

LLWIN Lucky Logic für Windows

fischertechnik®



Inhalt

1. LLWin macht das Steuern einfach	1
Installation von LLWin	2
Aufrufen und Beenden von LLWin	3
2. Beispiele	4
Beispiel 1 Motorsteuerung	4
Beispiel 2 Zeitschalter	8
Beispiel 3 Impulszähler	9
3. Menübefehle	11
Datei	11
Bearbeiten	12
Unterprogramm	14
Run	15
Optionen	16
Fenster	19
Hilfe	20
4. Hinweise, Begriffe, Tips und Tricks	20
5. Verwendung von Unterprogrammen	22
Grundlagen zur Unterprogrammtechnik in LLWin	22
Projektaufbau bei Verwendung von Unterprogrammen	23
Bearbeiten und Aussehen von Unterprogrammen	23
Beispiel für ein Unterprogramm	24



6. Eingaben in Bausteinen		26
Variablen	V1... V99	26
Analogwerte	EX und EY	27
Konstanten	(Zahlen von -999 bis 9999)	27
Terminalparameter	EA, EB, EC und ED	27
Binäreingänge	1 ... 48	28
Motorausgänge	1 ... 8	28
7. Die in LLWin vorhandenen Bausteine		28
DEC VARIABLE	Herunterzählen einer Variablen um 1	28
DISPLAY	Anzeigen einer Variablen im Terminal	28
EINGANG	Verzweigung bei einem Binäreingang	29
ENDE	Ende eines Ablaufes	29
FLANKE	Flanke von 1 nach 0 am Eingang abwarten	30
INC VARIABLE	Hochzählen einer Variablen um 1	30
MELDUNG	Meldung auf dem Terminal anzeigen	30
MOTOR	Motorausgang schalten	30
POSITION	Warten bis eine Position erreicht wurde	31
START	Start eines Ablaufes	32
TERMINAL	Bedienterminal	32
VARIABLE	Zuweisung auf eine Variable	33
Vergleich	Verzweigung nach einem Vergleich	33
WARTE	wartet eine definierte oder zufällige Zeitspanne	34
Input	Eingang eines Unterprogrammes	34
Output	Ausgang eines Unterprogrammes	35
Text	Kommentartext im Bild	35
Anhang		36
Übersicht	der wichtigsten Befehle und Kommandos	36



1. LLWin macht das Steuern einfach

Das Programm LLWin steuert die fischertechnik-Modelle über das am PC angeschlossene Interface. Die Bedienung von LLWin ist sehr einfach erlernbar. Die interaktive grafische Benutzeroberfläche wird fast ausschließlich mit der Maus bedient. Einige Befehle kannst Du aber auch über die Tastatur direkt anwählen.

Um ein fischertechnik-Modell zu steuern, muß auf dem PC das Programm LLWin aufgerufen, ein Programmablauf geladen und gestartet werden. Dieser Programmablauf kann die Taster und Lichtschranken abfragen und Motoren und Lampen ein- bzw. ausschalten. So wie mit fischertechnik viele Modelle aus den Bausteinen aufgebaut werden können, können auch für die Steuerung der Modelle aus Software-Bausteinen unterschiedliche Programmabläufe zusammengesetzt und auf der Festplatte als Datei abgespeichert werden. Auch in der Software müssen die Bausteine in der richtigen Reihenfolge miteinander verbunden werden, um ein Modell ansteuern zu können. Im Unterschied zu den Teilen im Baukasten kannst Du aber in der Software einen Baustein mehrfach verwenden und Dir selber neue Bausteine bauen. Wenn Du einen neu eingegebenen Programmablauf behalten willst, dann mußst Du ihn auf der Festplatte Deines PCs abspeichern. Ein LLWin-Programm heißt Projekt und hat immer die Endung .MDL.

Zwischen den Bausteinen müssen mit der Maus Verbindungen gezogen werden, ähnlich wie beim

Modell mit den „echten“ Drähten. Vom Baustein START beginnend werden die einzelnen Programmbausteine entsprechend den Verbindungen durchlaufen. Manche Bausteine können Verzweigungen realisieren oder auch auf ein Ereignis warten. Zur Steuerung von Abläufen und zur Anzeige von Zuständen können über Bausteine Variablen gesetzt, abgefragt und angezeigt werden. Das Ein- und Ausschalten der Motoren und Lampen erfolgt ebenfalls über Bausteine. Durch Rückführungen und Zählvariablen können Schleifenkonstruktionen programmiert werden.

Damit Du die Bedienung von LLWin möglichst schnell erlernen kannst, sind für die einzelnen Modelle verwendbare Beispielprojekte vorbereitet worden. Diese Projekte kannst Du starten und natürlich auch ändern und erweitern.

Bevor Du die Beispiel-Projekte ausprobierst, solltest Du zunächst die Beschreibung der Bausteine durchblättern. Du mußt nicht gleich alles lesen. Es reicht aus, wenn Du verstanden hast, wozu die einzelnen Bausteine angewendet werden. Zum schnellen Nachschlagen während des Programmierens oder auch während des laufenden Programmes kannst Du die Online-Hilfe von LLWin verwenden. Eine Online-Hilfe ist ein Handbuch auf der Festplatte. Im Unterschied zu dem auf Papier ausgedruckten Handbuch schlägt der PC automatisch die richtige Seite auf, wenn Du eine Frage hast. Es geht meistens schneller, etwas in der Online-Hilfe zu finden, als in diesem Buch danach zu suchen. Du brauchst nur die Taste <F1> auf der Tastatur zu



drücken. Statt umzublättern, klickst Du in der Online-Hilfe auf die grün hervorgehobenen Themen. Du kannst auch nach Begriffen und Themen suchen lassen. Über die Online-Hilfe kannst Du jederzeit Informationen zu Menüpunkten oder auch zu Bausteinen abfragen.

Um Dein erstes kleines Projekt eingeben und testen zu können, mußt Du:

- LLWin installieren
- das Interface an LPT1 (LPT2) anschließen
- einen Taster und einen Motor mit dem Interface verbinden
- alle im Beispiel 1 beschriebenen Arbeitsschritte am PC ausführen

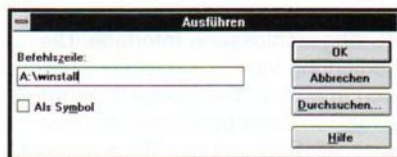
Installation von LLWin

Voraussetzung zur Installation von LLWin sind:

- ◆ ein IBM-kompatibler PC mit 486-Prozessor mit mindestens 4 MB RAM und ca. 8 MB freier Speicherkapazität auf der Festplatte (empfohlen 486-Prozessor, 66 MHz, 8 MB RAM)
- ◆ Microsoft® Windows, ab Version 3.1 oder Windows95
- ◆ Maus oder Trackball
- ◆ Druckerschnittstelle LPT1 oder LPT2

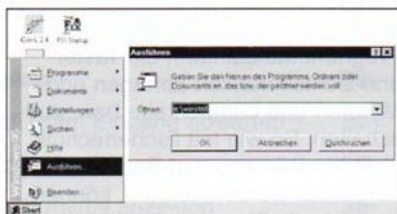
Stecke die Diskette LLWin in Laufwerk A: Wenn Dein PC nur für Laufwerk B: ein 3 1/2-Zoll-Laufwerk hat, mußt Du beim Installieren B: angeben.

Wenn auf Deinem PC Windows 3.1x installiert ist, dann wähle aus dem Programmanager im Menü **Datei** den Befehl **Programm ausführen**.



In Windows95 kannst Du auf den Startknopf klicken und dann den Befehl **Ausführen** anwählen.

Gebe



A:\WINSTALL

ein und bestätige durch einen Mausklick auf **OK** oder drücke **<Enter>**. Der PC beginnt dann die Diskette einzulesen. Wenn Du auf **Informationen** klickst, wird ein kurzer Text über LLWin angezeigt. Willst Du die Installation fortsetzen, so klicke mit der Maus auf **Installieren**.

Du wirst gefragt, ob Du eine neue Programmgruppe anlegen willst. Beantworte diese Frage mit **Ja**. Falls unter Windows 3.1x die Symbole nach der Installation übereinander angeordnet sind, dann rufe den Befehl **Automatisch anordnen** im Menü **Optionen** auf.



Nach erfolgreicher Installation wurde eine neue Programmgruppe mit dem Namen LLWin 2.03 angelegt.



Weitere Einstellungen (z.B. der verwendete Druckerport LPT1 oder LPT2) werden in LLWin im Menü **Optionen** mit dem Befehl **Interfaceeinstellungen** vorgenommen.

Aufrufen und Beenden von LLWin



Roboter

LLWin kann mit oder ohne Projekt gestartet werden. Um beim Aufrufen von LLWin gleich ein Projekt laden

zu können, muß das zu ladende Projekt mit einem Symbol verbunden sein. Dies ist für die mitgelieferten Beispielprojekte bereits erfolgt. Beim Doppelklick auf das Symbol **Roboter** wird LLWin gestartet. Es erscheint zunächst das Eröffnungsbild, das Projekt ROBOTER.MDL wird geöffnet und im Bearbeiten-Modus angezeigt.

Wenn Du das Projekt starten willst, mußt Du im Menü **Run** den Befehl **Start** aufrufen. Das geht viel schneller, wenn Du die Taste <F5> drückst.



LLWin

Wenn mit LLWin ein Projekt geöffnet werden soll, das mit keinem Symbol verbunden ist, so mußt Du einen

Doppelklick auf das Symbol mit dem Namen LLWin ausführen. Es erscheint zunächst ein Eröffnungsbild und dann eine leere Arbeitsfläche. Mit dem Befehl **Öffnen** im Menü **Datei** kannst Du ein bereits vorhandenes Projekt öffnen. Die mitgelieferten Beispielprojekte stehen alle im Verzeichnis /BEISPIEL.

Soll ein neues Projekt eingegeben werden, so kannst Du über den Befehl **Neu** ein neues Projekt eingeben (editieren). Lege Dir am besten ein eigenes Verzeichnis (z.B. mit dem Dateimanager) für die neu eingegebenen Projekte an.

LLWin kann mit dem Befehl **Beenden** im Menü **Datei** verlassen werden. Falls Du nicht zuletzt den Befehl **Speichern** oder **Speichern unter...** aufgerufen hast, erscheint eine Abfrage, ob das zuletzt geöffnete Projekt abgespeichert werden soll. Wenn Du diese Frage mit **Ja** beantwortest, so wird das Projekt gespeichert. Hat das Projekt noch keinen Namen, so wird automatisch das Dateifenster **Speichern** unter aufgerufen. Wähle das Verzeichnis und den Dateinamen, unter dem Du das Projekt speichern willst.



2. Beispiele

Beispiel 1 - Motorsteuerung

Unser erstes Programm soll einen Motor, der am Interfaceausgang M1 angeschlossen ist, mit einem Taster am Interfaceeingang E1 ein- und ausschalten. Nach dem ersten Tastendruck soll der Motor starten und nach dem zweiten Tastendruck soll der Motor stoppen.

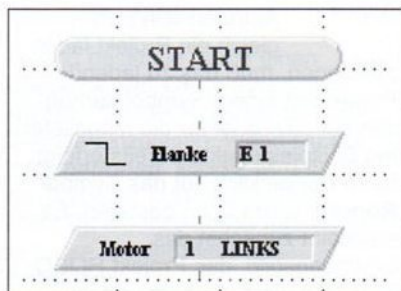
Rufe LLWin mit einem Doppelklick auf das Symbol LLWin auf. Nach dem Eröffnungsbild erscheint eine leere Arbeitsfläche. Wenn Du im Menü **Datei** auf den Befehl **Neu** klickst, wird gefragt, ob Du ein neues Projekt anlegen willst. Beantworte die Frage mit **Ja**. Dann wird das Hauptprogramm im Editierfenster angezeigt. Es ist bereits ein Baustein START vorhanden.

Zur Ansteuerung des Motors müssen die Bausteine FLANKE und MOTOR eingefügt werden. Wähle den Befehl **Baustein einfügen** aus dem Menü **Bearbeiten** oder drücke die Taste **<Einf>**. Es wird ein Fenster zur Auswahl des Bausteines angezeigt. Es ist die Bibliothek BAUSTEIN anzuklicken und der Baustein FLANKE auszuwählen. Der Name des angewählten Bausteines erscheint im Feld „Auswahl“ und wird mit Mausklick auf **OK** oder mit **<Enter>** bestätigt. Wenn Du den Baustein mit Doppelklick auswählst, dann wird der Baustein sofort eingefügt und Du kannst ihn anschließend verschieben. Einfacher geht es mit **OK** oder **<Enter>**. Der Baustein wird nun auf der Zeichenfläche mit linkem Mausklick unter dem Baustein START abgesetzt. Dabei sollte

soviel Platz zwischen den beiden Bausteinen gelassen werden, daß sich die Anschlüsse nicht berühren. Im Baustein FLANKE wird beim Einfügen die {Nummer des Einganges} abgefragt. Es ist die Nummer [1] einzugeben und mit **<Enter>** zu bestätigen.

