



akYtec ALP

Programmiersoftware für programmierbare Geräte akYtec

Benutzerhandbuch

Inhaltsverzeichnis

1	Allgemeines	4
1.1	Abkürzungen und Begriffe	4
1.2	Über Software	4
1.3	Systemanforderungen	4
2	Benutzeroberfläche	6
2.1	Menü	6
2.2	Symbolleisten	9
2.3	Arbeitsbereich	10
2.4	Library Box	11
2.5	Property Box	12
2.6	Display-Manager	13
2.7	Statusleiste	14
2.8	Variable Box	15
2.9	Komponenten-Manager	15
2.9.1	Online-Datenbank	16
2.9.2	Lokale Bibliothek	17
3	Arbeitsgrundlagen	18
3.1	Programmausführung	18
3.2	Programmzykluszeit	18
3.3	Projekterstellung	19
3.4	Verbindung zum Gerät	20
3.4.1	OFFLINE-Modus	20
3.5	Geräteinformationen	21
3.6	Projektinformationen	21
3.7	Projekt auf das Gerät hochladen	22
3.8	Firmware-Update / -Reparatur	22
3.8.1	Erzwungenes Firmware-Update / -Reparatur	23
4	Gerätekonfiguration	25
4.1	Display	25
4.2	Uhr	25
4.3	Schnittstellen	26
4.3.1	Arbeit mit Modbus	27
4.3.2	Schnittstelle hinzufügen / entfernen	28
4.3.3	RS485-Schnittstellenkonfiguration	29
4.4	Erweiterungsmodule	34
4.5	Eingänge	35
4.6	Ausgänge	36
4.7	Kalibrierung	36

4.7.1	Eingangs-Kalibrierung.....	37
4.7.2	Ausgangs-Kalibrierung.....	37
4.8	Zielgerät ändern	38
5	Variable.....	39
5.1	Eigenschaften.....	39
5.2	Datentyp	40
5.3	Standardvariablen	41
5.4	Servicevariablen.....	41
5.5	Netzwerkvariablen.....	41
5.6	Variablenbaustein kopieren / einfügen.....	42
6	Bibliothek	44
6.1	Funktionen.....	44
6.1.1	Logische Operatoren	44
6.1.2	Mathematische Operatoren	46
6.1.3	Vergleichsoperatoren.....	49
6.1.4	Bitshift-Operatoren.....	51
6.1.5	Bit-Operatoren	52
6.2	Funktionsbausteine	54
6.2.1	Trigger.....	54
6.2.2	Timer.....	56
6.2.3	Generatoren	59
6.2.4	Zähler	60
6.2.5	Analog.....	62
6.3	Projektmakros	64
6.3.1	Makro exportieren, importieren, herunterladen.....	65
6.3.2	FB in Makro.....	66
6.3.3	Neues Makro über Hauptmenü.....	66
6.3.4	Neues Makro über Kontextmenü	67
6.3.5	Makro aktualisieren	68
6.3.6	Makro ersetzen	69
6.3.7	Reihenfolge der E/A-Punkte ändern	69
6.4	Displayelemente.....	69
6.4.1	Textfeld	70
6.4.2	I/O-Box (INT/REAL)	70
6.4.3	I/O-Box (BOOL)	72
6.4.4	Dynamische Box	73
6.4.5	ComboBox	74
7	Schaltprogrammentwicklung	75
7.1	Verwendung von Bibliotheks-Bausteinen	75
7.2	Verwendung von Textfeld	76

7.3	Verwendung von Variablen	77
7.4	Verwendung von Konstanten	78
7.5	Verwendung von Verzögerungslinien	78
7.6	Netzwerk-Datenaustausch	79
7.7	Lesen / Schreiben in FB	80
7.8	Konvertierungsblöcke.....	81
7.9	Elemente anordnen	81
7.10	Ausführungsablauf.....	82
7.11	Simulation	82
7.11.1	Arbeitsweise.....	82
7.11.2	Beobachtungsfenster	84
7.12	Online-Debugging.....	84
8	Display-Programmierung	86
8.1	Display-Editor	87
8.2	Grafische Struktur	87
8.3	Displayform-Eigenschaften / Sprünge.....	87
8.4	Displayform kopieren / einfügen.....	89
9	Tastaturkürzel	90
10	Programmbeispiele	91
10.1	Aufgabe 1: Lichtschalter mit automatischer Abschaltung.....	91
10.2	Aufgabe 2: Steuerung eines Mischers	93

