Informationen zu Debugging-Programmen finden Sie in **Programme Debuggen** am Ende dieses Kapitels.

Ausführen F5

Es wird das Programm gestartet, das geöffnet und im Editor dargestellt ist. Dies beinhaltet folgende Schritte:

1. Falls aktuell keine Steuerung offen und mit dem APOSS-Fenster verbunden ist, wird versucht, eine Verbindung mit der als Default definierten Steuerung herzustellen. Ist keine Steuerung angeschlossen, wird **Ausführen** abgebrochen.
2. Es wird geprüft, ob eine Steuerung bereits ein Programm ausführt. Falls dies der Fall ist, wird der Anwender gefragt, ob das existierende Programm gestoppt werden kann, da die Steuerung nur ein Programm zur gleichen Zeit ausführen kann. Wenn die Steuerung nicht im Leerlauf ist, wird **Ausführen** abgebrochen.
3. Wenn der Anwender während des Editierens Änderungen im Programm vorgenommen hat, wurden diese automatisch auf der PC-Festplatte gesichert. Falls dies ein „neues" Programm ist, kann man jetzt einen Dateinamen für das Programm eingeben. Es ist aber zu diesem Zeitpunkt nicht notwendig, denn ohne Dateinamen wird eine temporäre Datei benutzt.
4. Das editierte Programm wird dann kompiliert und eine für die eingesetzte Steuerung passende Version des Programms mit Maschinencode erzeugt. Beachten Sie, dass der „Maschinencode" nicht editiert werden kann. Falls das Kompilieren aus irgendeinem Grund fehlschlägt, wird **Ausführen** abgebrochen.
5. Dieser Maschinencode wird dann in den temporären Speicher der Steuerung heruntergeladen. Beachten Sie, dass dieser temporäre Speicher verloren geht, sobald die Steuerung ausgeschaltet wird. Um ein Programm dauerhaft in der Steuerung zu sichern, benutzen Sie **Steuerung → Programme**.
6. Sobald der Download beendet ist, wird das Programm in der Steuerung ausgeführt.

Jedes Mal wenn **Ausführen** benutzt wird, wird das zuvor heruntergeladene Programm mit dem erneut heruntergeladenen überschrieben. Dies erlaubt es, sehr schnell und leicht Änderungen vorzunehmen und wieder zu testen.

Programme in mehreren Steuerungen ausführen

Sie können ganz einfach verschiedene Programme in verschiedenen Steuerungen zur gleichen Zeit ausführen: Dazu öffnen Sie jedes Programm in einem anderen APOSS Fenster und wählen in jedem Fenster mit **Entwicklung → Steuerung auswählen** die gewünschte Steuerung für dieses Programm aus. Dann starten Sie mit **Entwicklung → Ausführen** in jedem Fenster den Download und das Programm.

Beachten Sie, dass Sie zuerst eine Steuerung **Auswählen** müssen, bevor Sie **Ausführen** starten, weil **Ausführen** immer die Default-Steuerung öffnet, wenn keine Steuerung ausgewählt ist.

Jedes APOSS Fenster kann zur gleichen Zeit mit nur einer Steuerung verbunden sein und jedes APOSS Programm kann zur gleichen Zeit nur in einem APOSS Fenster editiert werden. Falls Sie also das gleiche Programm in mehreren Steuerungen ausführen wollen, können Sie eine der folgenden Methoden nutzen:

1. Verbinden Sie mit **Entwicklung → Steuerung auswählen** die erste Steuerung. Dann downloaden Sie mit **Entwicklung → Ausführen** das Programm und starten die Programmausführung in der Steuerung. Sobald es ausgeführt wird, verbinden Sie mit **Entwicklung → Steuerung auswählen** die zweite Steuerung. Das trennt zwar die Verbindung mit der ersten Steuerung, lässt aber das Programm weiter laufen. Downloaden Sie mit **Entwicklung → Ausführen** wieder das Programm und starten Sie die Programmausführung. Wiederholen Sie diese Vorgehensweise für jede der Steuerungen, in denen das Programm laufen soll.
2. Falls das Programm in mehreren Steuerungen laufen soll, die alle im gleichen Netzwerk sind (zum Beispiel in einem CAN Netzwerk), können Sie **Entwicklung → Download alle** nutzen. Dann können die oben genannten Schritte mit einem einzigen Befehl ausgeführt werden.

Wenn Sie das Programm in mehrere Steuerungen laden wollen, verbinden Sie das Programm mit der jeweiligen Steuerung und klicken auf → **Ausführen**.

Abbrechen [Esc] und Fortsetzen

Klicken Sie auf **Entwicklung → Abbrechen** oder drücken Sie **Esc** um alle Programme, die in der Steuerung ausgeführt werden, abzubrechen.

!!!

Falls eine Programmausführung abgebrochen wird, während sich der Antrieb dreht, wird der Antrieb mit der maximal zulässigen Verzögerung abgebremst.

Falls das Programm in mehreren Steuerungen gleichzeitig läuft, können Sie mit **Entwicklung → Abbruch alle** laufende Programme abbrechen.

Klicken Sie auf **Entwicklung → Fortsetzen**, um das eben abgebrochene Programm fortzusetzen. Dabei werden auch die unterbrochenen Positioniervorgänge zu Ende ausgeführt.

Wenn ein Programm mit einer Fehlermeldung abgebrochen wurde, können Sie es – nachdem Sie den Fehler behoben und/oder die Fehlermeldung gelöscht haben – mit dieser Funktion wieder → **Fortsetzen**.

Meldungen -> Log-Datei

Mit dieser Funktion starten Sie die Protokollierung aller Meldungen, die im Kommunikations-Fenster dargestellt werden, in eine Datei; mit **Log-Datei beenden** wird sie beendet.

Die Meldungen werden gecached und nicht immer sofort in die Datei geschrieben. Wenn also die Datei bearbeitet oder kopiert wird, während noch die Protokollierung läuft, werden die allerletzten Meldungen nicht immer enthalten sein. Sie können dies vermeiden, wenn Sie unmittelbar bevor dem Kopieren oder Bearbeiten die → **Log-Datei aktualisieren.** Dann werden auch alle zwischengespeicherten Meldungen bis zu diesem Zeitpunkt in die Datei geschrieben.

Beachten Sie, dass **Log-Datei beenden** und **Log-Datei aktualisieren** erst aktiviert werden, wenn die Protokollierung gestartet wurde.

Debug Befehle

Die folgenden Befehle wurden entwickelt, um den Anwender bei der Fehlersuche, also beim Debuggen neu entwickelter Programme zu unterstützen. Detaillierte Information über Debuggen und den Gebrauch der Befehle finden Sie in **Programme debuggen** am Ende dieses Kapitels.

Debugger starten

Dieser Befehle bereitet sowohl die APOSS-IDE als auch die Steuerung für das Debuggen vor. Dies beinhaltet das Kompilieren des Programms im „Debug"-Modus, das Einfügen von Breakpoints und das Downloaden des Programms.

Gehe zu Breakpoint

Dieser Befehl startet die Ausführung des Programms in der aktuellen Programmzeile und führt sie bis zum nächsten vom Anwender gesetzten Breakpoint fort.

Einzelschritt [F9]

Dieser Befehl führt eine einzelne Programmzeile aus und stoppt vor der nächsten Zeile. Die nächste Zeile wird nicht ausgeführt.

Debugger stoppen

Mit diesem Befehl wird der Debug-Modus beendet, und zwar sowohl in der APOSS-IDE als auch in der Steuerung.

Breakpoints entfernen

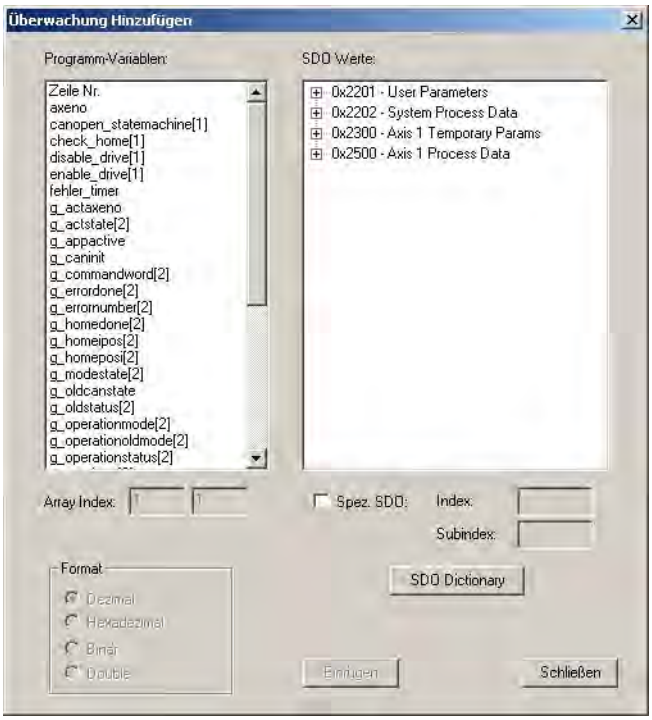
Dieser Befehl entfernt alle „Benutzer"-Breaktpoints.

Überwachung hinzufügen / starten / stoppen

Diese Funktion aktiviert die Online-Überwachung der Variablen, Arrays, System- und Achsprozessdaten und der Achsparameter.

Alle gerade überwachten Werte werden im **Überwachungsfenster** angezeigt. Um neue Werte dieser Liste hinzuzufügen, wählen Sie die gewünschten Werte mit **Entwicklung → Überwachung hinzufügen** aus und klicken auf → **Einfügen**. Zum Löschen vorhandener Werte klicken Sie auf den Wert im **Überwachungsfenster** und drücken die **Entf**-Taste oder löschen mit Rechtsklick und Auswahl im Kontextmenü.

Durch Rechtsklick auf einen Eintrag in der Liste und der Auswahl von → **Wert zuweisen** im Kontextmenü können Werte die Steuerung (wenn eine Steuerung verbunden ist) geschrieben werden. Die Überwachung muss nicht aktiv sein, um Werte zur Steuerung zu schreiben. „**Wert zuweisen**" wird nur aktiviert, wenn eine Steuerung aktuell verbunden ist.



Auf der linken Seite werden alle Programmvariablen und Arrays dargestellt, auf der rechten Seite alle Systemwerte (in einem Verzeichnisbaum). Um einen Wert zum Überwachungsfenster hinzuzufügen, markieren Sie den Wert, wählen dazu das Format, das für die Darstellung benutzt werden soll und klicken auf → **Einfügen**. Sie können Werte auch einfach durch Doppelklicken hinzufügen. Sie können mehrere Werte einfügen, bevor Sie das Dialogfenster → **schließen**.

Wenn ein Array-Wert zum Überwachungsfenster hinzugefügt wird, dann müssen die Array-Indices spezifiziert werden. Diese Felder sind deaktiviert, wenn der ausgewählte Wert nicht zu einem Array gehört. Beachten Sie, es können nur die ersten 250 Elemente eines Arrays überwacht werden.

Aktivieren und stoppen Sie das Monitoring mit → **Überwachung starten**, → **Überwachung stoppen** oder klicken Sie auf  .

!!! Wenn die Überwachung aktiviert ist, wird das Überwachungsfenster ständig aktualisiert. Dies belegt sowohl Ressourcen der Steuerung als auch des Netzwerks. Daher sollte in die Anzahl der überwachten Werte auf ein vernünftiges Maß begrenzt werden (im Allgemeinen nicht mehr als 10-15 Werte, abhängig von der Netzwerk-Geschwindigkeit). Wenn es notwendig ist, mehr Werte zu überwachen, dann sollte das **Oszilloskop** benutzt werden.

Syntaxprüfung F4

Entwicklung → Syntaxprüfung erzeugt eine „Test-Kompilierung" des Programms, das gerade editiert wird. Das kann während einer neuen Programmierung oder Änderung eines existierenden Programms sehr hilfreich sein. Es ist ein schneller Weg, Syntax-Fehler im Programm zu finden ohne das Programm in die Steuerung downloaden und ausführen zu müssen. Wenn ein Syntaxfehler gefunden wird, wird die Zeilennummer und eine Fehlerbeschreibung im Kommunikationsfenster ausgegeben und der Cursor automatisch an die Position mit dem Syntaxfehler gestellt.