Die Bausteine kannst Du auch später noch verschieben. Dazu mußt Du den Baustein greifen, also den Mauszeiger auf den Baustein schieben und die linke Maustaste ständig gedrückt halten. Du kannst den Baustein verschieben und dann die Maustaste loslassen.

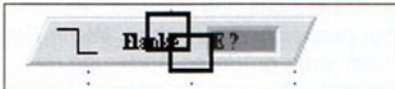
Nun wird der Baustein MOTOR eingefügt. Drücke dazu die Taste **<Einf>** und wähle MOTOR aus der Bibliothek BAUSTEIN. Die Abfrage {Nummer des Motors} wird mit [1] und die Abfrage {L(inks) / R(echts) / (Aus)} wird mit [L] für Links beantwortet.



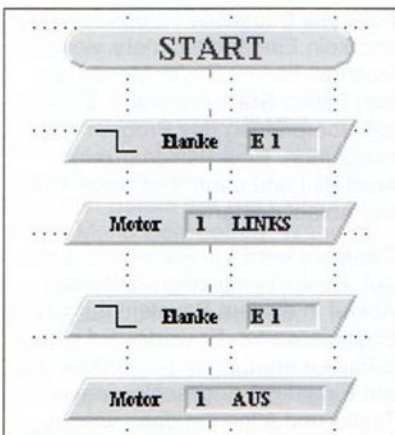
Die beiden folgenden Bausteine FLANKE und MOTOR kannst Du genauso einfügen. Etwas schneller geht es, wenn Du die Bausteine durch Kopieren der bereits eingefügten Bausteine ergänzt. Dazu mußt Du den Mauszeiger zunächst über dem Baustein FLANKE schieben. Dann drücke die Taste **<Shift>** (Umschaltung auf Großbuchstabe) und halte die Taste ständig



gedrückt. Nun kannst Du den Baustein mit der linken Maustaste greifen und auf die neue Position verschieben. Die Kopierfunktion ist an den beiden versetzten Rechtecken zu erkennen. Wichtig ist, daß Du die Taste <Shift> zuerst drückst und erst dann die linke Maustaste.



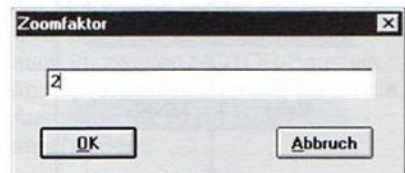
Der Baustein FLANKE wird unter dem Baustein MOTOR angeordnet. Als {Nummer des Einganges} wird wieder [1] eingegeben. Nun kannst Du den Baustein MOTOR kopieren und als {Nummer des Motors} eine [1] eingegeben. Wenn Du bei der Abfrage {L(inks) / R(echts) / (Aus)} sofort die Taste <Enter> drückst, wird automatisch AUS eingefügt. Unser Ablauf ist noch unvollständig. Er enthält schon alle Bausteine, doch es fehlen noch die Verbindungen.



Falls in einem der Bausteine noch ein Eingabefenster violett leuchtet, mußt Du die Abfragen für diesen

Baustein noch einmal aufrufen. Drücke dazu die Taste <Shift> und klicke mit der rechten Maustaste auf den Baustein mit dem leuchtenden Eingabefenster.

Beim Verbinden mußt Du sehr genau mit der Maus arbeiten. Dies geht viel einfacher, wenn Du mit der langen Taste <+> im Zifferblock rechts auf der Tastatur den Zoomfaktor (Vergrößerung) erhöhst. Mit der Taste <-> (rechts oben im Zifferblock) kannst Du den Zoomfaktor wieder verkleinern. Wenn Du ein Notebook verwendest, dann fehlt dieser Zifferblock oft. Du kannst aber den Befehl **Zoom** im Menü **Optionen** aufrufen. Gebe den Faktor 2 ein und bestätige mit <Enter>.



Zum Verbinden wird der Mauszeiger genau auf den Anschluß an der unteren Seite des Bausteines START verschoben. Der Mauszeiger wird als Stift dargestellt.

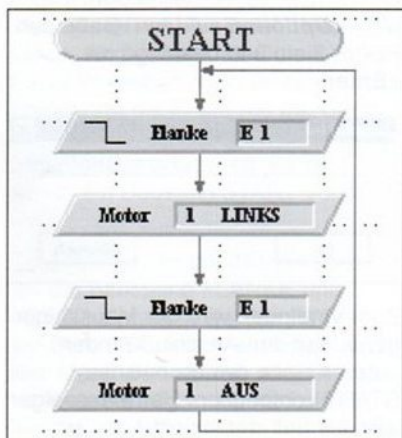


Durch Klick auf die **rechte Maustaste** wird die Verbindung begonnen. Der Mauszeiger wird nun als Fadenkreuz angezeigt. Schiebe das Fadenkreuz auf den oberen Eingang des Bausteines FLANKE. Durch einen **rechten Mausklick** wird die Verbindung hergestellt. Wenn die Verbindung gelungen ist,



sind die beiden Bausteine mit einem grünen Pfeil verbunden. Falls Du den Mauszeiger zu ungenau auf den oberen Eingang des Bausteines FLANKE geschoben hast, wird die Verbindung abgebrochen und Du mußt noch einmal am Baustein START beginnen.

Auf die gleiche Art und Weise werden alle Bausteine miteinander verbunden, bis nachfolgendes Bild entsteht. Die letzte Verbindung beendest Du mit einem rechten Mausklick auf die Mitte der Verbindung zwischen START und FLANKE.



Bevor Du Dein erstes kleines Projekt testest, solltest Du es auf der Festplatte abspeichern. Klicke mit der Maus auf den Befehl **Speichern** im Menü **Datei**. Es wird eine Dateibox mit den bereits vorhandenen Projekten angezeigt. Wähle das Verzeichnis, in dem Du das Projekt abspeichern willst. Entweder Du verwendest das Verzeichnis /BEISPIEL oder Du legst Dir ein eigenes Verzeichnis an. Gebe einen maximal 8 Zeichen langen Dateinamen an, der in der Box noch nicht enthalten

ist, z.B. MOTOR1 und bestätige mit einem linken Mausklick auf **OK**.



Tip:

Vor dem Testen solltest Du ein Projekt immer abspeichern. Falls einmal jemand Deinen PC abschaltet, nachdem Du ein großes Projekt gerade erfolgreich fertiggestellt hast, wirst Du sicherlich froh sein, es vorher gespeichert zu haben. Wenn Du nach jeder größeren Änderung das Projekt unter einem anderen Namen speicherst, kannst Du auch auf eine ältere Version zurückgreifen, wenn sich bei einer Änderung ein Fehler eingeschlichen hat.

Bei einem neu erstellten Projekt solltest Du zunächst den Befehl **Init** im Menü **Run** aufrufen. Bei Eintritt in den Init-Modus wird getestet, ob an allen Bausteinen die Anschlüsse verbunden worden sind. Nicht verbundene Bausteine leuchten violett auf. Dann mußt Du wieder zu Bearbeiten wechseln. Dies geht am schnellsten mit Taste <F6>.

Falls alle Bausteine verbunden sind und kein Eingabefeld mehr violett leuchtet, kannst Du im Menü **Run** den Befehl **Start** anwählen. Es wird die Überprüfung des Programmes ausgeführt. Eine Balkenanzeige erscheint und dann wird unser kleiner Ablauf gestartet.


Der Baustein FLANKE leuchtet rot auf. Es wird angezeigt, daß der Ablauf in diesem Baustein auf ein Ereignis wartet. Er wartet auf eine 1-0-Flanke am Eingang 1. Drücke den am Eingang 1 angeschlossenen Taster und öffne ihn dann wieder. Nach dem Öffnen des Tasters ist die Bedingung zum Weiterschalten im Baustein FLANKE erfüllt und der Baustein MOTOR wird aufgerufen.



Der Motor wird eingeschaltet und der Ablauf ruft im nächsten Schritt den unteren Baustein FLANKE auf.

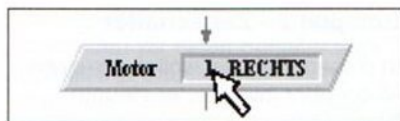
Wenn sich der Motor nicht einschalten läßt, dann überprüfe bitte die Verbindungen zwischen dem Interface und dem PC und zwischen dem Interface und dem Netzteil. Wenn die Leuchtdiode im Interface nicht leuchtet, mußt Du zuerst mit Taste <F9> den Ablauf stoppen und dann eventuell mit dem Befehl **Interfaceeinstellung** im Menü **Optionen** auf einen anderen Druckerport (z.B. auf LPT2) umschalten.

Mit der nächsten Betätigung des Tasters kannst Du den Motor wieder abschalten.

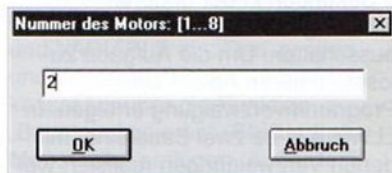
 **Hinweis:**

Da ein Schritt im Ablauf bei LLWin nur 10 Millisekunden (ms) dauert, kann der MOTOR-Baustein nicht rot dargestellt werden. Der Bildschirm wird nur alle 300 ms oder bei einem sehr langsamen PC in relativ großen Zeitabschnitten (mehrere Sekunden) aktualisiert.

Sollen in einem Projekt nur die Nummern der Eingänge und Ausgänge und die Zahlen geändert werden, so ist dies im Menü **Run** über den Befehl **Init** viel einfacher als im Menü **Bearbeiten**. Während beim Bearbeiten bei <Shift + rechter Mausklick> immer alle Abfragen eines Bausteines durchlaufen werden, können einzelne Eingabefelder im Init-Modus direkt durch Mausklick geöffnet werden. Wenn Du das gezeichnete Beispiel so ändern willst, daß der Motor 2 angesteuert wird, dann mußt Du zuerst mit Taste <F9> den Ablauf stoppen. Klicke im Init-Modus auf die Motornummer.



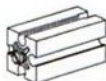
Dann öffnet sich ein Fenster und Du kannst die Motornummer 2 eintragen und mit OK bestätigen.



Die Abfrage der Drehrichtung wird nur dann aufgerufen, wenn Du zusätzlich auch auf das Wort LINKS im Baustein MOTOR klickst. Damit der Motor auch wieder abgeschaltet wird, mußt Du die Motornummer auch im zweiten MOTOR-Baustein ändern. Zum Testen mußt Du an die Ausgänge M2 einen Motor anschließen.

 **Tip:**

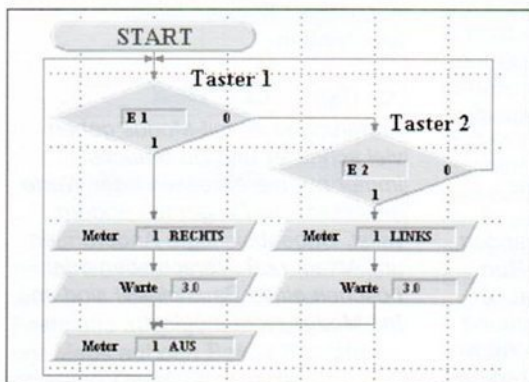
Änderungen im Init-Modus gehen viel schneller und Du brauchst immer nur die Adressen oder Werte anklicken, die Du gerade ändern willst. Unbeabsichtigte Änderungen am Ablauf (z.B. Verschieben oder Löschen eines Bausteines) sind im Init-Modus nicht möglich.



Beispiel 2 - Zeitschalter

In diesem Beispiel wollen wir einen Motor über zwei Taster steuern. Wenn Taster 1 gedrückt wird, soll sich der Motor für 3 Sekunden in der einen Richtung drehen und dann wieder ausschalten. Wenn Taster 2 gedrückt wird, soll sich der Motor für 3 Sekunden in der anderen Richtung drehen und dann wieder ausschalten. Um die Aufgabe zu lösen, muß im Ablauf eine Programmverzweigung erfolgen. In LLWin gibt es zwei Bausteine, mit denen Verzweigungen realisiert werden können. Dies sind der Baustein EINGANG und der Baustein VERGLEICH. In diesem Beispiel wollen wir eine Verzweigung mit dem Baustein EINGANG testen.

Lege mit dem Befehl **Neu** ein neues Projekt an. Gebe den nachfolgenden Ablauf ein:



Um neben den Bausteinen auch Kommentar („Taster 1“ und „Taster 2“) einfügen zu können, mußt Du den Befehl **Baustein einfügen** aufrufen und dann den Baustein Text aus der Bibliothek „Standard“ auswählen.