1 Allgemeines

1.1 Abkürzungen und Begriffe

In diesem Handbuch verwendete Abkürzungen und Begriffe:

Tabelle 1.1

Abkürzungen und Begriffe	Erklärung
ALP	Programmiersoftware
Arbeitsbereich	Ein Bereich in der Software-Benutzeroberfläche zum Erstellen des Benutzerprogramms, in dem Programmbausteine und Verbindungslinien zwischen ihnen platziert werden.
FBD	Funktionsblockdiagramm ist eine grafische Programmiersprache
Funktion	Eine Struktureinheit eines Programms mit einem Rückgabewert. Die Funktion speichert keine Informationen über ihren inneren Zustand, d. h. wenn die Funktion mit den gleichen Eingangswerten aufgerufen wird, wird der gleiche Ausgangswert zurückgegeben.
Funktionsbaustein	Eine Struktureinheit eines Programms mit internem Speicher und einem oder mehreren Ausgangswerten. Im Programm wird es als Instanz verwendet, d. h. als eine Kopie mit eigenem Speicher.
Makro	Ein vom Benutzer erstellter Funktionsbaustein
Programmzyklus	Die Ausführungszeit des Schaltprogramms, die von dessen Komplexität abhängt
Projekt	Eine mit ALP-Software für ein programmierbares Gerät erstellte Benutzeranwendung
Zielgerät	Ein Gerät, für das das Projekt erstellt wird

1.2 Über Software

ALP ist eine Programmiersoftware für programmierbare Geräte der Firma akYtec GmbH. Die Programmiersprache ist die grafische Sprache FBS (Funktionsplan). Die Software ermöglicht das Testen des erstellten Programms und das Hochladen in den nichtflüchtigen Speicher eines programmierbaren Geräts.

Jedes Projekt enthält ein oder mehrere Schaltprogramme und eine Gerätekonfiguration.

Der erste Arbeitsbereich enthält das Hauptschaltprogramm. Makros (Benutzer-Funktionsbausteine) können als Schaltprogramme in separaten Arbeitsbereichen erstellt werden.

Wenn das Zielgerät über ein Display verfügt, kann es mithilfe von Displayformen programmiert werden. Jedes dieser Displayformen wird in einem separaten Arbeitsbereich geöffnet.

Es kann jeweils nur ein Projekt geöffnet werden.

1.3 Systemanforderungen

Die ALP Software läuft unter Windows XP/Vista/7/8/10. Vorinstallierte Software .NET Framework 4.0 ist erforderlich. Wenn nicht installiert, werden Sie aufgefordert, es zu installieren.

Minimale Hardwareanforderungen:

- 1,5 GHz Prozessor

Allgemeines

- 1 GB RAM
- 100 MB freier Festplattenspeicher
- freie USB-Schnittstelle
- Tastatur und Maus
- 1024x768 Bildschirmauflösung

Empfohlene Hardwareanforderungen:

- 3,2 GHz Prozessor
- 4 GB RAM
- 200 MB freier Festplattenspeicher
- freie USB-Schnittstelle
- Tastatur und Maus
- 1280x800 Bildschirmauflösung

Internetverbindung ist erforderlich für:

- Software-Aktualisierung
- Geräte-Firmware- Aktualisierung
- Herunterladen der Slave-Gerätevorlagen
- Herunterladen von Makros im Komponentenmanager