Die **Syntaxprüfung** erzeugt eine zusätzliche Debug-Datei zur Prüfung der Syntax. Diese Datei erhält den gleichen Namen wie das Programm, jedoch mit der Extension „.ad$".

In Datei kompilieren

Der Befehl **Ausführen[F5]** kompiliert ein Programm in eine „maschinenlesbare" binäre Version, die in die Steuerung geladen und ausgeführt wird. Der Befehl **In Datei kompilieren** ist ähnlich, außer dass die kompilierte binäre Version nicht heruntergeladen und ausgeführt wird. Stattdessen wird die binäre Version als „.bin"-Datei auf der PC-Festplatte gespeichert. Binäre „.bin"-Dateien werden mit **Steuerung → Programme** verwaltet.

Beim → **In Datei kompilieren** bietet ein Dialogfeld die Möglichkeit, genau die Steuerung zu bestimmen, für die das Programm kompiliert werden soll. Beachten Sie, dass die kompilierten binären Versionen hardwareabhängig sind und für die Hardware kompiliert werden müssen, in welcher sie ausgeführt werden. Danach kann in einem **Sichern als** Dialog der Dateiname zum Speichern der Datei ausgewählt werden. Als Default wird der Programmname mit der Erweiterung „.bin" benutzt. Erst danach wird das Programm kompiliert und auf der Festplatte gesichert.

!!! Diese Funktion ist nur verfügbar, wenn **Binär-Datei Unterstützung** in **Einstellungen** → Optionen aktiviert ist.

Abbruch alle

Falls das Programm in mehreren Steuerungen läuft, die alle im gleichen Netzwerk hängen, klicken Sie auf **Entwicklung → Abbruch alle**, um alle laufenden Programme abzubrechen.

Das heißt, diese Funktion bricht nur die Programmausführung ab, die in Steuerungen laufen, die das gleiche Anschluss-Interface haben wie die Steuerung, die mit diesem APOSS Fenster verbunden ist (z.B. mehrere Steuerungen an einem CAN-Bus). Steuerungen, die andere Anschluss-Interfaces benutzen, werden nicht abgebrochen. Wäre zum Beispiel das APOSS-Fenster mit einem CAN-LPT verbunden, würden alle Steuerungen, die USB nutzen weiter laufen.

!!! Beachten Sie, dass die Programme auch in den Steuerungen abgebrochen werden, die aktuell nicht mit einem APOSS-Fenster verbunden sind, solange die Steuerung das gleiche Anschluss-Interface im originalen ID-Scanbereich nutzt. Wären zum Beispiel die Steuerungen 1 und 2 beide in einem CAN Netzwerk, aber APOSS ist gerade nur mit Steuerung 1 verbunden, dann würde **Abbruch alle** auch das Programm in Steuerung 2 abbrechen.

!!! Falls eine Programmausführung abgebrochen wird, während sich der Antrieb dreht, wird der Antrieb mit der maximal zulässigen Verzögerung abgebremst.

Start alle

Die Funktion → **Start alle** sendet einen 'Ausführen' Befehl an alle angeschlossenen Steuerungen, die das gleiche Anschluss-Interface nutzen, wie die Steuerung, die mit dem APOSS-Fenster verbunden ist (z.B. mehrere Steuerungen in einem CAN-Netzwerk). Wenn Sie Programme testen, die in mehreren Steuerungen in einem Netzwerk laufen, ist dies eine schnelle Möglichkeit, alle Steuerungen zur gleichen Zeit zu starten.

Download alle

Entwicklung → Download alle öffnet den **APOSS Download-Modus** Dialog, mit dem die gerade editierte .m-Datei in mehrere Steuerungen heruntergeladen werden kann.

Mehr Details und welche Optionen möglich sind, lesen Sie bitte im Abschnitt **Download → Programme**.

Alle Programme löschen

Entwicklung → Alle Programme löschen löscht alle Programme in allen Steuerungen, die die gleiche Schnittstelle benutzten, wie die Steuerung, die mit diesem APOSS-Fenster verbunden ist (d.h. mehrere Steuerungen an einem CAN-Bus). Verbundene Steuerungen, die andere Schnittstellen benutzen, sind nicht betroffen.

!!! Beachten Sie, dass die Programme auch in den Steuerungen gelöscht werden, die gerade nicht mit einem APOSS-Fenster verbunden sind. Solange wie die Steuerung die gleiche Schnittstelle benutzt und innerhalb des ID-Scanbereichs ist, werden die Programme gelöscht. Wenn zum Beispiel die beiden Steuerungen 1 und 2 in einem CAN-Netz sind, aber APOSS gerade nur mit der Steuerung 1 verbunden ist, wird → Alle Programme löschen die Programme in beiden Steuerung 1 und 2 löschen.

Dieser Befehl wird in Verbindung mit Entwicklung → Download alle benutzt.

Steuerung auswählen

Wenn mehr als eine Steuerung vom PC erreichbar ist, kann mit → Steuerung auswählen, die spezifische Steuerung gewählt werden, die mit dem APOSS-Fenster verbunden werden soll. Alle aktuell verfügbaren Steuerungen werden in einem Verzeichnisbaum dargestellt. Markieren Sie die gewünschte Steuerung und klicken Sie auf OK um die Steuerung zu verbinden.

Falls keine Steuerungen gezeigt werden oder die gewünschte Schnittstelle nicht vorhanden ist, wählen Sie diese im Popup-Menü aus und klicken auf → Schnittstelle öffnen. Dann können Sie das gewünschte Interface auswählen.

Beachten Sie, wenn Sie hier eine Schnittstelle auswählen, ändert das nicht die Default-Schnittstelle, die mit Einstellungen → Schnittstelle festgelegt wurde.

Wenn eine Steuerung schon mit einem APOSS-Fenster verbunden ist, kann mit Entwicklung → Steuerung auswählen das APOSS-Fenster mit einer anderen Steuerung verbunden werden. Benutzen Sie dazu einfach den Dialog Steuerung auswählen. Die vorhandene Verbindung wird dann geschlossen und eine Verbindung zur neuen Steuerung geöffnet.

Weil jede spezifische Steuerung zur gleichen Zeit mit nur einem APOSS-Fenster verbunden sein kann, führt das Auswählen einer Steuerung, die schon mit einem anderen APOSS-Fenster verbunden ist, dazu, dass die Steuerung vom anderen APOSS-Fenster zu diesem APOSS-Fenster wechselt.

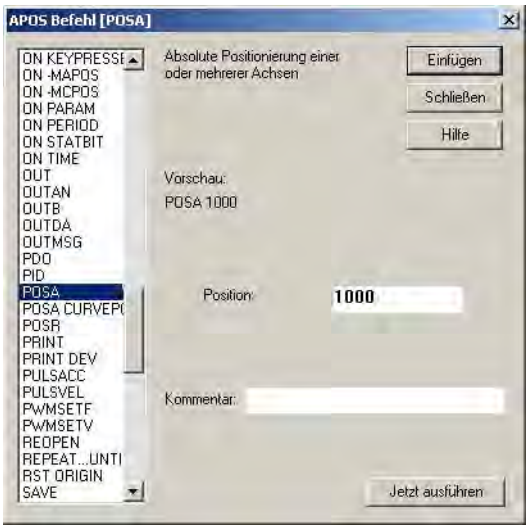
Schnittstelle schließen / Alle Schnittstellen schließen

Wenn eine Steuerung mit einem APOSS-Fenster verbunden ist, können Sie mit dieser Funktion die Verbindungen zu allen Steuerungen schließen, die das gleiche Anschluss-Interface nutzen (z.B. alle Steuerungen an einem CAN-Bus). Eine Meldung zeigt an, welche Verbindungen beendet werden.

Mit **Alle Schnittstellen schließen** werden alle Steuerungen an allen Anschluss-Interfaces geschlossen.

Befehlshilfe

Die Befehlshilfe zeigt alle Befehle mit der entsprechenden Syntax, die Sie ganz einfach in Ihr Programm → **Einfügen** oder direkt ausführen können. Besonders für selten benutzte Befehle kann dies ganz hilfreich sein. Sie erhalten ausführliche Informationen zu einem markierten Befehl, wenn Sie auf → **Hilfe** klicken oder **F1** drücken.



Zum Beispiel POSA: Geben Sie die Position in das Feld für die Achse ein. Die Vorschau zeigt Ihnen die genaue Syntax des Befehls.

Einfügen oder Jetzt ausführen

Stellen Sie den Cursor im Editierfenster an die Stelle, wo Sie ein oder mehrere Befehle einfügen wollen, wählen Sie **Entwicklung → Befehlshilfe** und hier den Befehl aus, zum Beispiel POSA, ergänzen die notwendigen Parameterwerte und ggf. einen Kommentar und klicken Sie auf → **Einfügen**.

Oder klicken Sie auf **Jetzt ausführen** und testen diesen Befehl, bevor Sie ihn in Ihr Programm → **Einfügen**.

!!! Abhängig vom Befehl können freigegebene Antriebe anlaufen, stoppen etc.

Menü Steuerung

Dieses Menü bietet Funktionen um die Konfiguration der Steuerung anzusehen und einzustellen sowie den Speicher in der Steuerung zu organisieren.

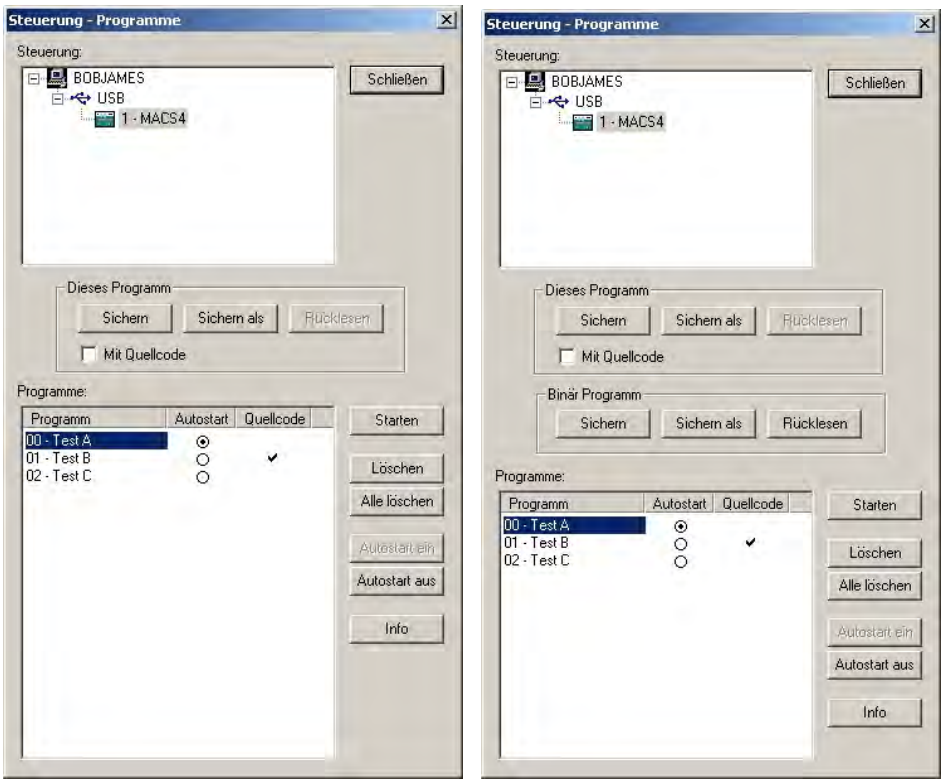
Dazu gehört: Verwalten der gespeicherten Programme (und markieren der Programme für den **Autostart**), prüfen und setzen der globalen und Achsparameter sowie sichern und zurückzusetzen des EPROM-Speichers der Steuerung.