Nach dem Start mit Taste <F5>

leuchten die beiden EINGANG-Bausteine abwechselnd rot auf. Bis Du Taster 1 oder Taster 2 drückst, werden nur diese beiden Bausteine durchlaufen. Solange Taster 1 gedrückt wird, ist dies im linken Baustein EINGANG erkennbar. Das kleine grüne Dreieck springt von dem linken auf den unteren Ausgang. Der Motor wird eingeschaltet und die Zeitverzögerung startet.

Dieses Beispiel kann statt mit zwei auch mit einem einzigen Baustein WARTE aufgebaut werden. Probiere es einfach aus! Rufe zuerst den Befehl **Stop** aus dem **Run**-Menü auf oder drücke Taste <F9>. Mit dem Befehl **Hauptprogramm...** im Menü **Bearbeiten** wechselst Du wieder in den Editor-Modus. Schneller geht dies mit <F6> .



Um den rechten WARTE-Baustein zu löschen, schiebe den Mauszeiger auf den Baustein, drücke die Taste <Entf> und klicke mit der Maus auf den Baustein. Dann kannst Du die Löschfunktion mit <ESC> abbrechen.

Verbinde den rechten MOTOR-Baustein mit der

Mitte der Verbindung zwischen dem linken MOTOR-Baustein und dem WARTE-Baustein.

Zum Testen kannst Du diesmal sofort die Taste <F5> drücken. Bei der Verwendung von nur einem WARTE-Baustein kannst Du im Bild nicht mehr erkennen, welche



Drehrichtung gerade eingeschaltet wurde. Wenn Du den Baustein TERMINAL einfügst, wird die Drehrichtung der Motoren in der oberen Reihe angezeigt. Der Baustein TERMINAL benötigt keine Verbindungen.

Beispiel 3 - Impulszähler

Im 3. Beispiel soll die Funktion eines Zählers erläutert werden. Wenn ein Roboter seinen Greifer immer genau zwischen zwei Ablagepunkten bewegen soll, dann ist es erforderlich, den Weg elektronisch zu messen. In den fischertechnik-Modellen werden die Motoren mit speziellen Zahnrädern (Impulsscheiben) verbunden, die einen Taster bei jeder Umdrehung genau 5 mal schließen und 5 mal öffnen. Solche Signale nennt man in der Elektrotechnik Flanken oder auch Impulse.

Das Zählen der Impulse übernimmt der Baustein POSITION (Zähler, wartet bis Endwert EW). Damit Du besser verstehst, wie dieser Baustein arbeitet, wollen wir die Funktion des Bausteines POSITION mit anderen Bausteinen nachbilden. Für dieses Beispiel benötigen wir

kein Interface. Als Eingangssignal verwenden wir einen Schalter vom TERMINAL-Baustein. Ein Motor würde die Taste so schnell öffnen und schließen, daß es nicht möglich ist, am Bildschirm im Ablaufplan die Funktion zu verfolgen.

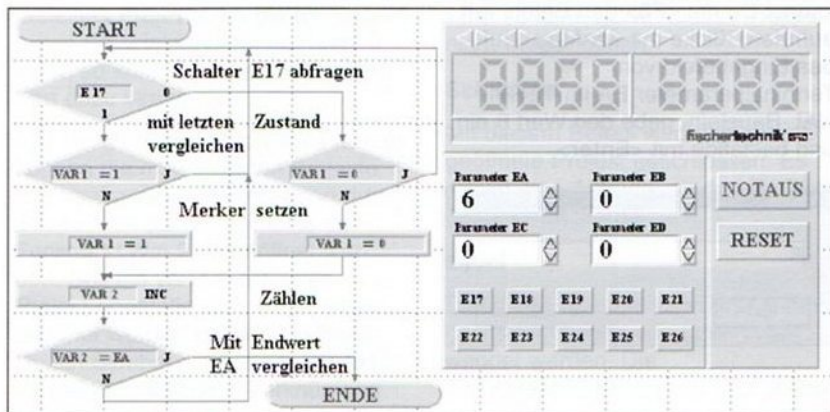
Rufe im **Menü** Datei den Befehl **Neu** auf und füge nacheinander die Bausteine EINGANG, VERGLEICH und VARIABLE ein. Den zweiten und dritten VERGLEICH-Baustein kannst Du über die Funktion „Baustein kopieren“ (**Shift + linke Maustaste**) einfügen. Außerdem benötigst Du noch die Bausteine INCVARIABLE, TERMINAL und ENDE.

Beantworte die Abfragen und verbinde die Bausteine wie im Bild dargestellt. Die Bausteine kannst Du auch nach dem Verbinden verschieben. Die Verbindungen bleiben dabei bestehen.



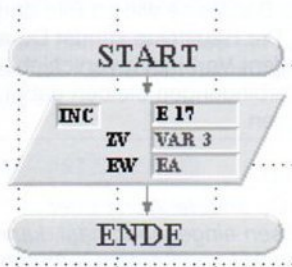
Tip:

Falls Du andere Werte oder Adressen eingegeben hast, kannst Du die Abfragen erneut aufrufen. Dazu mußt Du die Taste <Shift> drücken und mit der rechten Maustaste auf die Bausteine klicken.

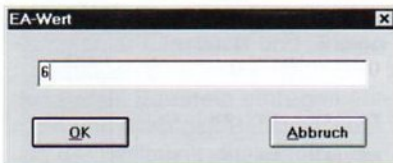


Im Baustein **EINGANG** ist die Nummer 17 anzugeben. Dies ist der Schalter **E17** auf dem **TERMINAL**. Im unteren **VERGLEICH**-Baustein wollen wir **EA** verwenden. **EA** ist ein Terminalparameter, dessen Wert Du auch während des laufenden Programmes ändern kannst.

Außerdem soll parallel zu unserem ersten Ablauf der Baustein **POSITION** abgearbeitet werden. Füge die folgenden drei Bausteine ein und verbinde sie. Als Zählvariable gebe die Nummer 3 ein. Die Variablen 1 und 2 werden im ersten Ablauf verwendet. Wenn Du die gleichen Variablen im zweiten Ablauf verwenden würdest, dann beeinflussen sich die beiden Abläufe gegenseitig.



Speichere Dein Projekt z.B. unter dem Namen **POSIT3** ab. Nun kannst Du mit **<F8>** den Befehl **Init** aufrufen. Klicke mit der Maus auf das Eingabefeld vom Terminalparameter **EA** im **TERMINAL**-Baustein, gebe den Wert 6 ein und bestätige mit **<Enter>**.



Wenn alle Bausteine richtig verbunden sind, dann drücke **<F5>**. Beobachte die Anzeige im Baustein **INCVARIABLE** und klicke mehrfach auf den Schalter **E17**. Vergleiche, ob die beiden Abläufe nach der gleichen Anzahl von Mausklicks in den Baustein **ENDE** schalten.

Um den Ablauf neu zu starten, kannst Du auch auf den Schalter **RESET** im **TERMINAL** klicken. Es leuchten die beiden **START**-Bausteine auf. Wenn Du den Schalter **RESET** rücksetzt, starten die beiden Abläufe wieder neu. Versuche zu verstehen, warum in **VAR 1** der Zustand des Eingangs **E17** zwischengespeichert werden muß. Beobachte dazu die grünen Dreiecke in den Bausteinen **EINGANG** und **VERZWEIGUNG** und versuche, den Ablauf Schritt für Schritt nachzuvollziehen.



3. Menübefehle

Datei

Wenn noch kein Projekt geöffnet wurde, sind nur die Befehle **Neu**, **Öffnen** und **Beenden** anwählbar.

Neu

Mit diesem Befehl wird ein Projekt unter dem Namen NEU.MDL angelegt. Das Projekt enthält bereits einen Baustein START. Um weitere Bausteine einzufügen, drücke die Taste **<Einfg>**.

Wenn Du den Befehl **Speichern** aufrufst, wird automatisch in das Fenster Speichern unter gewechselt. Wenn Du den Befehl **Beenden** aufrufst und die Frage, ob das Projekt abgespeichert werden soll, mit **Ja** beantwortest, wird ebenfalls automatisch das Dateifenster Speichern unter aufgerufen.

Öffnen

Mit dem Befehl **Öffnen** kann ein bereits auf der Festplatte vorhandenes Projekt geöffnet werden. Wenn Du ein auf der Festplatte gespeichertes Projekt (z.B. ein selbst erstelltes Projekt oder die mitgelieferten Beispielprojekte) aufrufen willst, so kannst Du den Befehl **Öffnen** dazu verwenden.

Aus der Windows-Oberfläche heraus können die mitgelieferten Projekte auch beim Starten von LLWin durch Doppelklick auf die Symbole im Fenster LLWin 2.02 geöffnet werden. Soll ein von Dir erstelltes Projekt ebenfalls über ein Symbol geöffnet werden können, so mußt Du ein Symbol mit diesem Projekt verknüpfen. Wie eine Verknüpfung

mit einem Symbol erstellt wird, kannst Du im Windows- Handbuch nachlesen.

Speichern <F10>

Mit diesem Befehl wird das gerade geöffnete Projekt abgespeichert. Der Befehl Speichern kann auch über die Funktionstaste **<F10>** aufgerufen werden. Falls es sich um ein mit dem Befehl Neu erzeugtes Programm handelt, wird automatisch der Befehl Speichern unter... ausgeführt.

Speichern unter...

Mit diesem Befehl wird das gerade geöffnete Projekt unter einem neuen Namen abgespeichert. Es erscheint das Dateiauswahlfenster Projekt speichern unter. Es kann ein bereits vorhandenes Projekt angewählt werden oder es wird ein neuer Projektname eingegeben. Wenn ein bereits vorhandener Projektname eingegeben wird, erfolgt eine Abfrage, ob die vorhandene Datei überschrieben werden soll.

Der Name darf maximal 8 Zeichen lang sein. Die Dateierdung .MDL wird automatisch ergänzt. Wenn eine andere Endung angegeben wird, so kann LLWin dieses Projekt nicht wieder öffnen.

Schließen

Mit diesem Befehl wird das gerade geöffnete Projekt geschlossen. Es erfolgt eine Abfrage, ob das Projekt gespeichert werden soll.

Level

Dieser Menübefehl kann erst in der nächsten Version von LLWin verwendet werden. Die Softwareentwicklung der fischerwer-



ke arbeitet an zusätzlichen Bibliotheken, um neue Bausteine für Anzeigen, Texteingaben und ... - alles wird hier noch nicht verraten.

Oder hast Du eine tolle Idee ? Dann schreibe an die Redaktion des fischertechnik-Fanclubs. Die neuen Bibliotheken werden auch zukünftig nicht in der Grundversion enthalten sein. Über Preis und Lieferertmin werden wir Dich rechtzeitig informieren. Deshalb bitte die Registrierung ausfüllen und abschicken.

Seite drucken

Das gerade auf dem Bildschirm angezeigte Haupt- oder Unterprogramm kann mit diesem Befehl ausgedruckt werden. Die Größe der Bausteine im Druckbild wird mit dem Befehl **Arbeitsblatt** im Menü **Optionen** eingestellt.

Projekt drucken

Werden in einem Projekt Unterprogramme verwendet und sollen das Hauptprogramm und alle Unterprogrammaufrufe ausgedruckt werden, so kann hierzu der Befehl **Projekt drucken** verwendet werden. Um den Befehl aufrufen zu können, müssen alle Fenster geschlossen werden. Hast Du in einem Projekt ein Unterprogramm mehrfach aufgerufen, wie dies z.B. im Beispielprojekt Roboter der Fall ist, dann wird dieses Unterprogramme mehrfach gedruckt. In diesem Fall solltest Du, um Papier und Zeit zu sparen, den Befehl **Seite drucken** verwenden.

Bearbeiten

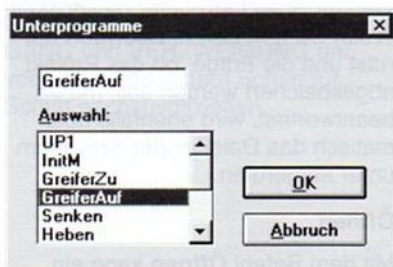
Hauptprogramm... <F6>

Dieser Befehl wird nach dem Öffnen

eines Projektes automatisch aufgerufen. Du kannst Bausteine einfügen und verbinden, sowie Bausteine und Verbindungen löschen und verschieben. Wenn ein Projekt im **Run**-Modus abgearbeitet wird, so muß Du zuerst **Stop** aufrufen oder die Taste <F9> drücken. Erst dann kannst Du über die Taste <F6> oder über den Befehl **Hauptprogramm...** Änderungen am Hauptprogramm vornehmen.

Unterprogramm... <F7>

Mit diesem Befehl kannst Du Unterprogramme eingeben. Es erscheint zunächst ein Fenster zur Auswahl des Unterprogrammes.