2 Benutzeroberfläche

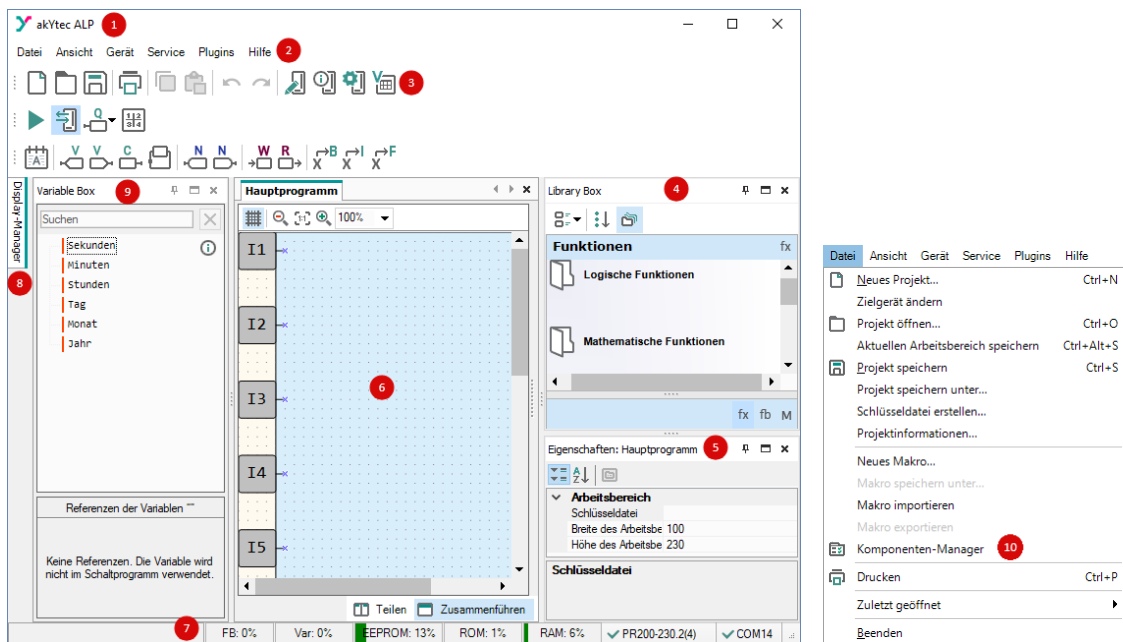


Abb. 2.1

1. **Titelleiste** – zeigt den Namen der Software und den Pfad zur geöffneten Projektdatei an
2. **Menüleiste** – besteht aus folgenden Gruppen: Datei, Ansicht, Gerät, Service, Plugins und Hilfe
3. **Symboleisten** – Standard, Service und Einfügen: Schneller Zugriff auf die wesentlichen Funktionen von ALP
4. **Library Box** – ein Panel, in dem alle Funktionen, Funktionsblöcke und Makros angezeigt werden, die dem Projekt hinzugefügt werden können
5. **Property Box** – ein Panel, in dem die Eigenschaften des ausgewählten Projektelements angezeigt und geändert werden können
6. **Arbeitsbereich** – ein Feld in der Benutzeroberfläche, in dem ein Schaltprogramm, eine Display-Struktur oder eine Displayform angezeigt und geändert werden kann
7. **Statusleiste** – zeigt Status- und Fehlermeldungen, Speicherauslastung des Zielgeräts, Status des angeschlossenen Geräts und der Programmierschnittstelle an
8. **Display-Manager** – ein Tool zum Programmieren der angezeigten Informationen (nur für Geräte mit Display verfügbar)
9. **Variable Box** – ein Panel, in dem alle Projektvariablen mit ihren Parametern und Verknüpfungen angezeigt werden. Verwenden Sie Drag-and-Drop, um einen Variablenbaustein in einem Schaltprogramm zu platzieren.
10. **Komponenten-Manager** – Spezialwerkzeug in einem separaten Fenster, um auf die Online-Datenbank zuzugreifen und die Elemente aus der Online-Datenbank einer Offline- oder Projektbibliothek hinzuzufügen. Internetverbindung wird benötigt.

2.1 Menü

Tabelle 2.1 Menü **Datei**

Neues Projekt	Ein neues Projekt öffnen. Das aktuelle Projekt wird geschlossen. (Abschn. 3.3)
Zielgerät ändern	Das Zielgerät im Projekt ändern (Abschn. 4.8)