Programme

Klicken Sie auf **Steuerung** → **Programme** und das Dialogfeld zeigt alle angeschlossenen Steuerungen. Es ist die Steuerung markiert, mit der das Programm im Editierfenster gerade verbunden ist. Sie können natürlich auch eine andere Steuerung markieren und bearbeiten.

!!!

Wenn **Binärdatei-Unterstützung** in **Einstellungen** → **Optionen** aktiviert ist, sieht das Dialogfeld wie rechts gezeigt aus.



Unter **Programme:** ist eine Liste aller Programme, die derzeit in der Steuerung gesichert sind. Jedes Programm wird mit einer Programmnummer sowie einen Programmnamen bezeichnet. Abhängig vom verfügbaren Speicher und der Größe der gesicherten Programme können bis zu 91 Programme zur gleichen Zeit gesichert werden. Das Symbol ⊙ in der Autostart-Spalte zeigt, dass das Programm für den **Autostart** gekennzeichnet wurde. Das Häkchen ✓ zeigt, dass das Programm **mit Quellcode** in der Steuerung gesichert ist.

Dieses Programm sichern

Immer wenn Sie ein Programm mit **Entwicklung** → **Ausführen [F5]** wird es kompiliert und in einen temporären Speicherbereich der Steuerung heruntergeladen. Dieser wird mit jedem folgenden **Ausführen** überschrieben und geht verloren, sobald die Steuerung ausgeschaltet wird. Mit **Steuerung** → **Programme** können Sie jedoch das aktuelle Programm aus dem APOSS-Fenster permanent → **Sichern**. Das Programm wird kompiliert und in einen permanenten Speicherbereich der Steuerung heruntergeladen. Das neue Programm wird an das Ende der Liste der vorhandenen Programme gehängt.

!!!

Beachten Sie, dass **Dieses Programm** → **Sichern** nicht automatisch die originale Quelldatei des Programms sichert. Sie sollten daher immer mit dem normalen Befehl die **Datei** auch auf der Festplatte des PCs → **Speichern**.

Klicken Sie auf → **Sichern** und geben Sie im folgenden Dialogfeld einen Namen ein oder bestätigen Sie den vorgeschlagenen Dateinamen. Die Programmnummer wird automatisch vergeben.

In Steuerungen mit Firmware Version 6.01.10 oder höher werden die Namen in UTF-8 Kodierung gesichert. Dies erlaubt jedes beliebige Zeichen im Namen, sogar Zeichen aus anderen Sprachen. Die Namen dürfen jedoch nicht länger als 8 Bytes sein (wohlgemerkt: einige UTF-8 Zeichen erfordern mehr als ein Byte). In Steuerungen mit Versionen älter als 6.01.10 sind die Namen auf ANSI Zeichen im Bereich von 0x20 bis 0x5F beschränkt (Akzentbuchstaben dürfen nicht benutzt werden) und der Name darf nicht länger als 8 Zeichen sein.

Klicken Sie auf → **Sichern als** und Sie können zusätzlich zum Namen auch die Programmnummer (0 bis 90) selbst bestimmen.

Sichern mit Quellcode

Wenn die Checkbox aktiviert ist, wird zusätzlich zur kompilierten und direkt ausführbaren Programmdatei der Quellcode in der Steuerung gesichert. Sie können diesen dann später wieder aus der Steuerung auslesen (upload) und auf dem PC erneut bearbeiten.

Wenn ein Programm-Quellcode heruntergeladen wird, werden eingebundene Dateien (Include-Dateien), die der Quellcode benutzt, auch mit dem Quellcode heruntergeladen. Dies ermöglicht es, statt Programmteile das komplette Programm in der Steuerung zu speichern.

Alle mit Quellcode gesicherten Programme erhalten in der Spalte Quellcode ein ✓ Häkchen.

!!!

Sollte in der Steuerung nicht genügend Speicher zur Verfügung stehen, um den Quellcode zu speichern, erfolgt eine Meldung und Sie müssen erst andere Programme löschen, bevor das neue Programm gespeichert werden kann.

Dieses Programm Rücklesen

Für alle mit dem Häkchen ✓ gekennzeichneten Programme ist der dazugehörende Quellcode in der Steuerung gespeichert. Dieser Quellcode kann zur weiteren Verwendung aus der Steuerung zurück auf den PC gelesen werden.

Wählen Sie das gewünschte Programm aus und klicken Sie auf **Dieses Programm → Rücklesen**. Es wird ein Default-Programmname generiert, der Datum und Uhrzeit enthält, zu der der Original-Quellcode in der Steuerung gespeichert wurde. Damit werden Probleme vermieden, die durch das Überschreiben von vorhandenen Dateien auf dem PC entstehen könnten.

Binär Programm Sichern oder Rücklesen

Die Funktion wird nur dargestellt, wenn **Binär Datei Unterstützung** im Dialogfeld **Einstellungen → Optionen** ausgewählt ist.

Wählen Sie das gewünschte Programm aus und klicken Sie auf **Sichern** oder **Sichern als** und wählen Sie im folgenden Dialogfeld die kompilierte .bin Datei aus. Geben Sie einen Programmnamen und eine Nummer ein. Sie wird dann in die Steuerung heruntergeladen und dort als binäre Datei gesichert.

Klicken Sie auf **Rücklesen** und das gerade ausgewählte binäre Programm wird aus der Steuerung ausgelesen.

!!!

Steuerungen mit einer älteren Version als 6.1.15 unterstützen nicht das Rücklesen der binären Dateien; in diesem Fall ist die **Rücklesen**-Schaltfläche nicht auswählbar.

Programme Starten

Sie können im Programm-Dialogfenster ein beliebiges gespeichertes Programm auswählen und direkt → **Starten**.

Autostart

Um den Einsatz der Steuerung als Stand-alone oder schlüsselfertige Anwendung zu unterstützen (d.h. ohne Eingaben des Anwenders), kann die Steuerung so konfiguriert werden, dass ein APOSS-Programm automatisch startet, wenn die Steuerung eingeschaltet wird. Typischerweise darf die Steuerung bei diesen Anwendungen auch nie gestoppt werden. Dann kann die Steuerung so konfiguriert werden, dass ein „Restart"-APOSS-Programm startet, wenn das ausgeführte Programm abbricht.

Der erste Teil der Autostart-Prozedur ist das **Autostart-Programm** (sofern eines definiert ist). Es wird normalerweise immer automatisch gestartet wenn die Steuerung eingeschaltet wird. Ist kein Autostart-Programm definiert, dann wird beim Einschalten kein Programm gestartet.

Sobald die Steuerung eingeschaltet wird, laufen verschiedene Selbsttest-Routinen ab. Wenn einer dieser Tests fehlschlägt, wird das Autostart-Programm nicht gestartet.

Der zweite Teil der Autostart-Prozedur ist das **Restart-Programm**. Es wird mit einigen Ausnahmen automatisch gestartet, wenn das derzeit ausgeführte Programm abbricht; dies beinhaltet insbesondere auch den Abbruch des Autostart-Programms.

!!! Wenn das **Autostart-Programm**, das **Restart-Programm** oder irgendein anderes Programm durch den Anwender abgebrochen wird, dann wird der Autostart-Mechanismus deaktiviert. Das Autostart-Programm oder ein Restart-Programm wird nicht automatisch wieder gestartet bis die Steuerung aus- und wieder eingeschaltet wird.

!!! Ein Programm, das als Autostart-Programm gekennzeichnet ist, kann immer auch manuell durch den Anwender gestartet werden. In diesem Fall wird das Programm jedoch als normales Programm ausgeführt und nicht als ein Autostart-Programm, und es wird auch kein Restart-Programm gestartet, wenn das Programm abbricht.

Autostart Programm

Die Steuerung behandelt das Autostart-Programm wie folgt:

1. Wenn ein Programm explizit als „Autostart-Programm" gekennzeichnet wurde (siehe oben), dann wird dieses Programm gestartet, sobald die Steuerung eingeschaltet wird.
2. Wenn I_PRGSTART (103) auf einen digitalen Eingang gesetzt ist, bestimmt I_PRGCHOICE (104) die Programmnummer des Autostart-Programms. Die Steuerung wartet bis der I_PRGSTART Eingang aktiviert wird und startet dann das Autostart-Programm. Beachten Sie, dass der Eingang schon aktiviert sein kann, wenn die Steuerung eingeschaltet wird. Diese Parameter sind dafür vorgesehen, um das Autostart-Programm von externen Quellen (z.B. eine SPS) auswählen zu können.
3. Oder es gibt kein Autostart-Programm.

Jedes gespeicherte Programm kann durch Auswahl und klicken auf → **Autostart ein** als Autostart-Programm bestimmt werden. Es wird dann in der Spalte mit einem ⦿ gekennzeichnet. Diese Zuweisung kann einfach durch erneute Auswahl und klicken auf → **Autostart aus** entfernt werden. Zur gleichen Zeit kann nur ein Autostart-Programm bestimmt sein. Wird also ein zweites Programm als Autostart-Programm ausgewählt, wird die Markierung für das vorhergehende entfernt.

Eine der Hauptaufgabe von Autostart-Programmen ist, in Anwendungen die verschiedene Programme in verschiedenen Situationen erfordern, auszuwählen, welches von mehreren Programmen ausgeführt werden soll. Das Autostart-Programm identifiziert (gewöhnlich entweder durch Konfigurationsparameter oder durch Setzen von digitalen Eingängen) welches Programm ausgeführt werden soll. Es bestimmt dann dieses Programm als das Restart-Programm und bricht ab. Danach startet die Steuerung automatisch das ausgewählte Programm.

!!! Die I_PRGSTART Prozedur funktioniert in einigen Steuerungen mit Firmware vor 6.7.19 nicht korrekt. Wenn dies die Anwendung tangiert, kontaktieren Sie bitte die zub machine control AG wegen eines Updates der Firmware.

Restart Programm

Wenn die Steuerung beim Einschalten ein Autostart-Programm ausführt, wird die Autostart-Prozedur aktiviert. Ist diese aktiv, startet die Steuerung automatisch das gewünschte Restart-Programm, wann immer das gerade ausgeführte Programm abbricht. Dies wird solange fortgeführt, bis die Autostart-Prozedur deaktiviert wird.

Die Autostart-Prozedur wird deaktiviert, wenn eines der folgenden Ereignisse eintritt:

1. Der Anwender hat einen „Abbruch"-Befehl ausgeführt (d.h. ein **Esc**).

2. Ein Programm wurde mit einem Fehler beendet (einem anderen als einen der unten aufgeführten).

Wenn die Autostart-Prozedur deaktiviert wurde kann sie nicht wieder durch Aus- und Wiedereinschalten der Steuerung reaktiviert werden.

Die Autostart-Prozedur wird nicht deaktiviert, wenn einer der folgenden Fehler auftritt:

1. Schleppfehler (Fehler-Nr. 8)
2. Endschalter erreicht (Fehler-Nr. 25)
3. Software Endschalter überschritten (Fehler-Nr. 11)

Die Steuerung bestimmt, welches Programm das Restart-Programm ist wie folgt:

1. Wenn PRGPAR (102) gesetzt ist (d.h. nicht -1), ist dies die Programmnummer des Restart-Programms.
2. Oder wenn I_PRGSTART (103) auf einen digitalen Eingang gesetzt ist: dann wartet die Steuerung bis dieser Eingang aktiviert wird. Sobald aktiviert, bestimmt I_PRGCHOICE (104) die Programmnummer des Restart-Programms.
   Beachten Sie, dass der Eingang nicht schon aktiviert sein darf, wenn das gerade laufende Programm abbricht; die Aktivierung wird ausschließlich durch eine steigende Flanke getriggert.
3. Oder wenn das Autostart-Programm das einzige Programm ist, das ausgeführt werden soll; dann ist das Autostart-Programm auch das Restart-Programm (d.h. das Autostart-Programm wird wiederholend ausgeführt).
4. Oder es gibt kein Restart-Programm und die Steuerung startet überhaupt kein Programm.