Beim Bearbeiten eines Unterprogrammes sind die gleichen Bedienschritte möglich wie bei der Eingabe des Hauptprogrammes. Zusätzlich können bei der Eingabe des Unterprogrammes die Bausteine Input und Output aus der Bibliothek Standard eingefügt werden. Diese Bausteine werden benötigt, wenn die Unterprogrammaufrufe mit anderen Bausteinen des Projektes verbunden werden sollen.

 **Hinweis:**

Beachte den Unterschied zwischen dem Unterprogrammaufruf und dem Unterprogramm. Der Unterpro-



grammaufruf ist ein Baustein und kann beliebig oft eingefügt werden. Das Unterprogramm existiert nur einmal. Es kann weitere Unterprogrammaufrufe und Bausteine enthalten.

In einem neu erstellten Projekt ist nur das Unterprogramm UP1 vorhanden. Du kannst einen neuen Unterprogrammnamen angeben, das Unterprogramm UP1 editieren oder aber Du nutzt es als Kopiervorlage. Weitere Hinweise zur Verwendung von Unterprogrammen findest Du im Kapitel 5.

Rückgängig

Dieser Befehl ist nur für Verschiebeoperationen verfügbar. Unmittelbar nach einer Verschiebung eines Bausteines, einer Verbindung oder nach dem Einfügen oder Löschen einer Spalte oder Zeile (siehe Anhang) kann der alte Zustand wieder hergestellt werden.

Baustein einfügen <Einf>

Mit diesem Befehl werden Bausteine eingefügt. Es erscheint das Auswahlfenster für die Bausteine.



Die Bibliothek BAUSTEIN enthält die zur Programmierung von Abläufen verfügbaren Bausteine. In der Bibliothek Projekt sind die im Projekt vorhandenen Unterprogrammaufrufe abgespeichert. Die Bibliothek Standard enthält die Kommentarbausteine und die für die Anschlüsse von Unterprogrammen benötigten Bausteine Input und Output.

Nach Anwahl der Bibliothek muß Du aus dem Feld **Baustein** den gewünschten Baustein wählen. Über die Schaltfläche **Hilfe** kann die Online-Hilfe zum Baustein angewählt werden. Durch Mausklick auf die Schaltfläche **OK** oder durch Betätigung von <Enter> bestätigst Du die Auswahl. Nun kannst Du den Baustein auf der Zeichenfläche platzieren. Anschließend werden die Eingabedialoge für den Baustein angezeigt. In diesen Eingabedialogen werden die Adressen und Konstanten eingegeben.

Unterprogrammaufrufe werden wie Bausteine eingefügt. In einem neu angelegten Projekt ist nur das als Kopiervorlage dienende Unterprogramm UP1 enthalten.

Kommentare zur Erläuterung der Funktion des Programmablaufes werden ebenfalls wie Bausteine eingefügt. Dazu ist aus der Bibliothek Standard der Baustein Text auszuwählen.

Baustein löschen <Entf>

Mit diesem Befehl wird der Mauszeiger umgeschaltet. Er nimmt die Form eines Hammers an. Alle mit dem Hammer angeklickten Bausteine werden gelöscht. Der Befehl **Rückgängig** ist nach dem Löschen von Bausteinen nicht ver-



fügar. Deshalb solltest Du vor dem Löschen von Bausteinen das Projekt speichern. Wurde ein Baustein unbeabsichtigt gelöscht, kannst Du dann die zuletzt abgespeicherte Version wieder laden.

Sollen Verbindungen gelöscht werden, so ist hierzu kein Menübefehl vorhanden. Du mußt die Taste <Strg> gedrückt halten und dann auf die zu löschende Verbindung mit der linken Maustaste klicken. Es können auch Verbindungsabschnitte gelöscht werden. Dazu ist bei gedrückter <Strg>-Taste auf den Ausgang des Bausteines zu klicken. Ein Verbindungsabschnitt ist ausgewählt, wenn bei aktivierter Cursorumschaltung (Menü Optionen) der Mauszeiger als Stift angezeigt wird.

Unterprogramm

Kopieren

Mit diesem Befehl kannst Du ein Unterprogramm auf einen neuen Namen kopieren. Dies ist dann vorteilhaft anwendbar, wenn sich Unterprogramme nur in wenigen Bausteinen unterscheiden. In einem neu eröffneten Projekt ist das Unterprogramm UP1 als Kopiervorlage enthalten.

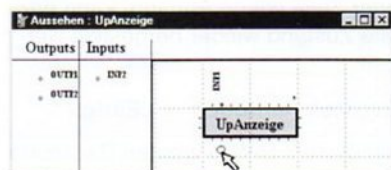


Tip:

Um UP1 auf einen neuen Namen zu kopieren, wähle den Befehl Kopieren im Menü Unterprogramme. Wenn Du mit UP1 oder mit einer Kopie arbeitest, dann sind schon zwei Anschlüsse am Unterprogrammaufruf vorhanden.

Aussehen

Mit dem Befehl **Aussehen** kannst Du die Größe des Unterprogrammaufrufes ändern. Hierzu mußt Du mit der linken Maustaste auf die rechte untere Rahmenecke klicken und dann mit gedrückter Maustaste den Rahmen auf die gewünschte Größe ziehen. Falls Du den Unterprogrammaufruf schon verwendet hast, kannst Du die Größe des Rahmens nicht mehr ändern. Sonst werden alle Unterprogrammaufrufe aus dem Projekt entfernt. Es wird eine entsprechende Meldung angezeigt. Wenn Du bei der Bearbeitung des Unterprogrammes Input- und Output-Bausteine eingefügt hast, müssen diese Inputs und Outputs auf dem Rahmen des Unterprogrammaufrufes platziert werden.



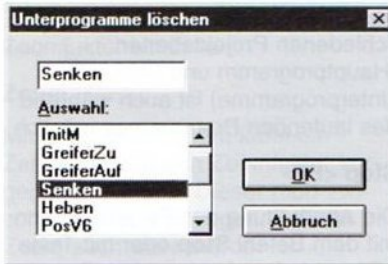
Dazu werden die Inputs und Outputs aus der links angeordneten Tabelle mit der Maus gegriffen und auf dem Unterprogramm-Rahmen an der gewünschten Position abgesetzt. Die Positionen der Inputs und Outputs stellen die Ein- und Ausgänge dar, über die der Unterprogrammaufruf mit anderen Bausteinen verbunden werden kann.

Löschen

Um ein Unterprogramm löschen zu können, mußt Du zuvor im Menü **Fenster** der Befehl **Alle schließen** aufrufen. Dann kannst Du den



Befehl **Löschen...** ausführen. Es erscheint das Fenster mit den im geöffneten Projekt enthaltenen Unterprogrammen.

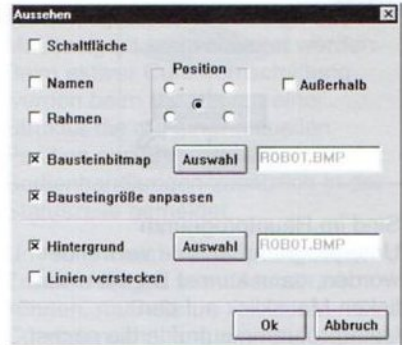


Wähle das Unterprogramm aus, und bestätige mit **OK**.

Beachte bitte den Unterschied zwischen dem Befehl **Baustein löschen** im Menü **Bearbeiten** und dem Befehl **Löschen...** im Menü **Unterprogramm**. Mit dem Befehl **Baustein löschen** können Bausteine und Unterprogrammaufrufe gelöscht werden. Dagegen wird mit dem Befehl **Löschen** im Menü **Unterprogramm** das Unterprogramm selbst mit allen in diesem Unterprogramm enthaltenen Bausteinen und Verbindungen gelöscht. Zusätzlich werden auch alle Unterprogrammaufrufe des gelöschten Unterprogrammes im gesamten Projekt entfernt (siehe auch das Bild in Kapitel 5).

Attribute

Mit dem Befehl **Attribute** kannst Du das Aussehen eines Unterprogrammaufrufes verändern. Statt des Bausteinnamens kann dieser z.B. eine Bitmap anzeigen. Der Befehl kann nur aufgerufen werden, wenn Du zuvor den Befehl **Aussehen...** aufgerufen hast. Es wird das folgende Fenster geöffnet:



Wenn eine Bitmap zugewiesen wird, dann wird diese in dem Unterverzeichnis Bitmap im Projektverzeichnis abgespeichert. Um eine Bausteinbitmap sichtbar zu machen, mußt Du die Schaltfläche und eventuell den Namen abschalten (siehe Abbildung).

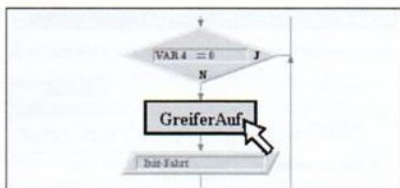
Run

Das Menü **Run** dient zum Übersetzen und Testen des geladenen Projektes. Das Einfügen und Löschen von Bausteinen und Änderungen an den Verbindungen sind im Run-Menü nicht möglich.

Init <F8>

Nach Aufruf von **Init** wird geprüft, ob alle Bausteine miteinander verbunden sind. Die nicht verbundenen Bausteine werden violett angezeigt. Außerdem kannst Du im Init-Modus die Nummern und Zahlenwerte in den Bausteinen verändern. Dazu mußt Du auf die Eingabefelder in den Bausteinen klicken. Es können alle Eingabefelder einzeln angewählt werden. Die gleichen Änderungen sind auch im Menü **Bearbeiten** möglich. Im Init-Modus geht dies aber schneller.

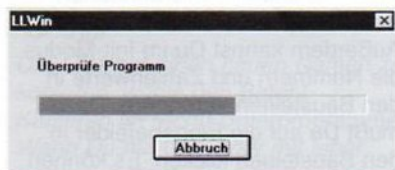




Sind im Hauptprogramm Unterprogrammaufrufe verwendet worden, dann kannst Du mit einem linken Mausklick auf den Unterprogrammaufruf in die nächst tiefere Programmebene gelangen. Die Unterprogrammaufrufe haben einen schwarzen Rahmen. Mit einem rechten Mausklick gelangst Du zurück in die nächst höhere Ebene. Vor dem Start kann mit jedem Mausklick auf den Befehl **Init** ein Run-Fenster geöffnet werden. Mit dem Befehl Fenster nebeneinander sind alle Fenster am Bildschirm sichtbar. So können Unterprogramme und das Hauptprogramm gleichzeitig angezeigt werden.

Start <F5>

Mit diesem Befehl wird die Übersetzung des Projektes gestartet und dann das Projekt abgearbeitet. Bei sehr langsamen PCs und großen Projekten kann dies etwas länger dauern. Während des Übersetzens erscheint eine Balkenanzeige.



Nach dem Übersetzen wird der Ablauf gestartet. Die Bausteine, die gerade abgearbeitet werden, sind rot dargestellt. Einige Bausteine zeigen zusätzlich den Wert der

Variablen oder des Einganges an. Wenn das Programm abgearbeitet wird, können nur die Schalter und Parameterfelder am TERMINAL bedient werden. Das im Init-Modus beschriebene Wechseln in die verschiedenen Projektebenen (Hauptprogramm und Unterprogramme) ist auch während des laufenden Programmes möglich.

Stop <F9>

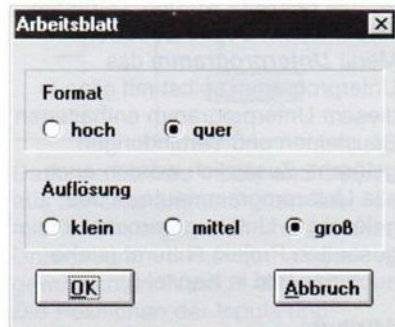
Die Abarbeitung des Projektes kann mit dem Befehl **Stop** oder mit Taste <F9> unterbrochen werden. Dabei wird wieder in den Init-Modus gewechselt.

Optionen

Arbeitsblatt

Die Einstellung des Arbeitsblattes bezieht sich auf das Druckbild.

Es kann Hoch- oder Querformat ein-



gestellt werden. Mit der Auflösung wird eingestellt, wie groß ein Baustein im Druckbild erscheint. Nur bei Nutzung eines hochauflösenden Druckers kannst Du die Auflösung auf „klein“ einstellen. Aber schon bei der großen Auflösung können bis ca. 60 Bausteine auf einem Blatt dargestellt werden. Beim Bearbeiten



wird die eingestellte Blattgröße angezeigt. Wenn Du die Blattgröße im Verhältnis zum aktuellen Programmabschnitt sehen willst, dann mußt Du mit der Taste „-“ im Ziffernblock einen kleinen Zoomfaktor einstellen.