Benutzeroberfläche

Projekt öffnen	Ein zuvor gespeichertes Projekt öffnen
Aktiven Arbeitsbereich speichern	Den aktuell geöffneten Arbeitsbereich speichern
Projekt speichern	Das aktuelle Projekt speichern
Projekt speichern unter...	Eine Kopie des Projekts in einem anderen Ordner oder unter einem anderen Namen erstellen
Schlüsseldatei erstellen...	Eine Datei mit Schlüssel erstellen, um das Projekt vor unbefugtem Zugriff zu schützen (in Entwicklung)
Projektinformationen...	Projektinformationen anzeigen und ändern (Abschn. 3.6)
Neues Makro...	Ein neues Makro in einem separaten Arbeitsbereich öffnen (Abschn. 6.3.1)
Makro speichern unter...	Das aktuelle Makro unter einem neuen Namen im Projektbibliothek speichern
Makro importieren	Ein Makro aus einer Datei in die Projektbibliothek importieren (Abschn. 6.3.3)
Makro exportieren	Das aktuelle Makro als Datei speichern (Abschn. 6.3.2)
Komponenten-Manager	Die Komponentenmanager-Oberfläche öffnen (Abschn. 2.9)
Drucken	Den Dialog zu öffnen, um die Druckoptionen festzulegen und den aktiven Arbeitsbereich zu drucken
Zuletzt geöffnete Projekte	Die zuletzt geöffneten Projekte auflisten
Beenden	ALP schließen

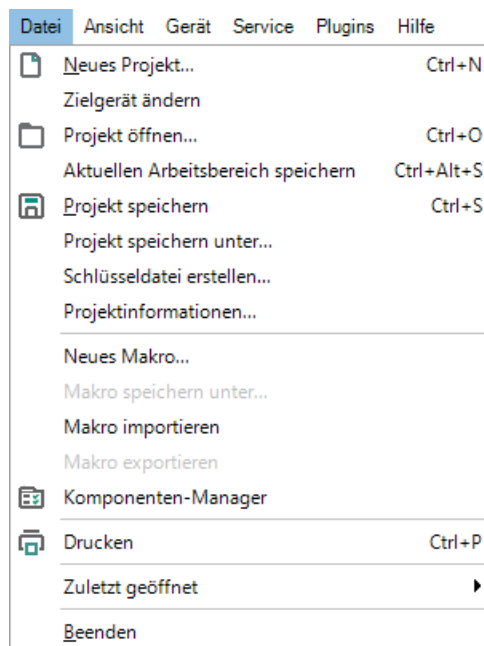


Abb. 2.2 Menü **Datei**

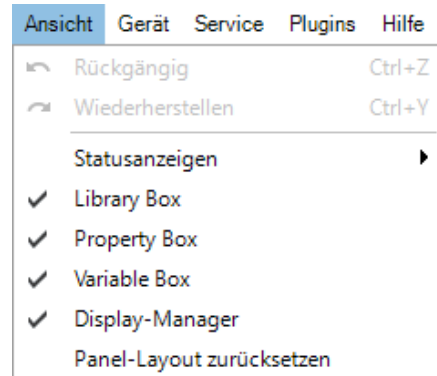


Abb. 2.3 Menü **Ansicht**

Tabelle 2.2 Menü **Ansicht**

Rückgängig	Die letzte Aktion rückgängig machen
Wiederherstellen	Die letzte rückgängig gemachte Aktion wiederherstellen
Statusindikatoren	Die Indikatoren zu der Statusleiste hinzufügen oder sie zu entfernen (Abschn. 2.7)
Library Box	Das Panel Library Box ein- / ausblenden (Abschn. 2.4)
Property Box	Das Panel Property Box ein- / ausblenden (Abschn. 2.5)

Benutzeroberfläche

Variable Box	Das Panel Variable Box ein- / ausblenden (Abschn. 2.8)
Display-Manager *	Das Panel Display-Manager ein- / ausblenden (Abschn. 2.6)

* Nur für Geräte mit Display verfügbar

Tabelle 2.3 Menü **Gerät**

Programm auf Gerät übertragen	Das aktuelle Projekt in den Gerätespeicher laden (Abschn. 3.7)
Firmware-Update...	Die Firmware des angeschlossenen Geräts aktualisieren (Abschn. 3.8)
Informationen...	Informationen über die Software, das Zielgerät und das angeschlossene Gerät (Abschn. 3.5)
Variablentabelle...	Die editierbare Tabelle der Projektvariablen mit ihren Parametern (Abschn. 5)
Kalibrierung...	Kalibrierung starten (Abschn. 4.7)
Konfiguration...	Gerätekonfiguration (Abschn. 4)
Port-Einstellungen...	Einstellungen der Programmierschnittstelle (Abschn. 3.4)