Der Parameter PRGPAR ist für die „Verkettung" von Programmen vorgesehen. Dazu setzt das gerade ausgeführte Programm einfach nur den Parameter, um festzulegen welches Programm als nächstes ausgeführt werden soll.
Beachten Sie, dass falls PRGPAR nicht zurückgesetzt wird, das laufende Programm wiederholt ausgeführt wird.

Die Parameter I_PRGSTART und I_PRGCHOICE sind dafür vorgesehen, das nächste Programm das ausgeführt werden soll von externen Quellen (wie z.B. eine SPS) auswählen zu können.

!!! Das ursprünglich bestimmte Autostart-Programm wird nur dann wieder gestartet, wenn PRGPAR und I_PRGSTART nicht benutzt wurden. Wurden entweder PRGPAR oder I_PRGSTART benutzt, wird das Autostart-Programm nur einmal ausgeführt, wenn die Steuerung eingeschaltet wird. Diese „einmalige" Ausführung kann für ausführende Funktionen wie HOME nützlich sein
Beachten Sie, dass der Parameter PRGPAR benutzt werden kann, um explizit das Autostart-Programm wieder zu starten.

!!! Die I_PRGSTART Prozedur funktioniert in einigen Steuerungen mit Firmware vor 6.7.19 nicht korrekt. Wenn dies die Anwendung tangiert, kontaktieren Sie bitte die zub machine control AG wegen eines Updates der Firmware.

Programme löschen Wählen Sie ein Programm aus und klicken sie auf **Löschen,** wenn Sie ein bestimmtes Programm in der Steuerung löschen wollen. Damit wird das Programm in der Steuerung vollständig gelöscht; Sie sollten also sicher sein, dass Sie es vorher auf dem PC gespeichert haben.

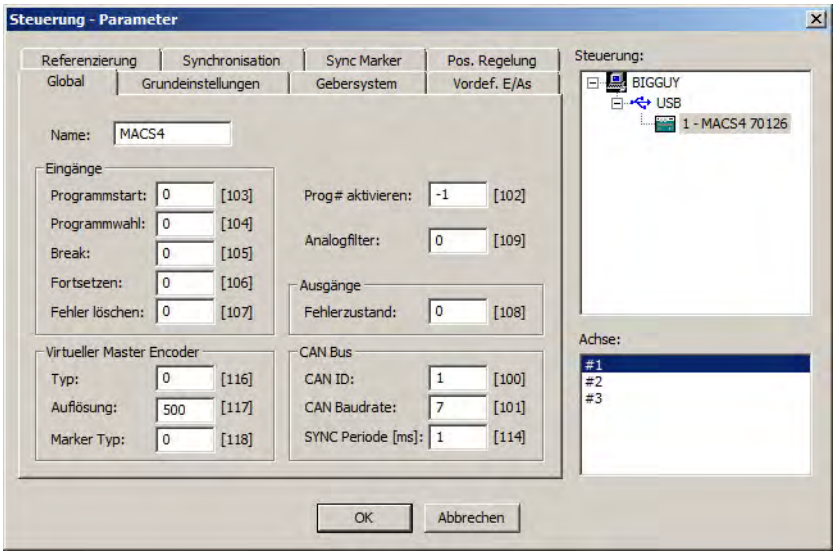
Klicken Sie auf **Alle Löschen**, wenn Sie alle Programme in der Steuerung löschen wollen.

!!! Vergewissern Sie sich zuvor, dass Sie die Programme noch im PC zur Sicherheit oder für das Archiv gespeichert haben, denn sie werden in der Steuerung vollständig gelöscht.

Info Wählen Sie ein Programm und klicken Sie auf → **Info** um die Compiler-Optionen zu sehen, die zum Kompilieren des Programms benutzt werden. Beachten Sie, dass diese Funktion bei einigen älteren Steuerungen nicht vorhanden ist.

Parameter Es gibt Globale Parameter, die für die gesamte Steuerung gelten und Achsparameter, die für jede Achse unterschiedlich sein können.

Parameter > Bearbeiten When parameters are being edited, then a dialog similar to the following will be displayed. A tab will be displayed for each of the various groups of parameters.



Es können verschiedene Achsen und verschiedene Steuerungen ausgewählt werden. Dann werden die Parameter für diese Achse und diese Steuerung dargestellt und können modifiziert werden. Beachten Sie, dass keine Änderung zu irgendeiner Steuerung geschrieben wird, bis auf „OK" geklickt wird. Erst wenn auf „OK" geklickt wird, werden die modifizierten Parameter in alle Steuerungen geschrieben.

In allen Parameter-Registern sehen Sie die interne Parameternummer nach dem Parameternamen. Detaillierte Informationen, Parameterwerte und Werkseinstellungen finden Sie anhand dieser Nummer in der Parameter-Referenz. Oder drücken Sie **F1,** wenn der Cursor in einem der Eingabefelder steht und die Informationen zu diesem Parameter werden eingeblendet.

!!! Mit dem globalen Parameter „***Name***" können Sie einen Namen der Steuerung zuweisen oder den vorhandenen Namen einer Steuerung ändern. Dieser Name wird in den diversen Dialogen in der APOSS Benutzeroberfläche dargestellt und erleichtert die Unterscheidung, wenn mehrere Steuerungen benutzt werden.

In Steuerungen mit Firmware Version 6.01.10 oder höher werden die Namen in UTF-8 Kodierung gesichert. Dies erlaubt jedes beliebige Zeichen im Namen, sogar Zeichen aus anderen Sprachen. Die Namen dürfen jedoch nicht länger als 8 Bytes sein (wohlgemerkt: einige UTF-8 Zeichen erfordern mehr als ein Byte). In Steuerungen mit Versionen älter als 6.01.10 sind die Namen auf ANSI Zeichen im Bereich von 0x20 bis 0x5F beschränkt (Akzentbuchstaben dürfen nicht benutzt werden) und der Name darf nicht länger als 8 Zeichen sein.

Parameter > Speichern in Datei Für Backup-Zwecke kann die gesamte Steuerungskonfiguration (d.h. die globalen Parameter, Achsparameter, Benutzerparameter und Arrays) als Datei in den PC geladen und gespeichert werden. Diese Datei kann später wieder in die Steuerung heruntergeladen werden, falls die Konfiguration wiederhergestellt werden muss.

Wählen Sie die Steuerung aus, klicken Sie dann auf → **Sichern** und geben Sie einen Namen in das folgende Dialogfeld ein. Falls Parameter von mehreren Steuerungen gesichert werden sollen, dann wählen Sie einfach nacheinander die Steuerungen aus und → **Sichern** erneut.

Parameter > Wiederherstellen aus Datei

Wenn eine Steuerungskonfiguration als Backup-Datei im PC gespeichert wurde, kann die Konfiguration durch Downloaden dieser Konfigurationsdatei wiederhergestellt werden.

Klicken Sie auf **Parameter → Wiederherstellen aus Datei** und wählen Sie im folgenden Dialogfeld die Datei aus, die geladen werden soll. Im nächsten Dialogfeld wählen Sie dann die Steuerung aus, in die die Daten geladen werden sollen. Markieren Sie im Dialogfeld, welche Parameter geladen werden sollen. Dann klicken Sie auf → **Wiederherstellen**. Die ausgewählten Parameter werden in die Steuerung geladen. Alle früheren Parameter werden überschrieben.

Falls die gleichen Daten in mehr als eine Steuerung geladen werden sollen, dann wählen Sie einfach eine andere Steuerung aus und klicken erneut auf → **Wiederherstellen**.

Parameter > Alle Dateien speichern

Wenn mehrere Steuerungen mit einer Schnittstelle verbunden sind (z.B. in einem CAN-Netz), können Sie ganz leicht und mit nur einem Befehl die Konfigurations-Informationen von allen Steuerungen dieses Netzwerks in Dateien speichern. Klicken Sie auf **Parameter → Alle in Dateien speichern**. Geben Sie im folgenden Dialog einen „Basis" Namen für die Dateien ein, in die die Konfigurationen gespeichert werden sollen. Es wird für jede Steuerung eine Konfigurationsdatei erzeugt. Die Dateien werden mit „Basis-id.zbc" bezeichnet, wobei „Basis" der zuvor definierte Name ist, „id" die ID-Nummer der Steuerung (z.B. CAN ID) und „zbc" die Dateierweiterung. Alle Dateien haben „zbc" als Dateierweiterung.

Speicher

RAM speichern

Mit **RAM speichern** werden alle Programme, Parameter und Arrays vom RAM in das EPROM gespeichert. Dies entspricht dem Befehl SAVEPROM, Daher wird diese Funktion normalerweise nur benötigt, um falls notwendig Arrays zu sichern, weil Programme und Parameter automatisch gesichert werden. Alle Daten, die nicht vom RAM in das EPROM gespeichert wurden, gehen verloren, wenn die Steuerung ausgeschaltet wird.

!!! Einige sehr alte Versionen von Steuerungen speichern nicht automatisch Programme in das EPROM. Benutzen Sie in diesem Fall die Funktion, um Programme in das EPROM zu speichern.

EPROM löschen

Mit **EPROM löschen** werden alle Parameter auf die Init-Werte zurückgesetzt und alle Arrays sowie Programme gelöscht. Die Steuerung wird auf die Werkseinstellungen zurückgesetzt (Reset). Allerdings erst nach dem Aus- und Wiedereinschalten der Steuerung.

!!! Beachten Sie also Folgendes, wenn Sie das EPROM löschen:

1. Prüfen Sie, ob Sie alle noch benötigten APOSS-Programme auf dem PC gesichert haben. Diese werden benötigt, um sie nach dem Wiedereinschalten wieder in die Steuerung zu laden.
2. Prüfen Sie, ob Sie die Steuerungskonfiguration in eine Backup-Datei auf dem PC gesichert haben (mit **Parameter → Speichern in Datei**). Mit dieser Datei können Sie die vorherigen Parameter und Arrays, falls erforderlich, wieder laden.
3. Klicken Sie auf **Speicher → EPROM löschen**.
4. Laden Sie die benötigten Konfigurationsparameter und APOSS-Programme wieder in die Steuerung.

Reset

Parameter

Mit **Reset → Parameter** werden alle globalen Parameter und alle Achsparameter in der Steuerung auf die Werkseinstellung (Init-Werte) zurückgesetzt.

Arrays

Mit **Reset → Arrays** können Sie alle Arrays im RAM löschen. Diese Funktion bewirkt das Gleiche, wie der Befehl DELETE ARRAYS.

!!! Wenn Sie anschließend **Speicher → RAM speichern** oder ein APOSS-Programm führt einen SAVE ARRAYS Befehl aus, werden auch die Arrays im EPROM überschrieben!

Vollständig

Mit **Reset → Vollständig** werden alle Parameter zurückgesetzt und alle Programme und Arrays gelöscht. Die Steuerung wird auf Werkseinstellungen zurückgesetzt.

!!! **Reset → Vollständig** wird sofort ausgeführt; nicht erst nach dem Aus- und Wiedereinschalten der Steuerung wie bei **Speicher → EPROM löschen**.

Fehler Historie

Klicken Sie auf **Steuerung → Fehler Historie**, um alle Fehler zu sehen, die in der Steuerung auftraten.

!!! Einige ältere Versionen von Steuerungen unterstützen diese Funktion nicht. Dann ist sie auch nicht aktiviert.

Die letzten Fehler werden am Anfang, die ältesten am Ende der Liste aufgelistet. Die Liste wird jedes Mal gelöscht, wenn die Steuerung ausgeschaltet wird. Sie enthält maximal 50 Fehler; dann wird der älteste Fehler entfernt, sobald ein neuer zur Liste hinzukommt.

Doppelklicken auf einen Fehler in der Liste schließt den Fehler-Historie-Dialog und markiert diese Zeile im Editierfenster.

!!! **Achtung:** Die APOSS-IDE kann nicht erkennen, ob das Programm das gerade bearbeitet wird, zu dem Programm gehört, das gerade in der Steuerung ausgeführt wird, als der Fehler auftrat. Das liegt in der Verantwortung des Anwenders. Wenn das Programm nicht dazu-gehört, wird die falsche Zeile im Editierfenster markiert.