Zoom

Mit dem Befehl Zoom kann ein Zahlenwert für den Zoomfaktor vorgegeben werden. Dieser muß zwischen 0.2 und 10 liegen. Soll der Zoom auch während der Abarbeitung des Programmes verändert werden, so können hierzu die Tasten „+“ und „-“ auf dem rechts auf der Tastatur befindlichen Ziffernblock verwendet werden. Die aktuelle Zoom-Einstellung wird beim Beenden von LLWin gespeichert und beim nächsten Aufruf voreingestellt.

Anschlüsse

Mit der Option **Anschlüsse** kann die Anzeige der Anschlußbezeichnungen an den Ein- und Ausgängen der Bausteine aktiviert werden. Diese Option wird nur benötigt, wenn Du Unterprogrammaufrufe mit mehreren Anschlüssen verbinden willst.

Statuszeile

Mit der Option **Statuszeile** kann die Statuszeile aktiviert, bzw. deaktiviert werden. Die Statuszeile meldet die im aktuellen Zustand ausführbaren Funktionen. Nach erfolgter Einarbeitung kann durch Deaktivierung der Statuszeile die Arbeitsfläche vergrößert werden.

Cursorumschaltung

Mit der Option **Cursorumschaltung** kann im Editiermodus auf eine

zustandsgesteuerte Anzeige des Mauszeigers umgeschaltet werden. Beim aktiver Cursorumschaltung werden beim Bearbeiten einer Struktur die auf einer aktuellen Position ausführbaren Bedienhandlungen zusätzlich in der Statuszeile gemeldet.

Um beim Bearbeiten Spalten oder Zeilen einfügen bzw. löschen zu können, muß die Cursorumschaltung aktiv sein. Diese Funktion kannst Du verwenden, um die Bausteine ab einer bestimmten Linie nach rechts bzw. nach unten zu verschieben. Spalten bzw. Zeilen werden eingefügt, wenn Du beim Bearbeiten die Taste <Shift> drückst und dann mit der rechten bzw. linken Maustaste klickst. Zum Löschen muß Du die Taste <Strg> drücken und dann mit der rechten bzw. linken Maustaste klicken. Spalten und Zeilen können nur eingefügt bzw. gelöscht werden, wenn sich auf der gedachten Trennlinie keine Bausteine befinden. Der Mauszeiger meldet, ob das Einfügen einer Spalte bzw. Zeile möglich ist.

Autorouting

Beim Autorouting zeichnet der Computer automatisch die günstigsten Verbindungslinien, wenn Du zwei Bausteine miteinander verbindest bzw. einen Baustein verschiebst. Im Menü **Optionen** kann mit dem Befehl **Autorouting** ein Fenster geöffnet werden. In diesem Fenster kann das Autorouting aktiviert bzw. deaktiviert werden. Die Deaktivierung des Autoroutings ist bei sehr langsamen PCs manchmal erwünscht.

Außerdem kann das Kreuzen von Textbausteinen verboten werden.



Der dann durch die Textbausteine gesperrte Bereich ist beim Verschieben der Textbausteine sichtbar. In diesem gesperrten Bereich können keine Verbindungen begonnen oder beendet werden.

Findet das Autorouting keine rechtwinklige Verbindung außerhalb der gesperrten Bereiche anderer Bausteine und Verbindungen, so wird die Verbindung als Linie ohne Knickpunkte zwischen den Anschlüssen dargestellt. Diese dann als schräge Linie angezeigte Verbindung ist funktionsfähig, es werden jedoch keine Pfeile für die Richtung dargestellt. Die an der Verbindung beteiligten Bausteine sind so zu verschieben, bis das Autorouting eine rechtwinklige Verbindung herstellen kann. Eine schräge Verbindung kann nur unmittelbar am Eingang oder am Ausgang gelöscht werden. Durch Klick mit der linken Maustaste auf einen Ausgang mit schräger Verbindungslinie kann diese in eine rechtwinklige Darstellung mit einem Knickpunkt umgewandelt werden.

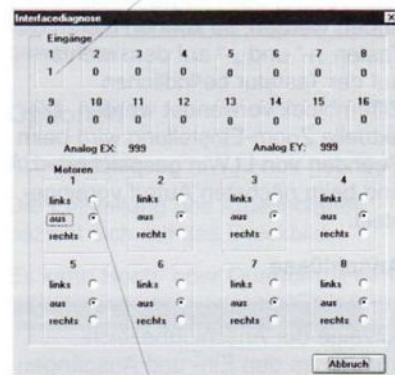
Interfacediagnose

Zur Überprüfung des Interfaces rufe im Menü **Optionen** den Befehl **Interfacediagnose** auf. Prüfe, ob die Anzeigen für die verwendeten Eingänge auf 1 schalten, wenn Du die angeschlossenen Taster drückst, bzw. wenn der Fototransistor einer Lichtschranke mit einer Lampe angeleuchtet wird.



Die Motoren können ebenfalls über das Fenster Interfacediagnose angesteuert werden. Dazu mußst Du in das Feld „links“ bzw. „rechts“ klicken. Du kannst aber auch den gerade angewählten Motor mit den Cursortasten der Tastatur schalten. Die Anwahl der Motornummer erfolgt durch Mausklick oder über die Taste <Tabulator> bzw. <Shift+Tabulator>.

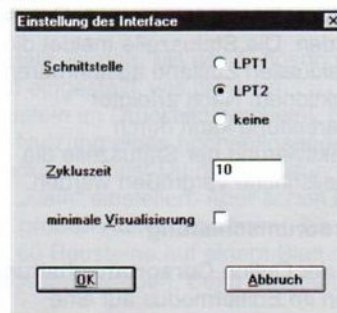
Anzeige
Eingang 1



Motor 1
einschalten

Interfaceeinstellung

Mit dem Befehl **Interfaceeinstellung...** kannst Du das nachfolgende Fenster öffnen:



Es kann die Schnittstelle LPT1 bzw. LPT2 oder „keine“ ausgewählt werden. Die Voreinstellung ist LPT1. Wenn Dein PC nur einen Anschluß für den Drucker hat, dann mußt Du LPT1 anwählen, um das Interface ansteuern zu können.

Die Zykluszeit ist auf 10 ms voreingestellt. Bei größeren Programmen und sehr langsamen PCs kann es erforderlich werden, die Zykluszeit zu vergrößern, da sonst die Windowsoberfläche zu wenig Rechenzeit erhält. Wenn Du auf eine Tastatureingabe mehrere Sekunden warten mußt, dann solltest Du versuchen, dieses Programm mit einer größeren Zykluszeit zu betreiben. Bei einer größeren Zykluszeit kann es beim Zählen von Impulsen zu Fehlern kommen. Dann zeichnet z.B. der Plotter nicht immer das gleiche Bild.

Wenn Du die „minimale Visualisierung“ aktivierst, werden die Variablen nicht mehr in den Bausteinen angezeigt. Damit wird mehr Rechenzeit für die Aktualisierung des Bildschirms und für die Abfrage der Tastatur und der Maus verfügbar.

Soll ein Programm ohne Interface ausprobiert werden, kannst Du bei Schnittstelle „keine“ eintragen. Dann kann die Zykluszeit auf 1000 ms (= 1 Sekunde) gesetzt werden. Bei einer Zykluszeit von 1000 ms kannst Du besser beobachten, wann die einzelnen Bausteine aufgerufen werden.

Fenster

Überlappend

Alle geöffneten Fenster werden so dargestellt, daß sie sich überlappen.

Die Fensterüberschriften aller Fenster sind sichtbar.

Nebeneinander

Alle geöffneten Fenster werden nebeneinander bzw. ab 4 Fenster neben- und untereinander angeordnet.

Symbole anordnen

Wenn Du Fenster zu Symbolen verkleinert hast, werden die Symbole mit diesem Befehl neu angeordnet.

Schließen

Wenn Du diesen Befehl aufrufst, wird das aktuelle Fenster geschlossen.

Alle schließen

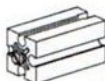
Wenn Du ein Unterprogramm löschen willst, mußt Du vorher den Befehl **Alle schließen** aufrufen. Es werden alle geöffneten Bearbeiten-Fenster bzw. Run-Fenster geschlossen.

Anwählen der geöffneten Fenster (1, 2, 3)

Wenn Du mehrere Fenster geöffnet hast und im Vollbild arbeitest, dann kannst Du im Menü Fenster mit der Maus oder über die Tastatur auf ein anderes Fenster umschalten. Am schnellsten geht dies, wenn Du die Tasten <Alt+F> und dann die Ziffer 1, 2, 3 ... drückst.

Kopieren zur Ablagemappe

Bei Ausführung dieses Befehles wird der Inhalt des aktuellen Fensters in die Zwischenablage kopiert. Die Zwischenablage kann von Zeichenprogrammen und Textverarbeitungsprogrammen eingelesen werden.

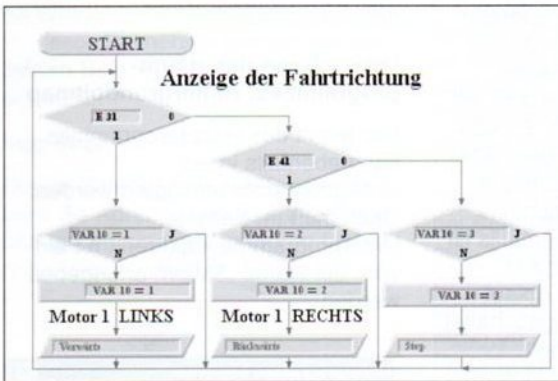


Auch dem Hauptprogramm kann eine Hintergrundbitmap zugeordnet werden. Dazu muß der Befehl Aussehen im Menü Unterprogramm aufgerufen werden und der Name des Hauptprogrammes (\$Main) eingegeben werden. Danach wird dann der Befehl Attribute aufgerufen.

Rücklesen der Motorzustände

Soll ein Ablauf über den aktuellen Zustand eines anderen Ablaufes gesteuert werden, so kann dies über Variablen oder auch über die Motorzustände erfolgen. Dazu werden die Motorzustände der Motoren 1 ... 8 auf die Eingänge 31 ... 38 (linke Drehrichtung) bzw. auf die Eingänge 41 ... 48 (rechte Drehrichtung) umkopiert. Damit können die Bausteine EINGANG, FLANKE und POSITION auch die Motorzustände abfragen.

Beispiel:



In der Abbildung wird für ein Modell die Fahrtrichtung im Baustein TERMINAL angezeigt. Der Meldetext wird immer nur dann aktualisiert, wenn sich der Zustand von Motor 1 ändert. Soll keine weitere Meldung in diesem Projekt angezeigt werden, können die Bausteine VERGLEICH und VARIABLE entfallen.

5. Verwendung von Unterprogrammen

Grundlagen zur Unterprogrammtechnik in LLWin

Unterprogramme können verwendet werden wenn:

- ein Projekt sehr viele Bausteine enthält und nicht mehr übersichtlich in einem Bild dargestellt werden kann
- der Abschnitt eines Ablaufes mehrfach verwendet werden soll
- als Bitmap vorliegende Bilder im Projekt angezeigt werden sollen

Unterprogramme werden nur dann abgearbeitet, wenn die zugehörigen Unterprogrammaufrufe im Hauptprogramm eingefügt worden

sind. Unterprogramme können weitere Unterprogramme aufrufen. Soll in einem Unterprogramm ein Ablauf aus dem Hauptprogramm fortgesetzt werden, muß eine Verbindung über die Bausteine Input und Output hergestellt werden. In Unterprogrammen können auch weitere parallel arbeitende Abläufe

ohne Verbindung zum Hauptprogramm eingesetzt werden. Unterprogramme mit eigenständigen Abläufen sollten jedoch nur einmal im Projekt aufgerufen werden.

Wenn ein Unterprogramm mit anderen Adressen und Werten angewendet werden soll, muß Du das

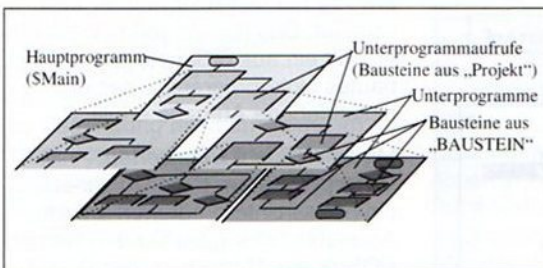


Unterprogramm auf einen anderen Namen umkopieren und dann die Adressen und Werte ändern.

In manchen Hochsprachen können Rekursionen realisiert werden, das heißt in einem Unterprogramm ist ein Aufruf des gleichen oder eines höherliegenden Unterprogrammes enthalten. In LLWin können keine Rekursionen realisiert werden - es können immer nur die Unterprogrammaufrufe eingefügt werden, die keine Rekursion auslösen.