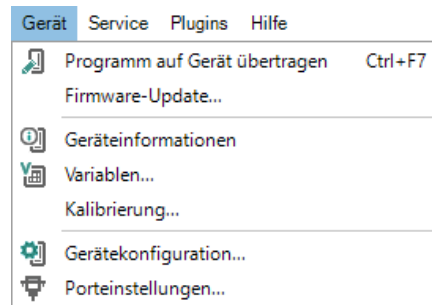


Abb. 2.4 Menü **Gerät**

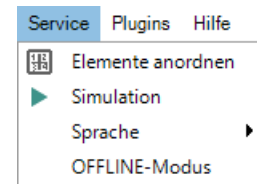


Abb. 2.5 Menü **Service**

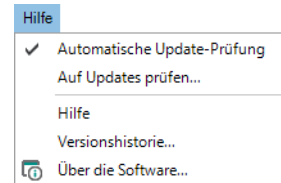


Abb. 2.6 Menü **Hilfe**

Tabelle 2.4 Menü **Service**

Elemente anordnen	Funktionsbausteine desselben Typs werden im Arbeitsbereich von oben nach unten und von rechts nach links automatisch neu nummeriert (Abschn. 7.9)
Simulation	Simulation starten / beenden (Abschn. 7.11)
Sprache	Die Sprache der Benutzeroberfläche auswählen
OFFLINE-Modus	OFFLINE-Modus aktivieren (Abschn. 3.4.1)

Tabelle 2.5 Menü **Hilfe**

Automatische Update-Prüfung	Wenn aktiviert, wird die Aktualisierungsprüfung beim Programmstart durchgeführt
Auf Updates prüfen	Prüfen, ob Software-Updates verfügbar sind
Hilfe	Online-Hilfe
Versionshistorie	Die Liste der Änderungen für alle Softwareversionen
Über Software	Informationen zur aktuellen Softwareversion

Benutzeroberfläche

2.2 Symbolleisten

Tabelle 2.6

Standard		
	Neues Projekt	Ein neues Projekt öffnen. Das aktuelle Projekt wird geschlossen.
	Projekt öffnen	Ein zuvor gespeichertes Projekt öffnen
	Projekt speichern	Das aktuelle Projekt speichern
	Drucken	Den Dialog zur Einstellung der Druckoptionen für den aktuellen Arbeitsbereich öffnen
	Kopieren	Das im Arbeitsbereich ausgewählte Element kopieren
	Einfügen	Das kopierte Element einfügen
	Rückgängig	Die letzte Aktion rückgängig machen
	Wiederherstellen	Die letzte rückgängig gemachte Aktion wiederherstellen
	Programm auf Gerät übertragen	Das aktuelle Projekt in den Gerätespeicher laden
	Geräteinformationen	Ein Fenster mit Informationen über die Software, das Zielgerät und das angeschlossene Gerät öffnen (Abschn. 3.5)
	Konfiguration...	Gerätekonfiguration (Abschn. 4)
	Variablen-tabelle	Die Tabelle der Projektvariablen zur Bearbeitung öffnen (Abschn. 5)
Service		
	Simulation	Simulation starten / beenden (Abschn. 7.11)
	Online-Debugging	Online-Debugging starten / beenden (Abschn. 7.12)
	Ausführungsablauf	Die Ausführungsreihenfolge für die Ausgänge oder Verzögerungslinien in Schaltprogramm oder in Makro ändern (Abschn. 7.10)
	Elemente anordnen	Funktionsbausteine desselben Typs werden im Arbeitsbereich von oben nach unten und von rechts nach links automatisch neu nummeriert (Abschn. 7.9)
Einfügen		
	Kommentarfeld	Textfeld für Programm-Kommentare (Abschn. 7.2)

Benutzeroberfläche

	Variablen-Ausgangsbaustein	Variable, deren Wert im Programm festgelegt werden kann (Abschn. 7.3)
	Variablen-Eingangsbaustein	Variable, deren Wert im Programm gelesen werden kann (Abschn. 7.3)
	Konstanten-Baustein	Konstante (Abschn. 7.4)
	Verzögerungslinie	Rückmeldung mit einer Verzögerung von einem Zyklus (Abschn. 7.5)
	Netzwerkvariablen-Ausgangsbaustein	Variable, deren Wert über ein Netzwerk geschrieben werden kann (Abschn. 7.6)
	Netzwerkvariablen-Eingangsbaustein	Variable, deren Wert über ein Netzwerk gelesen werden kann (Abschn. 7.6)
	Baustein WriteToFB	Der Eingangswert des Bausteins ist mit einem der Parameter des gewählten Funktionsbausteins verbunden und wird zu dessen Änderung verwendet (Abschn. 7.7)
	Baustein ReadFromFB	Der Ausgangswert des Bausteins ist mit einem der Parameter des ausgewählten Funktionsbausteins verbunden und wird verwendet, um dessen Wert zu lesen (Abschn. 7.7)
	Konvertierung in BOOL	Konvertierung eines beliebigen Wertes in einen BOOL-Wert (Abschn. 7.8)
	Konvertierung in INT	Konvertierung eines beliebigen Wertes in einen INT-Wert (Abschn. 7.8)
	Konvertierung in REAL	Konvertierung eines beliebigen Wertes in einen REAL-Wert (Abschn. 7.8)