Folgende Informationen werden aufgelistet:

Zeit Systemzeit der Steuerung, als der Fehler auftrat. Siehe „sys_clock" in den Systemprozessdaten.

Alter Alter des Fehlers (in Sekunden), wenn das Fehler Historie Dialogfeld geöffnet wird (d.h. wie lange es her ist, als der Fehler auftrat). Es wird jedes Mal aktualisiert, wenn das Dialogfeld aktualisiert wird.

Fehler Fehlernummer. Siehe Kapitel „Fehler-Referenz und Meldungen".

Meldung Text der Fehlermeldung.

Wert Optionaler Wert, der mit dem Fehler gespeichert wird. Die Bedeutung dieses Wertes hängt vom Fehler ab.

Offset Binärer Offset innerhalb des kompilierten Programms, in dem der Fehler auftrat.

Line Zeilennummer des Programms, das bearbeitet wurde, als der Fehler auftrat.

Quellcode Name des Programms, das bearbeitet wurde, als der Fehler auftrat.

Folgende Funktionen stehen zur Verfügung:

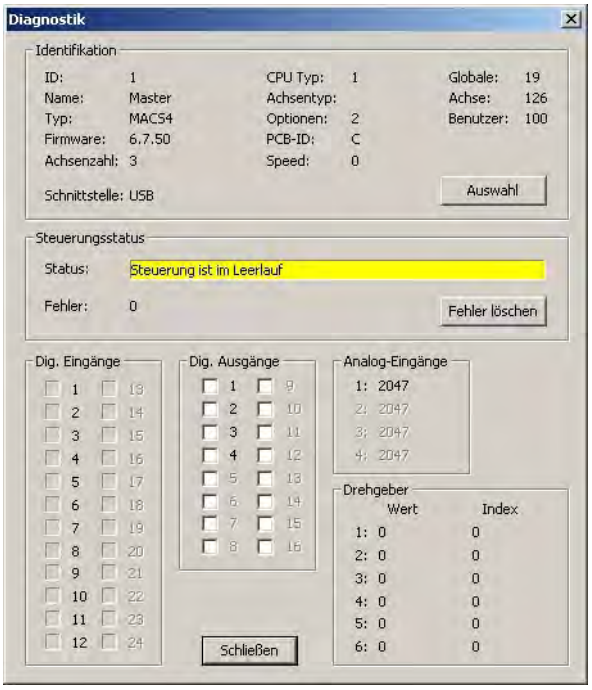
Neu lesen Aktualisiert die Liste mit der aktuellen Fehler Historie der Steuerung. Beachten Sie, dies aktualisiert das "age" der vorhandenen Fehler in der Historie.

Schreiben Schreibt die Fehler Historie in eine ASCII Textdatei, damit die Fehler analysiert werden können. Diese Datei kann später mit einem beliebigen Texteditor bearbeitet oder in ein Tabellenkalkulationsprogramm importiert werden

Löschen Löscht die Fehler Historie.

Diagnose Bericht Mit **Steuerung → Diagnose Bericht** können Sie eine Textdatei erzeugen, die den kompletten Status der Steuerung enthält. Diese Datei wird oft vom Support der zub machine control AG benötigt, um bei Diagnose Problemen helfen zu können.

Diagnostik Klicken auf **Steuerung → Diagnostik** öffnet ein Fenster mit den Diagnostik-Informationen der angeschlossenen Steuerung:



Die Checkboxen „Digitale Eingänge", „Digitale Ausgänge" und „Analoge Eingänge" sind grau, wenn die Ein- und Ausgänge nicht den physikalischen Pins der Steuerung entsprechen.

!!! Beachten Sie, dass in einigen Fällen digitale Ausgänge, die zwar keinem physikalischen Pin entsprechen, noch als „virtueller" digitaler Ausgang genutzt werden könnten.

Firmware Upload Mit **Steuerung → Firmware Upload** kann die aktuelle Firmware der Steuerung hochgeladen und in eine „.bin" Datei gespeichert werden. Es muss eine Steuerung angeschlossen sein, bevor die Funktion aktiviert wird. Diese Funktion ist nur für Steuerungen ab Firmware Version 6.07.68 verfügbar.

Menü Tools

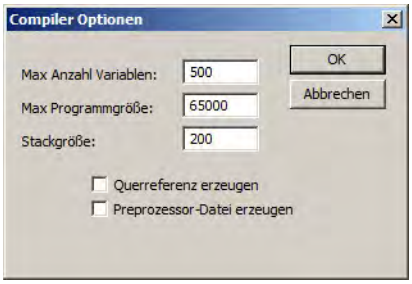
Das Menü **Tools** bietet umfangreiche Werkzeuge, die jeweils in eigenen Abschnitten im Kapitel „APOSS Tools" beschrieben werden: CAM-Editor, Array Editor und Oszilloskop.

Menü Einstellungen

In diesem Menü können Sie die Compiler Optionen auswählen, die Einstellungen der Schnittstelle ändern, die Farben des Editors nach Ihren Wünschen einstellen und verschiedene andere Optionen sowie die Sprache auswählen.

Compiler

Die Default-Werte für die Compiler Optionen sind für die meisten Anwendungen passend gesetzt.



Max Anzahl Variablen

Die **Maximale Anzahl** der **Variablen** hat direkten Einfluss auf die Menge des Speicherplatzes, der in der Steuerung reserviert wird. Die minimal benötigte Anzahl an Variablen kann anhand des Anwendungsprogramms wie folgt berechnet werden:

Minimaler Variablenspeicherplatz =
- Anzahl aller globalen Variablen
- + 1 Variable pro DIM-Array (unabhängig von der Array-Größe)
- + 1 Variable pro (!) Element von globalen Arrays
- + Reserve für spätere Erweiterungen (z.B.: 50 Variablen)

Die unterschiedlichen Typen von Arrays und Variablen und deren Verwendung sind in Variantenvergleich: „Arrays, Variablen: global, lokal?" in Kapitel „Programmieren mit APOSS" im Abschnitt „Datenhandling" erläutert.

Eine unzureichende Größe des Variablenspeichers wird beim Kompilieren, d.h. vor dem Download des Programms in die Steuerung erkannt. Es erscheint in diesem Fall eine entsprechende Compiler-Fehlermeldung.

Max Programmgröße

Die **maximale Programmgröße** definiert die maximale Größe eines Programms in Bytes nachdem es kompiliert und fertig zum Downloaden in die Steuerung ist.

Beachten Sie, dass einige ältere Steuerungen Programme > 65.000 Bytes nicht unterstützen.

Stackgröße

Die **Stackgröße** legt fest wie viel Speicher (Stack genannt) für das interne Handling von Funktionen und deren lokalen Daten reserviert wird. Bei stark verschachtelten Funktionsaufrufen oder starker Nutzung von lokalen Variablen oder insbesondere lokalen Arrays muss die Stackgröße erhöht werden. Die benötigte Stackgröße lässt sich anhand des Anwenderprogramms und unter Kenntnis des Programmablaufs überschlagmäßig wie folgt berechnen:

Minimale Stackgröße =
- Maximale Summe aller lokaler Variablen
- in verschachtelten Funktionsaufrufen,
- die gleichzeitig aktiv sind
- + maximale Summe aller lokalen Array-Elemente,
- in verschachtelten Funktionsaufrufen,
- die gleichzeitig aktiv sind
- + Reserve und Overhead (z.B.. 100 Bytes)

Die Verschachtelung von Funktionen wirkt sich somit stark auf die benötigte Stackgröße aus. Eine unzureichende Stackgröße kann bei verschachtelten Funktionen erst während der Programmausführung erkannt werden. Es wird in diesem Fall die nächste kritische Funktion nicht mehr aufgerufen und ein APOSS Fehler_93 gemeldet. Auf diesen Laufzeitfehler kann in der programmierten Fehlerbehandlung (ON ERROR … GOSUB) reagiert und ein kontrolliertes Programmverhalten sichergestellt werden.

Querreferenz erzeugen

Für das korrekte Funktionieren von APOSS sind Querreferenz-Dateien nicht notwendig und der Anwender darf diese Option deaktivieren. Die Dateien sind normalerweise nur für das Personal der zub machine control AG beim Debuggen des APOSS-Compilers von Interesse.

Preprozessor-Datei erzeugen

Alle APOSS Programmdateien werden vor dem Kompilieren aufbereitet (siehe Preprozessor in Kapitel „Programmieren mit APOSS"), wobei verschiedene Strings im Originalprogramm ersetzt werden. Wenn der Anwender genau sehen muss, was der Compiler kompiliert, kann mit dieser Option eine Preprozessor-Datei erzeugt werden. Das ist eine „.txt" Datei mit dem gleichen Namen und im gleichen Verzeichnis wie die Originaldatei. Diese Datei enthält den Output des Preprozessors und wird jedes Mal erzeugt, wenn der Anwender das Programm kompiliert.

Schnittstellen Parameter

Normalerweise wurden die Schnittstellen Parameter schon bei der Inbetriebnahme der Steuerung festgelegt (siehe Kapitel „Erste Schritte", Steuerung anschließen). Falls jedoch aus irgendeinem Grund die Verbindung geändert werden muss (z.B. wenn sich die Baudrate oder der COM Port geändert hat, wenn der ID Scanbereich der Steuerung erweitert werden muss, wenn eine neue Schnittstelle hinzugefügt wurde, usw.) können mit **Einstellungen → Schnittstelle** die Parameter aktualisiert werden. Ein Dialogfenster ähnlich dem folgenden wird dargestellt:

Wählen Sie einfach die passende Schnittstelle aus und ändern Sie die notwendigen Parameter.

!!! Beachten Sie, dass nur eine Schnittstelle als Default gewählt werden kann. Diese Schnittstelle wird benutzt um die Verbindungen zu den Steuerungen nach einem **Esc** wiederherzustellen.

!!! Die Einstellung der Schnittstelle wirkt sich nur auf künftige Verbindungen aus. Verbindungen, die schon vorhanden sind, bleiben erhalten und werden nicht geändert. Wenn eine bereits vorhandene Verbindung zu einer Steuerung geändert werden muss, dann müssen Sie diese **→ Schnittstelle schließen**, die Schnittstellen-Einstellungen ändern und danach erneut die **→ Steuerung auswählen**.

Sprache

Wenn Sie eine andere Sprache wünschen, klicken Sie auf **Einstellungen → Sprache** und wählen im darauf folgenden Dialogfeld zwischen Englisch und Deutsch. Danach müssen Sie APOSS schließen und neu starten.

Editor Einstellungen

Zur besseren Übersicht können den verschiedenen Programmelementen wie Kommentar, Schlüsselwort, Ziffer usw. unterschiedliche Farben zugeordnet werden. Auch Tabulatoren können für eine bessere Lesbarkeit genutzt werden.

Farbe — Markieren Sie das Programmelement, z. B. Kommentar und wählen Sie dazu die gewünschte Farbe.

Tabulatoren — Wählen Sie den gewünschten Tabulator-Wert ab 2 Zeichen aufwärts.

Tabulatoren konvertieren — Klicken Sie auf → **Tab in Leerzeichen umwandeln,** um alle Tab-Zeichen während des Schreibens in Leerzeichen umzuwandeln.

Benutzen Sie **Bearbeiten → Tab. Umwandeln**, wenn Sie alle vorhandenen Tabs eines Programms in Leerzeichen umwandeln wollen.

Leerzeichen am Zeilenende entfernen — Entfernt unerwünschte Leerzeichen sowie Tab-Zeichen am Zeilenende, bevor die Dateien gespeichert werden.

Syntax-Tooltips aktivieren — Wenn aktiviert, werden Hilfe-Tipps in kleinen gelben Popup-Boxen mit der Syntax des Befehls dargestellt, wenn der Mauszeiger über einem Befehl in der Programmzeile verweilt.

Optionen — Das Dialogfenster bietet verschiedene Optionen um das Verhalten von APOSS zu steuern. Geänderte Einstellungen werden sofort beim nächsten Einsatz genutzt.