Projektaufbau bei Verwendung von Unterprogrammen

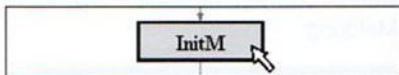
Nachfolgendes Bild veranschaulicht die Hierarchieebenen bei der Anwendung von Unterprogrammen. Es wird zwischen dem Unterprogramm und dem Unterprogrammaufruf unterschieden. Der Unterprogrammaufruf liegt im Projekt immer eine Ebene höher. Der Unterprogrammaufruf ist ein Baustein aus der Bibliothek „Projekt“ und wird wie die anderen Bausteine beim Bearbeiten eingefügt und verbunden.



Um einen Unterprogrammaufruf mit anderen Bausteinen verbinden zu können, muß dieser Ein- und Ausgänge haben. Dazu werden die Bausteine Input und Output aus der Bibliothek Standard beim Bearbeiten des Unterprogrammes eingefügt.

Außerdem müssen die Anschlüsse des Unterprogrammaufrufes auf dem Unterprogramm-Rahmen platziert werden. Dies geschieht mit dem Befehl **Aussehen...** im Menü **Unterprogramm** (siehe Kapitel 3).

Um ein Unterprogramm im Run-Modus anzuzeigen, muß Du mit der linken Maustaste in den Unterpro-



grammaufruf klicken.

Unterprogrammaufrufe sind die zu einem Unterprogramm gehörenden Bausteine und haben einen schwarzen Rahmen. Um zurück in die aufrufende Ebene zu gelangen, muß Du die rechte Maustaste drücken.

Bearbeiten und Aussehen von Unterprogrammen

Es können beim Bearbeiten auch Unterprogramme editiert werden, die nicht im aktuellen Hauptprogramm aufgerufen werden. Dies ist bei der Eingabe eines neuen Unterprogrammes immer der Fall. Erst nach

Einfügen des Unterprogrammaufrufes im Hauptprogramm bzw. in einem anderen vom Hauptprogramm aufgerufenen Unterprogramm wird das Unterprogramm abgearbeitet.

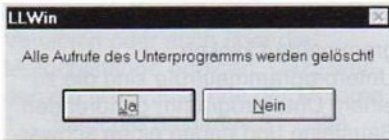
Wenn Du ein neues Unterprogramm anlegst,

indem Du im Fenster Unterprogramme einen neuen Namen einträgst, dann hat dieses Unterprogramm noch keine Anschlüsse. Du mußst zunächst die Bausteine Input und Output aus der Bibliothek Standard einfügen und



dann den Befehl **Aussehen** im Menü **Unterprogramm** aufrufen (siehe Kapitel 3).

Das Löschen von Input- und Output-Bausteinen eines Unterprogramms im Bearbeiten-Modus führt immer zum Entfernen aller bereits verwendeten Unterprogrammaufrufe. Ist der Unterprogrammaufruf bereits verwendet worden, erscheint folgende Meldung:



Wenn Du mit **Ja** bestätigst, werden die bereits verwendeten Aufrufe dieses Unterprogramms aus dem gesamten Projekt gelöscht.

Das Einfügen des Unterprogramms erfolgt mit dem **Befehl Baustein** einfügen oder mit der Taste **<Einf>**. Es ist die Bibliothek Projekt anzuwählen:



Wenn Du ein neues Projekt erstellst, dann ist in der Bibliothek Projekt ein vorbereitetes Unterprogramm UP1 enthalten. Dieses Unterprogramm

hat einen Eingang und einen Ausgang, enthält aber noch keine Bausteine. Wenn Du ein Unterprogramm mit einem Eingang und einem Ausgang erstellen willst, dann solltest Du das vorbereitete Unterprogramm UP1 mit dem Befehl **Kopieren...** im Menü **Unterprogramm** (siehe Kapitel 3) auf einen anderen Namen kopieren. Dieser Schritt vereinfacht die Einbindung von Unterprogrammen.

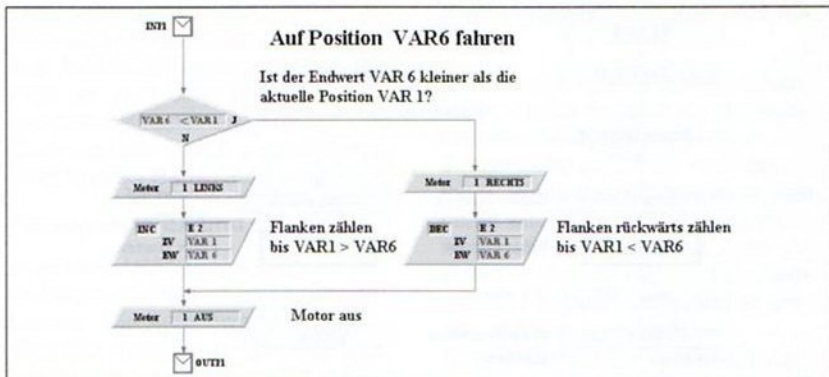
Unterprogramme können auch zum Einbinden von Bildern in Form von Bitmaps verwendet werden. Wie das gemacht wird, kannst Du im Kapitel 3 Unterprogramm/Aussehen nachlesen.

Beispiel für ein Unterprogramm

Eine oft benötigte Anwendung für ein Unterprogramm ist die Ansteuerung eines Motors, der mit der Impulsscheibe einen Taster betätigt. Für diesen Aufbau wird ein Programmabschnitt benötigt, der den Motor einschaltet, bis der Zähler einen vorgegebenen Sollwert erreicht hat und dann wieder ausschaltet. Das nachfolgende Beispiel enthält ein aus 6 Bausteinen aufgebautes Unterprogramm, das die beschriebene Aufgabe erfüllt.

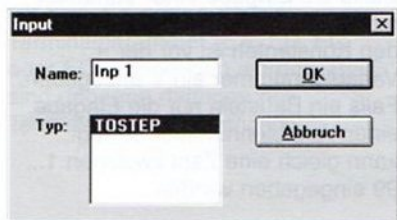
Mit dem Befehl **Bearbeiten Unterprogramm** legst Du zunächst ein neues Unterprogramm an. Im Auswahlfenster gibst Du den Namen POSV6 ein. Danach kannst Du mit **Baustein einfügen** folgenden Ablauf eingeben:

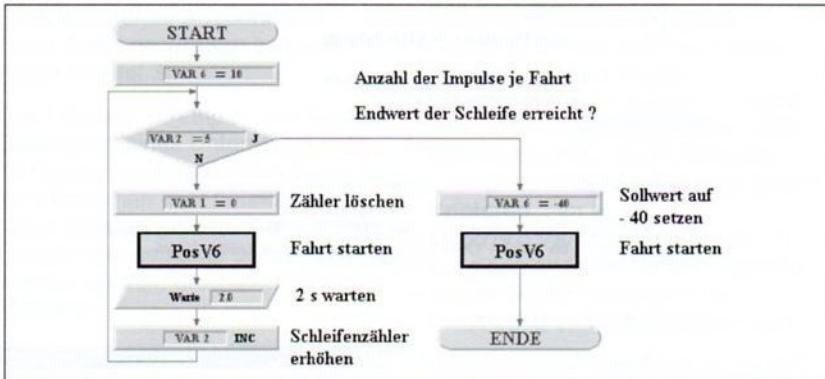




In der Variablen VAR 6 wird der Endwert EW des Zählers (Baustein POSITION) vom aufrufenden Ablauf an das Unterprogramm übergeben. Dieser Endwert wird auch als Sollwert bezeichnet, weil der Zählerstand VAR 1 (Istwert ZW) nach Beendigung der Aktion den Wert VAR6 erreichen soll. Mit dem Baustein VERGLEICH wird ermittelt, ob der Motor in der linken oder rechten Drehrichtung eingeschaltet werden muß, um den Endwert EW zu erreichen. Je nach gewählter Drehrichtung wird dann der Baustein POSITION als Vorwärts- bzw. Rückwärtszähler (INC bzw. DEC) aufgerufen. Nach dem Erreichen des Endwertes EW wird dann der Motor abgeschaltet. Die Verbindung mit dem Ablauf im Hauptprogramm erfolgt über die Bausteine Input und Output. Füge die Bausteine Input und Output aus der Bibliothek Standard ein.

Der Name wird für die Unterscheidung der einzelnen Anschlüsse benötigt. Anschließend muß Du mit dem Befehl **Aussehen** aus dem Menü **Unterprogramme** (siehe Kapitel 3) die Anschlüsse auf dem Rahmen des Unterprogrammaufrufes plazieren. Danach kannst Du das Unterprogramm in das Hauptprogramm einfügen und mit anderen Bausteinen verbinden. In dem Beispiel muß vor dem Aufruf von POSV6 auf die Variable VAR 6 ein Wert zugewiesen werden. Außerdem darf im Projekt die Variable VAR 1 nicht mehr für andere Aufgaben verwendet werden. Wenn Du VAR 1 überschreibst, hat das Unterprogramm die alte Position des Motors „vergessen“. Dies kann z.B. angewendet werden, wenn eine Achse in einer Schleife mit 5 Durchläufen alle 2 s um 10 Impulse weitergefahren werden soll (siehe Abbildung).





Nach dem Start des Ablaufes werden alle Variablen gelöscht. Deshalb braucht die Variable VAR 2 nicht auf 0 gesetzt werden. Nach 5 Durchläufen wird der Sollwert VAR 6 auf einen negativen Wert gesetzt. Beachte, daß der Istwert ZW in VAR 1 zu diesem Zeitpunkt nicht 0, sondern 10 ist. Wenn der Ablauf beendet ist, steht der Motor wieder in Ausgangsposition.

Wenn Du zuerst das Unterprogramm UP1 auf den Namen des neuen Unterprogrammes umkopierst, und dann die anderen Bausteine einfügst, kannst Du das Unterprogramm sofort im Hauptprogramm aufrufen. Andernfalls mußt Du im nächsten Schritt die Anschlüsse der Bausteine Input und Output auf dem Rahmen des Unterprogrammaufrufes mit dem Befehl **Aussehen** im Menü **Unterprogramm** plazieren.

6. Eingaben in Bausteinen

Variablen V1... V99

In einem LLWin-Projekt können maximal 99 verschiedene Variablen verwendet werden. Eine Variable ist eine ganze Zahl (Integer). Ihr Wert kann im Start-Modus von Run im Bereich von -999 bis 9999 in den Bausteinen angezeigt werden. Intern können von den Bausteinen die Zahlen von -32567 bis + 32567 verarbeitet werden. Ist der Wert größer als 9999 oder kleiner als -999 so wird **** angezeigt.

Variablen können als Zähler verwendet werden. Sie können aber auch als Zwischenspeicher von Analogwerten dienen. Wenn im Eingabefenster eines Bausteines in der Überschrift die drei Buchstaben VAR angegeben sind, dann ist an dieser Stelle die Eingabe einer Variablen möglich. Zur Unterscheidung von den Konstanten ist vor der Variablennummer ein V anzugeben. Falls ein Baustein nur die Eingabe einer Variablennummer abfragt, kann gleich eine Zahl zwischen 1... 99 eingegeben werden.



Über Variablen können Informationen zwischen verschiedenen Abläufen ausgetauscht werden. Falls der Wert einer Variablen in mehreren Bausteinen geändert wird, dann wird der zuletzt zugewiesene Wert in allen Bausteinen angezeigt.

Analogwerte EX und EY

In einem LLWin-Projekt können die Analogwerte EX und EY verwendet werden. Die Analogwerte werden über das Interface eingelesen. Der am Interface an EX bzw. EY angeschlossenen Widerstand im Bereich von 0 ... 5 k Ω wird in einen Zahlenwert zwischen ca. 50 und 500 umgewandelt. Diese Zahlenwerte können bei einigen PCs auch etwas abweichen. Ein offener Analog-Eingang wird als 999 abgebildet.

Sind zwei Interfaces am Computer angeschlossen, dann können nur die Analogwerte am ersten Interface eingelesen werden.

Analogwerte können nicht von allen Bausteinen ausgewertet werden. In solchen Fällen müssen Analogwerte auf eine Variable umgespeichert werden. Soll z.B. ein Analogwert als Endwert im Baustein Position eingesetzt werden, so muß in einem parallel arbeitenden Ablauf der Analogwert mit dem Baustein VARIABLE umgespeichert werden.

Wenn im Eingabefenster eines Bausteines in der Überschrift E# angegeben ist, kann der Analogwert EX bzw. EY oder auch ein Terminalparameter (EA, EB, EC oder ED) angegeben werden. Bei E* sind dagegen dagegen nur Terminalparameter möglich.

Konstanten (Zahlen von -999 bis 9999)

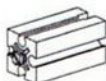
Bei Zuweisungen und Vergleichen können auch Konstanten eingegeben werden. Konstanten sind Zahlenwerte zwischen -999 bis 9999. Wenn das Programm abgearbeitet wird, kann eine Konstante nicht geändert werden. Soll der Zahlenwert auch während des laufenden Programmes geändert werden können, dann muß ein Terminalparameter verwendet werden.