2.3 Arbeitsbereich

Beim Öffnen eines Projekts wird der Arbeitsbereich mit der Registerkarte **Hauptprogramm** im mittleren Teil des Fensters angezeigt (Abb. 2.7).

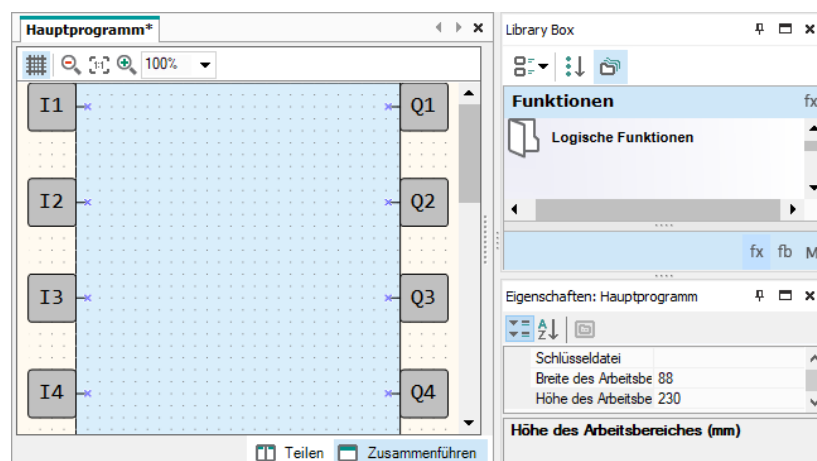


Abb. 2.7 Arbeitsbereich

Ein Schaltprogramm wird erstellt und geändert, indem Programmbausteine und Verbindungslinien zwischen ihnen im Arbeitsbereich platziert werden. Die Größe des Arbeitsbereiches kann in der Property Box geändert werden. Die Eingänge (links) und Ausgänge (rechts) sind wie folgt gekennzeichnet:

- Ix – digitale Eingänge
- Alx – analoge Eingänge
- Qx – Relaisausgänge

Benutzeroberfläche

AOx – Analoge Ausgänge

Fx – LED-Indikatoren

Die Nummern (x) entsprechen den Sequenznummern der physikalischen E/A-Punkten des Zielgeräts. E/A-Punkte können im Arbeitsbereich per Drag-and-Drop nach oben und unten verschoben werden.

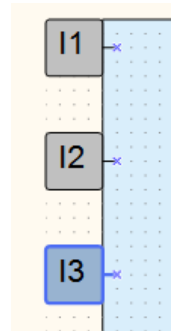


Abb. 2.8

Tabelle 2.7 Symbolleiste des Arbeitsbereiches

	Raster anzeigen/verbergen	Vertikale und horizontale Lineale und ein Raster im Arbeitsbereich anzeigen / ausblenden. Wenn der Raster ist sichtbar, werden die Bausteine und Verbindungslinien am Raster ausgerichtet.
	Zoom -	Arbeitsbereich um 10 % verkleinern
	Originalgröße	Zur ursprünglichen Größe (100%) zurückkehren
	Zoom +	Arbeitsbereich um 10 % vergrößern
	Skalierung auswählen	Zoomfaktoren von 20% bis 400 %

Die Symbole **Teilen** und **Zusammenführen** befinden sich unterhalb des Arbeitsbereichs. Verwenden Sie das Symbol **Teilen**, um dasselbe Schaltprogramm in zwei Arbeitsbereichen anzuzeigen. Wenn ein Programm zu groß ist, könnte es nützlich sein, zwei verschiedene Teile des Programms gleichzeitig zu betrachten. Verwenden Sie das Symbol **Zusammenführen**, um zu einem Arbeitsbereich zurückzukehren.