Abbrechen nur mit Umschalt + Esc — Normalerweise sendet in APOSS die **Esc**-Taste einen „Abbrechen"-Befehl zu einer verbundenen Steuerung, um alle laufenden Programme abzubrechen. Falls sich der Antrieb dreht, wird er mit der maximal erlaubten Geschwindigkeit abgebremst. Stoppt man einen Antrieb während der Fahrt auf diese Weise, kann das System, das mit dem Antrieb verbunden ist, beschädigt werden.

Um zu vermeiden, dass der Anwender unbeabsichtigt durch Drücken der **Esc**-Taste ein laufendes Programm abbricht, kann in **Einstellungen → Optionen** eingestellt werden, dass Abbrechen nur mit **Umschalt+Esc** möglich ist. Ist diese Option gesetzt, kann der Anwender weiterhin ein ausführendes Programm abbrechen, jedoch wird durch Drücken der **Umschalt+Esc** verhindert, dass es unbeabsichtigt passiert.

Zuletzt geöffnete Dateien wiederherstellen — Wenn diese Option aktiviert ist, wird beim Starten von APOSS versucht, automatisch alle Dateien wieder zu öffnen, die offen waren, als APOSS zuletzt benutzt wurde.

Schnittstellen wieder öffnen — Wenn diese Option aktiviert ist, wird – sobald APOSS gestartet ist und vorher offene Dateien wieder geöffnet sind – automatisch versucht, die Dateien mit den Steuerungen wieder zu verbinden, mit denen sie zuletzt verbunden waren.

Automatisch neu verbinden bei Verbindungsabbruch	Wenn diese Option aktiviert ist, wird APOSS nach einem Verbindungsabbruch automatisch versuchen, wieder die mit den zuvor verbunden Steuerungen zu verbinden. APOSS wird eine Minute lang alle zwei Sekunden versuchen die Verbindung herzustellen.
Fenster maximiert öffnen	Wenn diese Funktion aktiviert ist, werden alle Fenster in APOSS in Bildschirmgröβe geöffnet. Sie können danach falls gewünscht verkleinert werden; sie werden immer in Bildschirmgröβe geöffnet. Ist diese Funktion nicht aktiviert, wird nur das erste Fenster in maximaler Größe geöffnet und alle weiteren immer ein wenig kleiner, so dass alle Fenster gleichzeitig zu sehen sind.
Auto-Speichern	Wenn **Auto-Speichern** aktiviert ist, speichert APOSS die aktuelle Datei, die gerade editiert wurde, automatisch auf die Festplatte, bevor diese kompiliert und ausgeführt wird. Ist **Auto-Speichern** nicht aktiviert, benutzt APOSS zum Kompilieren eine temporäre Datei.
Binär Datei Unterstützung	Wenn aktiviert, erlaubt APOSS das direkte Handling der binären Daten (Image) eines kompilierten Programms im Fenster **Steuerung → Programm**e. Dies beinhaltet den Upload kompilierter Programme von der Steuerung, so dass diese im PC gespeichert werden können und beinhaltet das Downloaden früherer gespeicherter Images zurück in die Steuerung. Beachten Sie, dass diese binären Images nicht bearbeitet werden können; sie können nur gesichert und zurück geladen werden. Beachten Sie auch, dass einige ältere Steuerungen diese Funktion nicht unterstützen. Wenn die Option aktiviert ist, erscheint die Funktion → **In Datei kompilieren** auch im Menü **Entwicklung** (sobald APOSS das nächste Mal wieder gestartet wird). Dann können Sie die aktuelle Datei manuell kompilieren und in eine Binär-Datei sichern. Siehe auch BinFileMap in „Abbildungen“ im Kapitel „Technische Referenz“.
In Farbe drucken	Wenn aktiviert, werden die Programme aus dem Editierfenster mit den Farben gedruckt, wie sie im Editierfenster benutzt wurden. Ansonsten werden sie Schwarzweiβ gedruckt.
Array Editor Parameter Unterstützung	Normalerweise liest und schreibt der Array Editor alle Benutzerparameter und Arrays. Wenn jedoch diese Einstellung aktiviert ist, liest und schreibt der Array Editor auch die globalen Parameter, die temporären und permanenten Achsenparameter, die Achsprozessdaten und Systemprozessparameter.
Debug-Datei erzeugen	Aktivieren Sie diese Funktion, wenn Sie eine Debug-Datei mitschreiben lassen wollen. Normalerweise wird dies nur für den Support benötigt; daher sollte die Option in der Regel nicht aktiviert sein.
Protokoll Font	Zeigt die Meldungen im Kommunikations-Fenster als → **Proportionalschrift** oder mit → **festem Zeichenabstand**. Eine Änderung der Auswahl wirkt sich erst in neu geöffneten Fenstern aus.
Benutzer-Modus	Mit der Einstellung des Benutzer-Modus kann APOSS auf den Erfahrungsgrad des Anwenders zugeschnitten werden: Einsteiger, Standard oder Experte. Zurzeit gilt diese Einstellung hauptsächlich für die **Oszilloskop** Funktionen. Der Benutzer-Modus bestimmt auch, wie viele SDO Werte in den verschiedenen SDO Auswahllisten angeboten werden. Wenig benutzte SDO Werte werden für Nicht-Experten aus der Liste gelöscht.
Anwenderspezifische Bereiche	**Benutzerdefinierte SDO Dateien** erlauben dem erfahrenen Anwender, die Inhalte der SDO Auswahllisten, die in APOSS benutzt werden, individuell anzupassen. Zum Beispiel jene, die im Überwachungsfenster und im **Oszilloskop** benutzt werden. Die Dateinamen können auch per Drag & Drop eingegeben werden. Mittels der Einstellung **Programm mit der Zustandsmaschine** kann der erfahrene Anwender ein kundenspezifisches Programm mit der Zustandsmaschine im **Tune Oszilloskop** benutzen. Die Dateinamen können auch per Drag & Drop eingegeben werden.

In **Tune Oszilloskop Monitor Datei** kann der erfahrene Anwender ein kundenspezifisches Testfenster auswählen und im **Tune Oszilloskop** benutzen. Die Dateinamen können auch per Drag & Drop eingegeben werden.

Im **Debug Verzeichnis** kann der Anwender festlegen, wo die Debug Logdateien gespeichert werden sollen. Die Dateinamen können auch per Drag & Drop eingegeben werden. Wenn eine Datei in dieses Feld gezogen wurde, wird auch das Verzeichnis auf jenes gesetzt, das die Datei enthält.

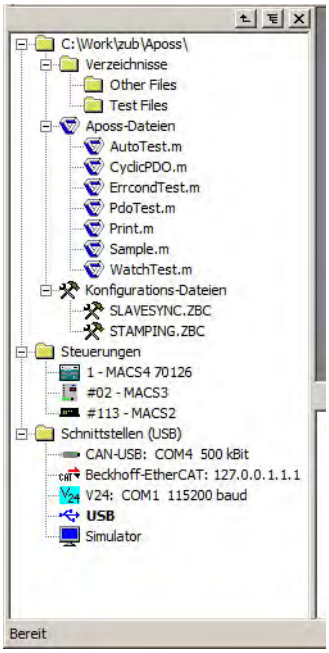
Oszilloskop Dieser Dialog ermöglicht die Auswahl von verschiedenen Optionen, um das Verhalten des APOSS Oszilloskop zu steuern. Siehe **Oszilloskop Einstellungen** in Kapitel „APOSS Tools“, APOSS Oszilloskop.

Menü Fenster

Die Funktionen des Menüs **Fenster** verhalten sich Windows-konform (d.h. **Kaskadieren** für sich überlappende Fenster nach unten und nach rechts versetzt, **Vertikal** für jedes neben dem anderen oder **Horizontal** für jedes unter dem anderen).

APOSS Sidebar

Die **Sidebar** bieten einen schnellen Zugriff insbesondere wenn mehrere Programme, Steuerungen und Schnittstellen genutzt werden.



Wenn das Fenster nicht gewünscht wird, kann es mit dem Schließen-Button ☒ rechts oben geschlossen werden. Es kann jederzeit mit **Fenster → Sidebar anzeigen** wieder geöffnet werden. Die Breite des Fensters kann durch Klicken und Ziehen der rechten Seite des Fensters angepasst werden, ähnlich wie bei der Trennleiste (Splitter) im Editierfenster.

Das Fenster zeigt alle Verzeichnisse und Dateien im aktuellen Verzeichnis sowie alle gerade verbundenen Steuerungen und alle verfügbaren Schnittstellen. Die gerade ausgewählte Default-Schnittstelle steht in Klammern hinter dem Schnittstellen-Titel. Wenn die Liste der Schnittstellen expandiert wird, ist die Default-Schnittstelle fett dargestellt.

Doppelklicken auf eine der Positionen in der Sidebar bewirkt Folgendes:

Verzeichnis — Es wird in dieses Verzeichnis gewechselt.

Datei — Wenn die Datei bereits offen ist, wird der Schwerpunkt auf diese gesetzt. Ist die Datei nicht offen, wird sie geöffnet. Beachten Sie, dass Doppelklicken auf eine Konfigurationsdatei den CAM Editor öffnet, auf eine Oszilloskop-Datei das Oszilloskop öffnet, usw.

Steuerung — Wenn diese Steuerung gerade mit einem Editierfenster verbunden ist, wird der Schwerpunkt auf dieses Fenster gesetzt. Dadurch kann das verbundene Fenster „leicht" gefunden werden, vor allem wenn mehrere Fenster geöffnet sind. Ist die Steuerung mit keinem Editierfenster verbunden ist, wird der Diagnostik-Dialog für diese Steuerung geöffnet.

Schnittstellen — Wenn das Fenster, auf dem gerade der Schwerpunkt liegt, ein Editierfenster ist, wird mit dem Doppelklick automatisch eine Steuerung mit der ausgewählten Schnittstelle verbunden.

Ist das Editierfenster schon mit einer Steuerung verbunden, wird die vorhandene Verbindung geschlossen und eine neue geöffnet. Beachten Sie, dass die ausgewählte Schnittstelle keine Default-Schnittstelle benötigt; jede beliebige Schnittstelle kann ausgewählt werden. So kann man schnell verschiedene Steuerungen mit verschiedenen Editierfenstern verbinden. Ist das aktuelle Fenster kein Editierfenster, erfolgt eine Warnung,

Rechtsklick auf eine Schnittstelle öffnet ein Kontextmenü, in dem diese Schnittstelle als Default-Schnittstelle ausgewählt werden kann oder die Schnittstellenparameter für diese Schnittstelle gesetzt werden können.

Die **Sidebar** enthält außerdem die folgenden Buttons in der Ecke oben rechts:

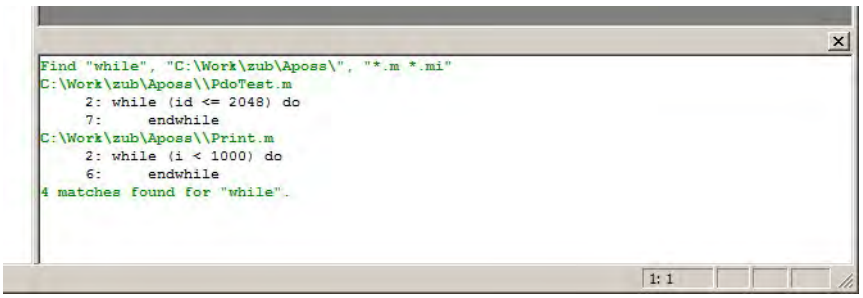
Aufwärts — Das in der Sidebar gezeigte Verzeichnis wird in eine Ebene höher gewechselt.

Browse — Es wird ein Dialog geöffnet, in dem der Anwender das Verzeichnis auswählen kann, das in der Sidebar dargestellt werden soll.

Schließen — Das Sidebar Fenster wird geschlossen. Es kann jederzeit mit **Fenster → Sidebar anzeigen** wieder geöffnet werden.