Wenn im Eingabefenster eines Bausteines in der Überschrift das Wort Zahl angegeben ist, dann ist an dieser Stelle die Eingabe einer Konstanten möglich.

Terminalparameter EA, EB, EC und ED

Sollen während der Abarbeitung des Programmes am PC Werte eingegeben werden können, dann muß der Baustein TERMINAL eingefügt werden. Im TERMINAL sind die vier Terminalparameter EA, EB, EC und ED vorhanden. Durch einen linken Mausklick auf die rechts neben den Terminalparametern angeordneten Dreiecken können die Terminalparameter jeweils um 1 erhöht bzw. um 1 reduziert werden. Durch Mausklick auf die Terminalparameter kann ein Zahlenwert von -999 bis 9999 eingegeben werden.

Wenn im Eingabefenster eines Bausteines in der Überschrift E* oder E# angegeben ist, dann ist an dieser Stelle die Eingabe eines Terminalparameters EA, EB, EC bzw. ED möglich.



Binäreingänge 1 ... 26

Taster und Lichtschranken können nur zwei Zustände haben. Sie sind entweder eingeschaltet (1) oder ausgeschaltet (0). Deshalb bezeichnet man diese Signale auch als Binärwerte. Ein fischertechnik-Interface hat 8 Binäreingänge. Wenn zwei Interfacebaugruppen gekoppelt werden, können maximal 16 Binäreingänge E1 bis E16 in LLWin eingelesen werden. Das TERMINAL hat 10 Softwareschalter E17 bis E26, die ebenfalls vom Programm wie Binärwerte behandelt werden. Die Softwareschalter bleiben im Unterschied zu den Tasten in ihrem Zustand. Deshalb sind sie besonders für dauerhafte Umschaltungen im Ablaufprogramm geeignet.

Die Binäreingänge 1 ... 26 werden in den Programmverzweigungen mit dem Baustein EINGANG, in den Zählbausteinen POSITION und im Baustein FLANKE verwendet.

Motorausgänge 1 ... 8

Zur Ansteuerung von Motoren, Lampen und Elektromagneten wird der Baustein MOTOR verwendet. Bei Lampen und Elektromagneten kann eine der beiden Richtungen gewählt werden. Wenn ein Motor sich nicht in der im Ablauf vorgesehenen Richtung dreht, dann ist es am einfachsten, die Kabel am Motor umzustecken. Wenn ein Interface angeschlossen ist, können die Motorausgänge 1 ... 4 und wenn zwei Interfaces angeschlossen sind, können die Motorausgänge 1 ... 8 angesteuert werden.

Auf dem TERMINAL werden die gerade eingeschalteten Motorausgänge rot bzw. grün angezeigt.

7. Die in LLWin verwendbaren Bausteine

Anweisung

DECVARIABLE - Herunterzählen einer Variablen um 1

Symbol:



Beschreibung:

Der Baustein DECVARIABLE wird zum Herunterzählen einer Variablen um 1 verwendet.

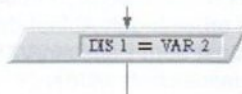
Beim Einfügen des Bausteins wird die Nummer der Variablen abgefragt. Es kann eine Nummer 1...99 eingegeben werden.

Im Init-Modus von Run kann die Variablennummer durch einen Klick mit der linken Maustaste in das Nummernfeld geändert werden.

Ein-/Ausgabe

DISPLAY - Anzeigen einer Variablen im Terminal

Symbol:



Beschreibung:

Der Baustein DISPLAY dient zur Anzeige einer Variablen in einer der beiden roten Siebensegmentanzeigen des TERMINAL -Bausteines.

Beim Einfügen des Bausteines wird zuerst die Nummer der Siebensegmentanzeige abgefragt.



Es kann die Nummer 1 für die linke Anzeige oder 2 für die rechte Anzeige eingetragen werden.

Es können Variablen, Konstanten, Analogwerte oder Terminalparameter angezeigt werden. Soll eine Variable angezeigt werden, so muß der Buchstabe V, gefolgt von der Variablennummer eingetragen werden. Soll ein Analogwert bzw. Terminalparameter angezeigt werden, so muß der Buchstabe E, gefolgt von einem der Buchstaben X oder Y für Analogwerte bzw. A, B, C oder D für Terminalparameter eingetragen werden. Soll eine Konstante angezeigt werden, so muß der Zahlenwert - z.B. 123 ohne einen vorangestellten Buchstaben eingetragen werden.

Im Init-Modus von Run können die Displaynummer und der anzuzeigende Wert durch einen Klick mit der linken Maustaste in den entsprechenden Feldbereich geändert werden.

wird entweder mit dem Baustein am unteren oder mit dem Baustein am rechten Ausgang fortgesetzt.

Beim Einfügen des Bausteines wird zuerst die Nummer des Einganges abgefragt. Es kann eine Nummer von 1 ... 26 eingegeben werden. Davon sind die Nummern 1 bis 8 für das erste Interface und von 9 bis 16 für das zweite Interface (falls vorhanden) vorgesehen. Mit den Nummern 17 bis 26 können die 10 Schalter des Bausteines TERMINAL abgefragt werden.

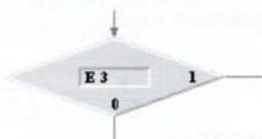
Dann wird abgefragt, bei welchem Eingangszustand nach rechts verzweigt werden soll. Es ist entweder eine 0 oder eine 1 einzugeben.

Im Init-Modus von Run kann sowohl die Eingangsnummer als auch der Eingangszustand, bei dem nach rechts verzweigt werden soll, durch einen Klick mit der linken Maustaste in den entsprechenden Feldbereich geändert werden.



EINGANG - Verzweigung bei einem Binäreingang

Symbol:



Beschreibung:

Zur Abfrage von Binäreingängen (z.B. Tastern, Lichtschranken oder Terminalschalter) wird der Baustein EINGANG verwendet. Mit diesem Baustein wird eine Verzweigung des Ablaufes programmiert. Der Ablauf

Anfang und Ende

ENDE - Ende eines Ablaufes

Symbol:



Beschreibung:

Soll ein Ablauf unter bestimmten Bedingungen oder auch immer nach Abarbeitung von ein oder mehreren Bausteinen beendet werden, wird der Ausgang des letzten abzuarbeitenden Bausteines mit dem Baustein ENDE verbunden. Es ist möglich, in einem Ablauf mehrere ENDE-Bausteine zu verwenden, es ist aber auch möglich, mehrere Ausgänge mit einem ENDE Baustein zu verbinden.



Ein-/Ausgabe

FLANKE - Flanke von 1 nach 0 am Eingang abwarten

Symbol:



Beschreibung:

Der Baustein FLANKE wartet auf eine 1-0-Flanke an einem Binäreingang. Ist der Eingang bereits 0, wenn der Baustein Flanke aufgerufen wird, dann wird erst dann der nächste Baustein abgearbeitet, wenn der Eingang zunächst auf 1 und dann auf 0 gewechselt hat.

Beim Einfügen des Bausteines FLANKE wird die Nummer des Einganges abgefragt. Es kann eine Nummer von 1 ... 26 eingegeben werden. Davon sind die Nummern 1 bis 8 für das erste Interface und von 9 bis 16 für das zweite Interface (falls vorhanden) vorgesehen. Mit den Nummern 17 bis 26 können die 10 Tasten des Bausteines TERMINAL abgefragt werden.

Im Init-Modus von Run kann die Eingangsnummer durch einen Klick mit der linken Maustaste in den entsprechenden Feldbereich geändert werden.

Beschreibung:

Der Baustein INCVARIABLE wird zum Hochzählen einer Variablen um 1 verwendet.

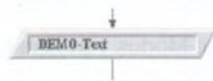
Beim Einfügen des Bausteins wird die Nummer der Variablen abgefragt. Es kann eine Nummer von 1 ... 99 eingegeben werden.

Im Init-Modus von Run kann die Variablennummer durch einen Klick mit der linken Maustaste in das Nummernfeld geändert werden.

Ein-/Ausgabe

MELDUNG - Meldung auf dem Terminal anzeigen

Symbol:



Beschreibung:

Mit dem Baustein MELDUNG kann eine maximal 17 Zeichen lange Meldung auf dem TERMINAL angezeigt werden. Nach dem Einfügen des Bausteines wird der anzuzeigende Text eingegeben.

Im Init-Modus von Run kann der Anzeigetext durch einen Klick mit der linken Maustaste in das Textfeld geändert werden.

Anweisung

INCVARIABLE - Hochzählen einer Variablen um 1

Symbol:



Ein-/Ausgabe

MOTOR - Motorausgang schalten

Symbol:



Beschreibung:

Der Baustein MOTOR wird benötigt, um einen Motor, einen Elektromagnet oder eine Lampe anzusteuern.

Beim Einfügen des Bausteines wird zuerst die Nummer des Ausganges abgefragt. Es kann eine Nummer von 1 ... 8 eingegeben werden. Mit einer Nummer von 1 bis 4 wird ein Ausgang auf dem ersten Interface und den Nummern von 5 bis 8 wird ein Ausgang auf dem zweiten Interface (falls vorhanden) angesteuert.

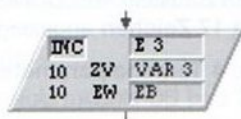
Dann wird abgefragt, in welcher Richtung der Motor angesteuert werden soll, bzw. ob der Motor ausgeschaltet werden soll. Mit L bzw. R wird der Motor mit der Drehrichtung **LINKS** bzw. **RECHTS** eingeschaltet. Mit A wird der Motor abgeschaltet (**AUS**).

Im Init-Modus von Run kann die Motornummer und Richtung durch einen Klick mit der linken Maustaste in den entsprechenden Feldbereich geändert werden.

Ein-/Ausgabe

POSITION - Warten bis eine Position erreicht wurde

Symbol:



Beschreibung:

Der Baustein POSITION zählt die Pegeländerungen an einem Eingang und vergleicht die Anzahl mit einem Endwert. Wenn der Endwert über-

schritten wird, wird der nächste Baustein abgearbeitet.

Beim Einfügen des Bausteines POSITION wird zuerst die Nummer des Einganges abgefragt. Es kann eine Nummer von 1 ...26 eingegeben werden. Davon sind die Nummern 1 bis 8 für das erste Interface und von 9 bis 16 für das zweite Interface (falls vorhanden). Mit den Nummern 17 bis 26 können die 10 Schalter des Bausteines TERMINAL abgefragt werden.

Dann wird abgefragt, ob vorwärts oder rückwärts gezählt werden soll. Hier sind die Eingaben I für **INC** (vorwärts zählen) bzw. D für **DEC** (rückwärts zählen) möglich.

Im nächsten Eingabefenster wird die Nummer der Zählvariablen ZV abgefragt. Hier können die Variablennummern 1 ... 99 eingetragen werden.

Dann erfolgt die Abfrage des Endwertes EW. Der Endwert kann eine Konstante, eine Variable oder ein Terminalparameter sein.

- 999 ... 9999 Konstante
- V1 ... V99 Variable 1 ... 99
- EA... ED Terminalparameter

 Hinweis:

Wenn ein Analogwert EX bzw. EY als Endwert verwendet werden soll, so muß dieser zuvor in eine Variable umgespeichert werden. Damit ständig der aktuelle Analogwert in der Variablen eingetragen wird, muß das Umkopieren in einer Endlosschleife mit dem Baustein VARIABLE in einem anderen Ablauf ausgeführt werden.

Im Init-Modus von Run kann die Eingangsnummer, die Zählrichtung,



die Zählvariable für die aktuelle Position und der Endwert durch einen Klick mit der linken Maustaste in den entsprechenden Feldbereichen geändert werden.

Anfang und Ende

START - Start eines Ablaufes

Symbol:

START

Beschreibung:

Jeder Ablauf sollte mit genau einem START-Baustein beginnen. Wenn in einem Ablauf kein START-Baustein enthalten ist, dann kann der Ablauf nicht abgearbeitet werden.

Enthält ein Ablauf mehrere START-Bausteine, dann werden je Zyklus auch mehrere Bausteine in diesem Ablauf abgearbeitet. Das Verhalten eines Ablaufes mit mehreren START-Bausteinen ist nur sehr schwer zu verstehen.

Es können mehrere Abläufe in einem Projekt enthalten sein. Eigenständige Abläufe haben untereinander keine Verbindungen. Es können aber auf die gleichen Variablen, Eingänge und Ausgänge zugegriffen werden. Die Aufspaltung in mehrere Abläufe ist dann erforderlich, wenn, während in einem Ablauf auf ein Ereignis gewartet wird, in einem anderen Ablauf eine Aktion ausgeführt werden soll.