2.4 Library Box

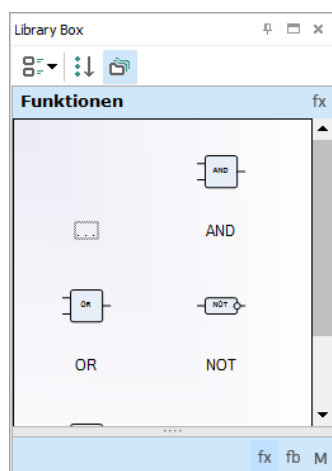


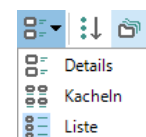
Abb. 2.9 Gekachelte Ansicht

Das Panel Library Box ist eine Baustein -Bibliothek. Das Panel enthält drei Bibliotheken: **Funktionen**, **Funktionsbausteine** und **Projektmakros**.

Wählen Sie im unteren Teil des Panels ein Symbol aus, um den jeweiligen Inhalt anzuzeigen.

Die Standardposition des Panels im Fenster ist die obere rechte Ecke (kann geändert werden).

Die Ansichtsoptionen können über die Symbole in der oberen Zeile des Panels geändert werden.



Klicken Sie auf das Symbol **Alle anzeigen**, um alle Bausteine der ausgewählten Bibliothek anzuzeigen (Abb. 2.10):

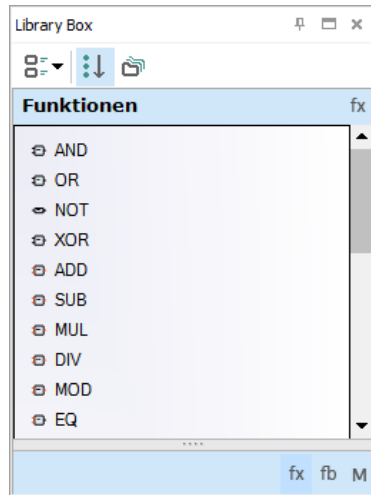


Abb. 2.10 Alle anzeigen

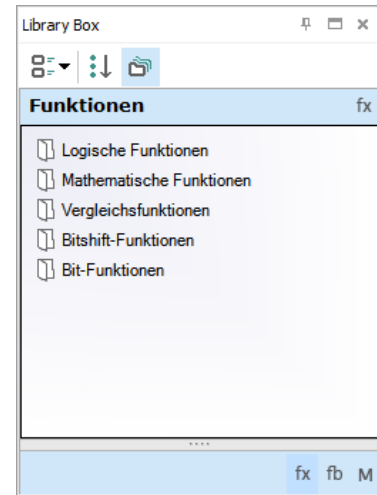


Abb. 2.11 Gruppirt anzeigen

Klicken Sie auf das Symbol **Gruppirt anzeigen**, um die Bausteine der ausgewählten Bibliothek gruppiert anzuzeigen (Abb. 1.11). Doppelklicken Sie auf einen Ordner, um ihn zu öffnen.

Für die Beschreibungen der Bibliotheksgruppen und einzelnen Bausteine siehe Abschn. 6.

2.5 Property Box

Das Panel Property Box dient zum Anzeigen und Ändern der Parameter von Programmelementen. Wählen Sie das gewünschte Element aus, um seine Eigenschaften anzuzeigen.

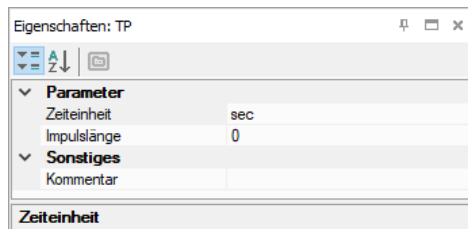


Abb. 2.12

Die Standardposition des Panels im Fenster ist die untere rechte Ecke (kann geändert werden).

Die Parameter werden standardmäßig nach Kategorien gruppiert angezeigt.

Um sie in alphabetischer Reihenfolge anzuzeigen, klicken Sie auf das Symbol . Um sie gruppiert anzuzeigen, klicken Sie auf das Symbol .

Wählen Sie das Eingabefeld eines Parameters aus, um seinen Wert zu bearbeiten.