APOSS Findbar

Die **Findbar** wird zur Darstellung der Ergebnisse aus **Bearbeiten → Suchen in Dateien** verwendet. Es steht immer zuunterst im Haupt-APOSS-Fenster. Beispiel:



Wenn das Fenster nicht gewünscht wird, kann es mit dem Schließen-Button rechts oben geschlossen werden. Es wird automatisch wieder geöffnet, sobald **Bearbeiten → Suchen in Dateien** das nächste Mal verwendet wird. Es kann auch jederzeit mit **Fenster → Findbar anzeigen** geöffnet werden. Die Höhe des Fensters kann durch Klicken und Ziehen des oberen Fensterrandes angepasst werden, ähnlich wie bei der Trennleiste (Splitter) im Editierfenster.

Doppelklicken auf irgendeine Zeile in der **Findbar** öffnet automatisch die entsprechende Datei (sofern sie nicht schon offen ist) und platziert den Textcursor in diese Zeile.

Menü Hilfe

Die Funktion → **Inhalt** startet die Online-Hilfe mit der Registerkarte „Inhalt", die Funktion → **Index** startet sie mit der Registerkarte „Index".

Wählen Sie → **SDO Dictionary,** dann wird sofort dieses Kapitel der Online-Hilfe aufgerufen.

Das Aussehen der Online-Hilfe hängt vom eingesetzten Betriebssystem ab.

Kontextsensitive Hilfe

Die **Befehlshilfe** und alle Parameter-Dialogfelder im Menü **Steuerung** sowie die Registerkarten im CAM-Editor bieten einen direkten Zugang zur Online-Hilfe. Markieren Sie den Befehl in der **Befehlshilfe** oder stellen Sie den Mauscursor in das Eingabefeld eines Parameters und drücken Sie **F1**. Sie erhalten dann den entsprechenden Abschnitt der Hilfe direkt angezeigt.

zub Website öffnen

→ **zub Website** öffnet den Standard-Internet-Browser und die zub Homepage.

Programminfo

Hier finden Sie die Versionsnummern des APOSS-Programms, des Interface-Treibers und des Compilers.

Für den C-Programmierer steht ein Handbuch des zb-Moc Interface-Treibers zur Verfügung. Damit kann der erfahrene Programmierer das Steuerungsprogramm direkt in das C-Programm einbinden.

Menü Download

Das Menü **Download** bietet eine einfache Schnittstelle um das Downloaden von Firmware und Programmen in mehrere Steuerungen zu unterstützen. Beachten Sie, dass alle APOSS Fenster, mit Ausnahme des APOSS-Fensters selbst, geschlossen sein müssen, bevor dieses Menü zu sehen ist.

Download Firmware

Bei allen neueren Steuerungen können Sie die Firmware (d.h. nicht kompilierte APOSS Programme) direkt mit APOSS updaten.

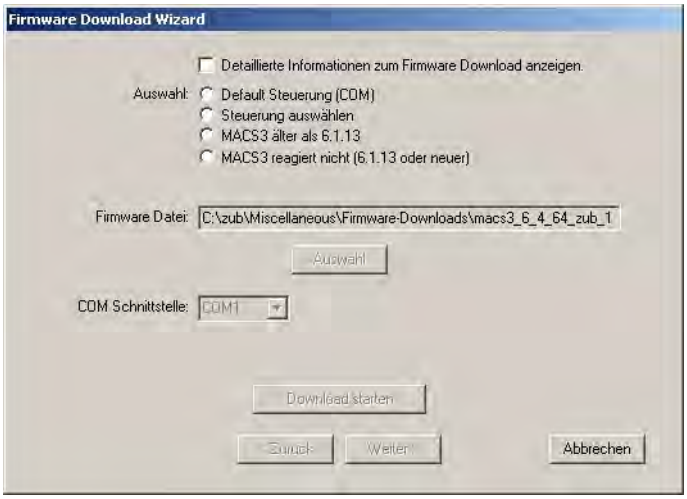
Sie erhalten die Firmware für verschiedene Steuerungen oder eine speziell für Ihre Anwendung optimierte Version von der zub machine control AG, entweder als Download von der zub machine control AG Website (**www.zub.ch**) oder über e-Mail (info@zub.ch). Auf Anfrage unterstützt Sie die zub AG auch, wenn Sie kundenspezifische Firmware oder Produkte benötigen, oder adaptiert spezielle Funktionen hierfür. Bleiben Sie in Kontakt mit der zub AG (info@zub.ch) um das Beste aus unseren Produkten und Ihren Maschinen herauszuholen!

Firmware Download Assistenten benutzen

Stellen Sie sicher, dass alle APOSS-Fenster mit Ausnahme des Haupt-APOSS-IDE-Fensters geschlossen sind. Klicken Sie dann auf → **Download** und wählen Sie → **Firmware**.

Wenn Sie das erste Mal Firmware downloaden oder wenn Sie mit dem Firmware-Download noch nicht vertraut sind, klicken Sie auf **Weiter** und folgen Sie den Anweisungen, die Sie durch den Download-Prozess führen.

Wenn Sie mit dem Download-Prozess vertraut sind, aktivieren Sie den → **Firmware Experten-Modus** und klicken auf **Weiter.** Sie können dann den gesamten Download-Prozess mit einem einzigen Dialogfenster und mit einem Minimum an Anweisungen durchführen:



Falls dieser Dialog gezeigt wird, Sie aber „kein Experte" sind, aktivieren Sie → **Detaillierte Informationen zum Firmware-Download zeigen** und klicken auf **Weiter**.

Der Dateiname der zuletzt heruntergeladenen Datei wird gespeichert und als Default benutzt, wenn **Download** erneut benutzt wird

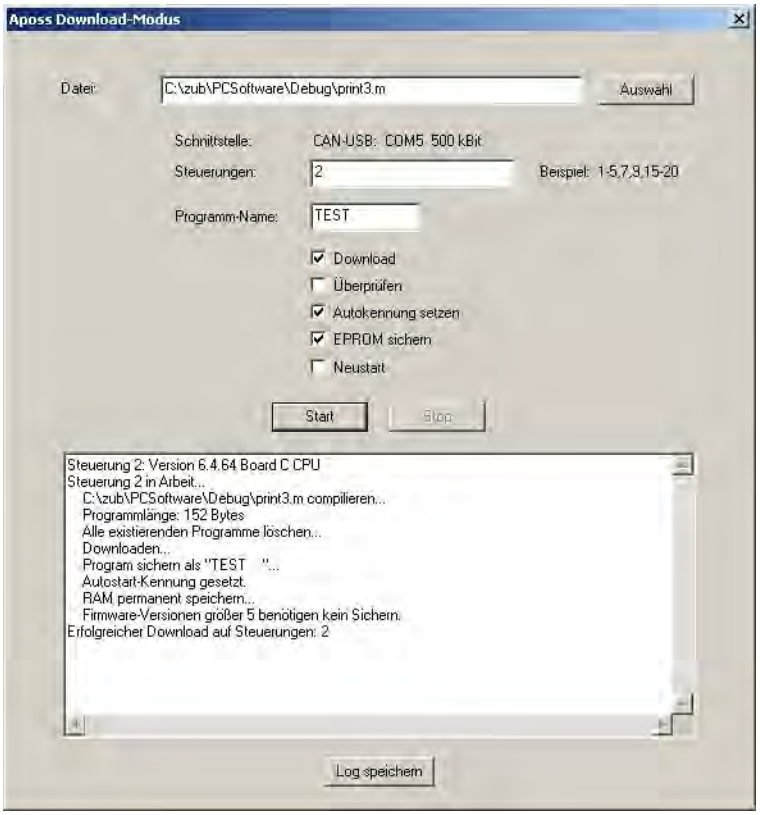
!!! Bitte lesen Sie das Hardware-Handbuch und vor allem die Sicherheitsanweisungen.

Download Programme Wählen Sie → **Programme** und im folgenden Dialogfenster das APOSS Programm das heruntergeladen werden soll. Falls die Schnittstelle mehrere Steuerungen unterstützt, dann geben Sie den ID-Bereich der Steuerungen ein, in die das Programm geladen werden soll.

Geben Sie einen Programmnamen ein, der zum Identifizieren des Programms in der Steuerung benutzt wird.

In Steuerungen mit Firmware Version 6.01.10 oder höher werden die Namen in UTF-8 Code gesichert. Dies erlaubt jedes beliebige Zeichen im Namen, sogar Zeichen aus anderen Sprachen. Die Namen dürfen jedoch nicht länger als 8 Bytes sein (wohlgemerkt: einige UTF-8 Zeichen erfordern mehr als ein Byte). In Steuerungen mit Versionen älter als 6.01.10 sind die Namen auf ANSI Zeichen im Bereich von 0x20 bis 0x5F beschränkt (Akzentbuchstaben dürfen nicht benutzt werden) und der Name darf nicht länger als 8 Zeichen sein.

Wählen Sie die gewünschten Aufgaben in den Checkboxen aus: **Download**, **Überprüfen**, **Autokennung setzen**, **EPROM sichern** und/oder **Neustart**. Wenn Sie zum Beispiel nur **Überprüfen** aber nicht **Download**en wollen, wird das Download-Programm nur überprüfen, ob die ausgewählte Datei diejenige ist, die schon in der Steuerung ist.



Start Download Programme Klicken Sie auf **Start**. Es wird mit der Default-Schnittstelle eine Verbindung zu den Steuerungen hergestellt. Alle Steuerungen, die gerade ein Programm ausführen, werden angehalten. Die folgenden Schritte werden für jede erreichbare Steuerung nacheinander ausgeführt.

Sie könnten zum Beispiel auch nur **Überprüfen** auswählen. Dann wird nur geprüft, ob die ausgewählte Datei diejenige ist, die schon in der Steuerung ist.

– Das Programm wird mit den zur Steuerung gehörenden Compiler-Einstellungen kompiliert.

- Alle vorhandenen Programme der Steuerung werden gelöscht.
- Das Programm wird heruntergeladen.
- Das Programm wird als Programm Nummer 0 mit dem festgelegten Programmnamen gespeichert. Ist kein Programmname festgelegt, werden die ersten 8 Zeichen des Dateinamens benutzt.
- Wenn **Überprüfen** ausgewählt ist, wird das heruntergeladene Programm wieder hochgeladen und die Download- und Upload-Versionen miteinander verglichen. Falls sie voneinander abweichen, wird der Download für diese Steuerung angehalten.
  Beachten Sie, dass nicht alle älteren Steuerungen eine Firmware enthalten, die ein Upload unterstützen. In diesem Fall wird die Auswahl der Funktion **Überprüfen** ignoriert.
- Die **Autokennung** wird gesetzt, falls ausgewählt.
- Alle Daten werden **im EPROM** gespeichert, falls ausgewählt. Bei neueren Steuerungen wird dies automatisch durchgeführt.
- Die Steuerung wird mit dem neuen Programm **neu gestartet,** falls ausgewählt.

Die Verbindung zu den Steuerungen wird geschlossen.

Treten während des Ausführens in einer Steuerung Fehler auf, wird die Ausführung für diese Steuerung angehalten, aber für alle anderen Steuerungen des ausgewählten ID-Bereichs fortgeführt. Wenn der gesamte Download fertig ist, wird eine Zusammenfassung im Dialogfeld ausgegeben. Diese zeigt die IDs von den Steuerungen, die ohne Fehler und jene bei denen Fehler auftraten. Sie können durch die Zusammenfassung scrollen, um festzustellen welche Probleme auftraten.

Log speichern

Der Fortgang des Downloads wird in dem unteren Teil des Dialogfelds gezeigt. Falls gewünscht, können diese Informationen mit → **Log speichern** in eine Textdatei gespeichert werden.

Falls der Download für eine oder mehrere Steuerungen fehlschlägt, erhalten Sie eine entsprechende Fehlermeldung. Wenn Sie die **Log**-Datei **speichern**, können Sie später die Fehler suchen und bearbeiten.