TERMINAL - Bedienerterminal



Symbol:



Beschreibung:

Der Baustein TERMINAL ermöglicht die Bedienung und Anzeige während der Abarbeitung des Programmes.

Welche der **Motorausgänge** 1 ... 8 eingeschaltet sind, wird im TERMINAL ständig angezeigt:

-  Motor **LINKS**
-  Motor **RECHTS**
-  Motor **AUS**

Durch Verwendung des Bausteines DISPLAY können Variablen, Konstanten oder Analogwerte auf den beiden roten **Siebensegmentdisplays** angezeigt werden.

Unter dem linken Siebensegmentdisplay kann mit dem Baustein MELDUNG ein **Text mit 17 Zeichen** angezeigt werden.

Die **4 Terminalparameter EA, EB, EC und ED** sind im Unterschied zu den Konstanten während des laufenden Programmes änderbar.

Außerdem hat das Terminal **10 Schalter E17 ... E26**, die wie Binäreingänge verwendet werden können. Auch während der Abarbeitung des Programmes kön-



nen diese Schalter über Mausklick umgeschaltet werden.

Mit dem Schalter **RESET** werden alle Motorausgänge auf AUS geschaltet. Das Programm wird wie über den Befehl Start im Menü Run neu gestartet, wenn der Schalter RESET rückgesetzt wird.

Mit dem Schalter **NOTAUS** werden alle Motorausgänge auf **AUS** geschaltet. Das Programm läuft weiter. Die zuletzt eingeschalteten Motorausgänge werden wieder zugeschaltet, wenn der Schalter NOTAUS rückgesetzt wird.

Die Bedienung der Terminals ist nur im Modus Run möglich. Terminalparameter und Schalter können auch nach Aufruf des Befehles Init gesetzt werden. Die Einstellungen werden nach dem Programmstart übernommen.

Programme, in denen nur Schalter und Terminalparameter ausgewertet werden, kann man auch ohne Interface starten und testen.

Anweisung

VARIABLE - Zuweisung auf eine Variable

Symbol:



Beschreibung:

Mit dem Baustein VARIABLE wird eine Variable auf einen Wert gesetzt. Dieser Wert kann eine Konstante, ein Analogwert, ein Terminalparameter oder eine andere Variable sein.

Beim Einfügen des Bausteines erfolgen nacheinander zwei Abfragen.

Zuerst wird die Nummer der zu überschreibenden Variablen abgefragt. Es können die Nummern 1 ... 99 gewählt werden.

Dann wird gefragt, mit welchem Wert die Variable überschrieben werden soll. Es sind folgende Eingaben möglich:

-999 ... 9999	Konstante
V1 ... V99	Variable 1 ... 99
EX, EY	Analogwert
EA ... ED	Terminalparameter

Im Init-Modus von Run kann die Variablennummer der zu überschreibenden Variable und die Zuweisung durch einen Klick mit der linken Maustaste in den entsprechenden Feldbereich geändert werden.



Vergleich - Verzweigung nach einem Vergleich

Symbol:



Beschreibung:

Um eine Programmverzweigung in Abhängigkeit einer Variablen zu realisieren, wird der Baustein VERGLEICH verwendet. Entsprechend des Vergleichergebnisses wird ein Ablauf entweder mit dem Baustein am unteren oder mit dem Baustein am rechten Ausgang fortgesetzt.

Beim Einfügen des Bausteines wird zuerst die Nummer der Variablen auf der linken Seite des Vergleiches abgefragt. Es sind die Nummern von 1 ... 99 möglich.



Dann wird der Vergleichsoperator abgefragt. = / > / < sind als Operator erlaubt. Wird kein Operator eingegeben, so wird automatisch der Operator = eingefügt.

Zuletzt wird das Argument auf der rechten Seite des Vergleiches abgefragt. Es kann eine Konstante, ein Terminalparameter oder eine andere Variable als Argument eingetragen werden.

-999 ... 9999	Konstante
V1 ... V99	Variable 1 ... 99
EA... ED	Terminalparameter

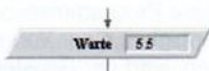
Wenn ein Analogwert als Vergleichsoperator eingesetzt werden soll, so muß dieser mit dem Baustein VARIABLE in eine Variable umkopiert werden.

Im Init-Modus von Run kann die Variablennummer auf der linken Seite, der Operator in der Mitte und das Argument auf der rechten Seite durch einen Klick mit der linken Maustaste in den entsprechenden Feldbereich geändert werden.

Ein-/Ausgabe

WARTE - wartet eine definierte oder zufällige Zeitspanne

Symbol:



Beschreibung:

Der Baustein WARTE dient zur Programmierung einer Wartezeit im Ablauf. Die Wartezeit wird erst gestartet, wenn der WARTE-Baustein aufgerufen wird. Der Ablauf verweilt solange im WARTE-Baustein, bis die eingetragene Zeit abgelaufen ist.

Beim Einfügen des Bausteins wird die Wartezeit in Sekunden abgefragt. Soll eine definierte Zeitspanne gewartet werden, so kann ein Wert zwischen 0.1 ... 999.9 eingetragen werden.

Der Baustein kann auch verwendet werden, um eine zufällige Zeitspanne zu warten. Hierzu muß eine negative Zeit eingetragen werden (- 0.1 ... -999.9) Der Baustein bildet einen Zufallswert, der zwischen 0.0 und dem Betrag der eingetragenen Zahl liegt. Die Zufallszahl bildet dann die Wartezeit des Bausteins in Sekunden. Bei jedem neuen Programmzyklus wird ein neuer Zufallswert ermittelt.

Im Init-Modus von Run kann die Wartezeit durch einen Klick mit der linken Maustaste auf den Zahlenwert geändert werden.

INPUT - Eingang eines Unterprogrammes

Symbol:



Beschreibung:

Dieser Baustein stellt die Verbindung zwischen dem Beginn eines Unterprogramm-Ablaufs und der übergeordneten Programmebene her. Ihm kann ein Name zugewiesen werden. Dieser wird zur korrekten Zuordnung des Eingangs in den beiden Programmebenen benötigt.

Nach dem Einfügen von Input-Bausteinen sind diese auf dem Rahmen des Unterprogramm-Aufrufs als kleine Kreise zu platzieren. Dazu muß im Menü Unterprogramm der Befehl



Aussehen aufgerufen werden.

! Achtung:

Das Entfernen eines Input-Bausteins aus einem Unterprogramm, das bereits in einer höheren Programmebene eingefügt wurde, ist nicht möglich. Es wird eine Warnung angezeigt.

OUTPUT - Ausgang eines Unterprogrammes

Symbol:



Beschreibung:

Dieser Baustein stellt die Verbindung zwischen dem Ende eines Unterprogramm-Ablaufs und der übergeordneten Programmebene her. Ihm kann ein Name zugewiesen werden. Dieser wird zur korrekten Zuordnung des Ausgangs in den beiden Programmebenen benötigt.

Nach dem Einfügen von Output-Bausteinen sind diese auf dem Rahmen des Unterprogrammaufrufs als kleine Kreise zu plazieren. Dazu muß im Menü Unterprogramm der Befehl Aussehen aufgerufen werden.

! Achtung:

Das Entfernen eines Output-Bausteins aus einem Unterprogramm, das bereits in einer höheren Programmebene eingefügt wurde, ist nicht möglich. Es wird eine Warnung angezeigt.

Text - Kommentartext im Bild

Symbol: [Text]

Beschreibung:

Der eingegebene Text dient zur Programmdokumentation. Dabei können drei Schriftgrößen gewählt werden. Der Kommentar kann wie andere Bausteine verschoben und entfernt werden.

Soll ein bereits eingegebener Text modifiziert werden, so kann dies beim Bearbeiten durch Positionieren des Mauszeigers auf dem Text, Drücken der Taste <Shift> und Mausklick mit der rechten Maustaste erfolgen.



Anhang

Übersicht der wichtigsten Befehle und Kommandos

Baustein einfügen

- Menü **Bearbeiten** Befehl **Baustein ...**,
oder **<Einf>**
- Bibliothek wählen
- Bausteinname eingeben oder mit der Maus anwählen
- Bestätigung mit **ENTER** oder durch Mausklick auf **OK**

Baustein kopieren

- Greifen des Bausteins mit **linker Maustaste** bei gedrückter Taste **<Shift>**
- bei gedrückter Maustaste auf gewünschte Position kopieren und Maustaste loslassen

Baustein ändern (Werte und Adressen)

- Menü **Bearbeiten**
- **<Shift>** drücken und **rechter Mausklick** auf den Baustein / Text
- oder Menü **Run, Init-Modus** und auf Eingabefelder klicken

Hinweis: Der Text kann nur im Bearbeiten-Modus geändert werden.

Baustein verschieben

- Baustein mit **linker Maustaste** greifen
- bei gedrückter Maustaste auf gewünschte Position verschieben und Maustaste loslassen

Gruppen verschieben

- Im Menü **Optionen** die **Cursorumschaltung** aktivieren
- Der Mauszeiger zeigt an, ob auf der aktuellen Position eine Spalte bzw. Zeile eingefügt oder gelöscht werden kann
- Mit **<Shift> Mausklick links** Spalte einfügen, mit **<Shift> Mausklick rechts** Zeile einfügen.
- Mit **<Strg> Mausklick links** Spalte löschen, mit **<Strg> Mausklick rechts** Zeile löschen, jeweils bis zur Cursorposition.

Baustein löschen

- Menü **Bearbeiten** Befehl **Baustein löschen** oder **<Entf>**
- Hammer auf Baustein positionieren, **linker Mausklick**

Verbinden

- Beginn am Ausgang bzw. Eingang eines Bausteins mit **rechtem Mausklick**
- Beenden am Eingang bzw. Ausgang eines Bausteins mit **rechtem Mausklick**



Verbindung löschen

- Mausclick mit **linker Maustaste** bei gedrückter Taste **<Strg>** auf die zu löschende Verbindung

Verbindungsabschnitt löschen

- Mausclick mit **linker Maustaste** bei gedrückter Taste **<Strg>** auf den Ausgang (Stift muß angezeigt werden)

Zoomfaktor

- Vergrößern des Zoomfaktors durch Betätigung der Taste „+“ im Ziffernblock
- Verkleinern des Zoomfaktors durch Betätigung der Taste „-“ im Ziffernblock
- Normierung des Zooms auf 1.0 durch Betätigung der Taste #
- *oder* numerische Eingabe des Zoomfaktors im Menü **Optionen**

Kommentartext einfügen

- Menü **Bearbeiten** Befehl **Baustein einfügen ...**, *oder* **<Einf>**
- Bibliothek **Standard** anwählen
- Baustein **Text** anwählen
- Eingabe des Textes, Bestätigung durch Mausclick auf **OK**

Anschlüsse in Unterprogramme einfügen

- Menü **Bearbeiten** Befehl **Baustein einfügen...**, *oder* **<Einf>**
- Bibliothek **Standard** wählen
- Bausteinname **Input** für Eingang bzw. **Output** für Ausgang anwählen
- Namen des Anschlusses eingeben (max. 6 Zeichen)
- Anschlüsse im Menü **Unterprogramm**, Befehl **Aussehen** plazieren

Hinweis: Der Name eines Input- oder Outputbausteines kann modifiziert werden, wenn Du mit **<Shift> rechte Maustaste** auf den Baustein klickst.

Aufruf der Online-Hilfe

- Die Online-Hilfe kann über **<F1>** nach Anwahl der Menüzeile mit dem Mauszeiger oder mit den Cursortasten zu einem speziellen Kommando aufgerufen werden
- Eintritt in die Online-Hilfe über das Inhaltsverzeichnis, über das Suchverzeichnis, aus der Bausteinauswahlbox
- Onlinehilfe zu einem speziellen Baustein auf dem Bildschirm durch Positionierung des Mauszeigers auf einen Baustein und **<Strg + F1>**

Hinweis: Im Hilfemenü kannst Du bei ? auf „immer im Vordergrund“ schalten.

Unterprogramm anzeigen (Run-Modus)

- Klick mit der **linken Maustaste** in den Unterprogrammaufruf
- Zurück mit Klick mit der **rechten Maustaste**



© 1996 fischerwerke Artur Fischer GmbH & Co. KG
D-72178 Tumlingen/Waldachtal

Alle Rechte, auch das des auszugsweisen Nachdruckes,
der auszugsweisen und vollständigen photomechanischen
Wiedergabe (Photokopie, Microkopie) vorbehalten.

PRINTED in Germany
2. Auflage 04/97

fischerwerke Artur Fischer GmbH & Co. KG
Weinhalde 14-18, D-72178 Waldachtal
Telefon (074 43) 12-43 69
Telefax (074 43) 12-45 91