Programme debuggen

Die APOSS-IDE enthält einen mächtigen integrierten Debugger. Dieser bietet allgemeine Debug-Funktionen wie Einzelschritt, Breakpoints und die Möglichkeit, Programmvariablen zu lesen und zu setzen.

Der Debugger kann auch benutzt werden, wenn die Steuerung bereits ein Programm ausführt: Wenn die Steuerung kein Programm ausführt, starten Sie mit → **Programme debuggen** das Debuggen der Quelle, die Sie gerade bearbeiten.
Wenn die Steuerung aber bereits ein Programm ausführt, müssen Sie sicherstellen, dass die Quelle, die bearbeitet wird, zum dem Programm passt, das gerade ausgeführt wird.

!!! Der Debugger kann nicht in allen Fällen benutzt werden. Zum Beispiel ist es nicht möglich, den Debugger mit Programmen zu benutzen, die aktiv einen Motor steuern. Das Stoppen der Programmausführung an einem Breakpoint ist gleichbedeutend wie das Drücken der **Esc**-Taste, um die Programmausführung abzubrechen. Dies bremst und stoppt den Motor mit der maximal erlaubten Verzögerung. In vielen Fällen macht das Anhalten des Motors auf diese Art die Testprozedur ungültig und die Debug-Ergebnisse wertlos. Ebenso ist es unwahrscheinlich, dass wenn die Programmausführung nach einem Breakpoint fortgesetzt wird, dass der Motor korrekt wieder gestartet werden kann um das System in den Status zu bringen, in dem es vor dem Breakpoint war.

!!! Der Debugger kann auch nicht mit Programmen benutzt werden, die auf ON PERIOD Funktionen beruhen. Der interne Timer, der ON PERIOD Funktionen triggert, stoppt nicht wenn die Pogrammausführung an einem Breakpoint anhält. So könnte ein anstehender Interrupt dann einen ON PERIOD Aufruf auslösen, wenn die Programmausführung fortgesetzt wird.

Für Situationen, in denen der Debugger wie die oben nicht benutzt werden kann, bietet das Oszilloskop ausgezeichnete Debug-Einsatzmöglichkeiten. Es kann Programmvariablen und den Systemstatus überwachen und aufzeichnen, ohne dass die Programmausführung unterbrochen werden muss. Diese können dann später geprüft werden, um Probleme zu identifizieren. Für mehr Information siehe APOSS Oszilloskop im Kapitel „APOSS Tools".

Debugger starten

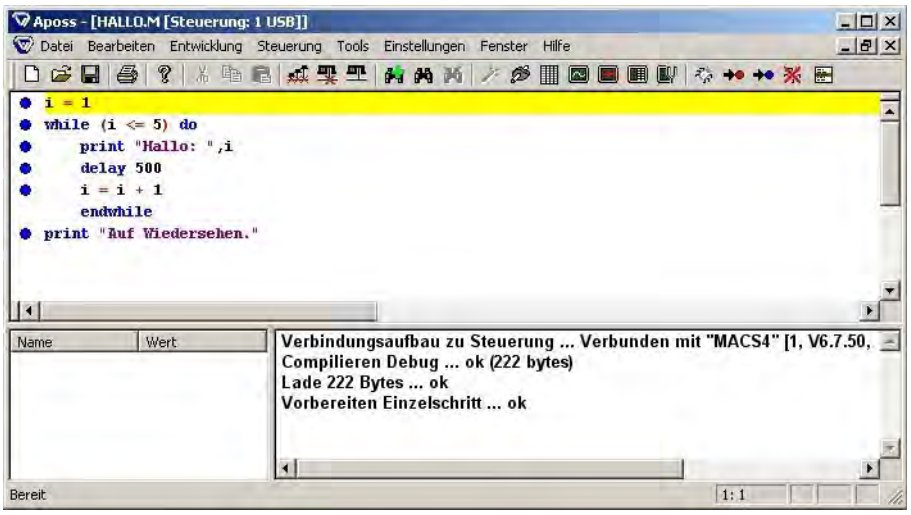
Zum Starten des Debuggers bearbeiten Sie das Programm ganz normal, so dass es im Editierfenster dargestellt wird. Dann klicken Sie auf **Entwicklung → Debugger starten** oder auf  . Wenn die Steuerung nicht gerade ein Programm ausführte, bewirkt dies folgende Aktionen:

1. Das Programm wird im Debug-Modus kompiliert und in die Steuerung heruntergeladen. Die Programmausführung wird jedoch zu diesem Zeitpunkt nicht gestartet.
2. Vor jeder ausführbaren Anweisung im Programm wird ein blauer Punkt gesetzt. Dies sind die Positionen, an denen der Anwender Breakpoints einfügen kann.
3. Die nächste Anweisung die ausgeführt wird (d.h. wenn die Ausführung gestartet ist oder fortgesetzt wird) wird gelb markiert.

Wenn die Steuerung aber bereits ein Programm ausführt, bewirkt das Starten des Debuggers folgende Aktionen:

1. Ein blauer Punkt wird im Programm vor jede ausführbare Anweisung gesetzt. Dies sind die Positionen an denen Sie Breakpoints einfügen können.
2. Die Programmausführung wird fortgesetzt.
3. Sie können das Programm zu jeder beliebigen Zeit durch Drücken der **Esc**-Taste anhalten (Pause). Die nächste auszuführende Anweisung (d.h. wenn die Ausführung fortgesetzt wird) wird gelb markiert.

So könnte das Editierfenster dann aussehen:



!!! Das Programm sollte nicht verändert werden, während der Debugger aktiv ist. Falls doch, stimmt die APOSS-IDE nicht mehr mit der Programmversion übereinstimmen, die in der Steuerung ausgeführt wird und der Debugger ist nicht mehr in der Lage, der Programmausführung korrekt zu folgen. Falls das Programm geändert werden muss, sollte der Debugger angehalten und der Test erneut von Anfang an gestartet werden.

Debugger stoppen

Um den Debugger zu stoppen und die Debug-Sitzung zu beenden klicken Sie auf **Entwicklung → Debugger stoppen** oder auf  Dies bewirkt Folgendes:

1. Entfernen der blauen Punkte, die ausführbare Anweisungen markieren.
2. Entfernen aller Breakpoints, die der Anwender gesetzt hat.

Falls die Steuerung ein Programm ausführt, wenn der Debugger gestoppt wird, wird die Steuerung das Programm weiter ausführen. Wenn das Programm gestoppt hat, kann der Anwender es an der Stelle, an der es gestoppt wurde, mit **Entwicklung → Fortsetzen** wieder starten.

Einzelschritt

Um mit einzelnen Schritten durch das Programm zu gehen, benutzen Sie **Entwicklung → Einzelschritt,** drücken **F9** oder klicken auf  . Dies führt die nächste Anweisung aus (d.h. die Anweisung, die gerade gelb markiert ist) und stoppt automatisch bevor die nächste Anweisung ausgeführt wird (d.h. an der nächsten Anweisung mit einem blauen Punkt). Dann wird diese nächste Anweisung gelb markiert.

Während die Ausführung angehalten ist, steht es dem Anwender frei den Wert einer beliebigen Programmvariablen zu prüfen und zu verändern, Breakpoints zu setzen oder zu entfernen, etc.

Zu jeder Zeit kann die Programmausführung ohne Einzelschritte mit **Entwicklung → Gehe zu Breakpoint** oder durch Klicken auf  fortgesetzt werden.

Breakpoints anwenden

Zum Setzen von Breakpoints doppelklicken Sie irgendwo in die Anweisung im Programm (außer auf den blauen Punkt), vor der der Breakpoint stehen soll. Der blaue Punkt ändert sich Rot und zeigt so den gesetzten Breakpoint an.

Erneutes Doppelklicken in eine Anweisung, die schon einen Breakpoint enthält, entfernt diesen wieder und der Punkt ist wieder blau markiert.

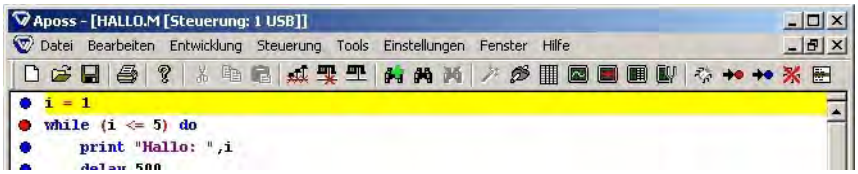
An jeder ausführbaren Anweisung im Programm (mit blauen Punkten markiert) kann ein Breakpoint gesetzt werden. Wenn die Programmausführung auf einen Breakpoint trifft, wird die Ausführung sofort vor der Anweisung mit dem Breakpoint angehalten. Die Anweisung wird dann gelb markiert, da es die nächste auszuführende Anweisung ist.

Während die Ausführung angehalten ist, kann der Wert einer beliebigen Programmvariablen geprüft und geändert werden, können Breakpoints gesetzt oder entfernt werden, etc.

Danach kann die die Pogrammausführung mit **Entwicklung → Gehe zu Breakpoint** oder durch Klicken auf  fortgesetzt werden. Das Programm wird dann so lange ausgeführt, bis es auf einen anderen Breakpoint trifft (d.h. bis zur nächsten Anweisung mit einem roten Punkt). Es ist auch möglich, nach dem Anhalten bei einem Breakpoint mit Einzelschritten fortzufahren.

Zu jeder Zeit kann der Anwender die Programmausführung durch Drücken der **Esc**-Taste anhalten. Die Ausführung stoppt sofort, nicht erst beim nächsten Breakpoint. Die nächste auszuführende Anweisung wird gelb markiert. Auch in diesem Fall kann der Anwender entweder mit **Entwicklung → Gehe zu Breakpoint** oder **→ Einzelschritt** [F9] fortfahren.

Das folgende Beispiel zeigt Breakpoints vor den PRINT Anweisungen:



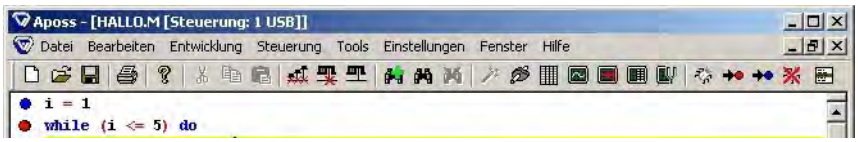
!!! Maximal 10 Breakpoints sind erlaubt.

!!! Beginnend mit der Steuerungs-Firmware Version 5.22.00 haben die Breakpoints die Befehle #DEBUG on/off ersetzt. Vorhandene #DEBUG Befehle müssen nicht unbedingt aus dem Programm entfernt werden, da sie ignoriert werden.

Darstellung und Änderung von Variablen

Zu jeder Zeit, während der Debugger aktiv ist (d.h. auch wenn das Programm ausgeführt oder angehalten ist), kann der aktuelle Wert einer Programmvariablen dargestellt werden. Klicken Sie dazu mit der linken Maustaste auf (oder direkt hinter) eine Variable. Der aktuelle Wert wird in einer gelben Popupbox so lange dargestellt, bis die Maus davon weg bewegt wird.

Im folgenden Beispiel wurde nahe der Variablen „i" in der PRINT Anweisung geklickt.



Zu jeder Zeit, während der Debugger aktiv ist, kann der aktuelle Wert einer Programmvariablen auch geändert werden. Klicken Sie dazu mit der rechten Maustaste auf (oder direkt hinter) die Variable.

Der folgende Dialog ermöglicht die Änderung des Wertes der Variablen:

Darstellung der ausführenden Zeile

Jederzeit – gleichgültig ob der Debugger gestartet wurde oder nicht – kann die aktuelle Anweisung, die gerade von der Steuerung ausgeführt wird (oder gleich ausgeführt werden soll), durch Klicken auf das Schaltsymbol  gelb markiert werden. Die Markierung wird durch nochmaliges Klicken auf selbige wieder entfernt. So kann man feststellen, welcher Programmteil zu einer bestimmten Zeit ausgeführt wird. Mehrfache Anwendung der Funktion hilft, die Programmteile zu ermitteln, die am häufigsten ausgeführt werden.