2.6 Display-Manager

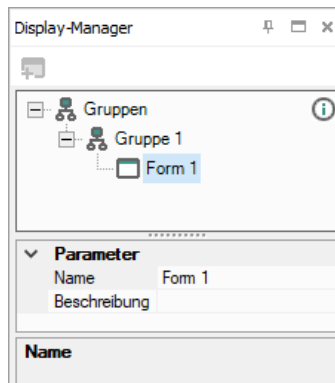


Abb. 2.13

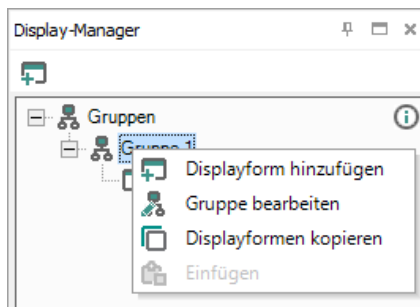


Abb. 2.14

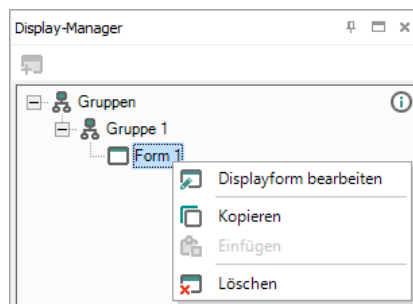


Abb. 2.15

Wenn das Zielgerät über ein Display verfügt, können die angezeigten Informationen mithilfe eines oder mehrerer Display-Formen programmiert werden. Weitere Details zur Display-Programmierung finden Sie im Abschn. 8.

Die Programmierung erfolgt mit dem Programmierwerkzeug **Display Manager**. Die Registerkarte Display-Manager befindet sich in der oberen linken Ecke des Fensters. Klicken Sie auf die Registerkarte, um das Panel zu öffnen. Das Panel enthält eine Symbolleiste, die hierarchische Struktur der Displayformen (Strukturbaum) und die Liste der Formeigenschaften.

Die Parameter der ausgewählten Displayform werden im unteren Teil des Panels angezeigt.

Um die ausgewählte Form zu programmieren, öffnen Sie sie in einem separaten Arbeitsbereich **Display-Editor** (Abb. 2.16) über das Kontextmenü oder doppelklicken Sie auf die Form in der Gruppe (Abb. 2.15).

Der Arbeitsbereich enthält eine Displayform mit den Symbolen rechts, die es ermöglichen, die Anzahl der angezeigten Zeilen zu ändern. Die zuerst angezeigten Zeilen sind fett umrandet.

Zur Verwendung des Display-Editors siehe Abschn. 8.1.

Wenn mehrere Anzeigeformen erstellt werden, sollten Sie „Sprünge“ zwischen ihnen definieren, damit der Operator zwischen den Formen wechseln kann, um die gewünschten Informationen zu sehen. Dies kann in einem separaten Arbeitsbereich **Struktur-Editor** (Abb. 2.17) erfolgen, der die grafische Struktur von Displayformen mit „Sprüngen“ und deren Bedingungen darstellt.

Um den Struktur-Editor zu öffnen, verwenden Sie den Punkt **Gruppe bearbeiten** im Gruppenkontextmenü (Abb. 2.14).

Zur Verwendung des Struktur-Editors siehe Abschn. 8.2.

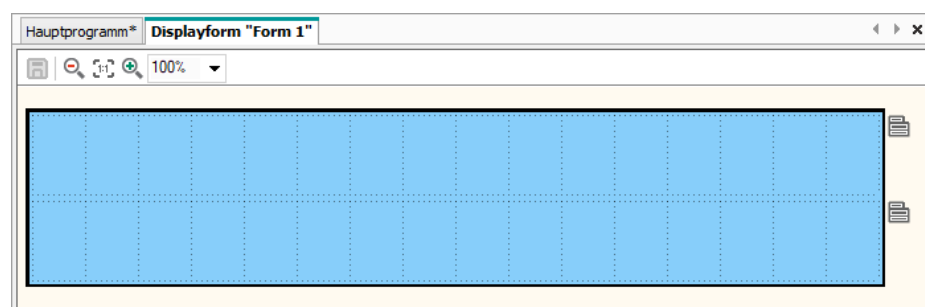


Abb. 2.16 Display-Editor

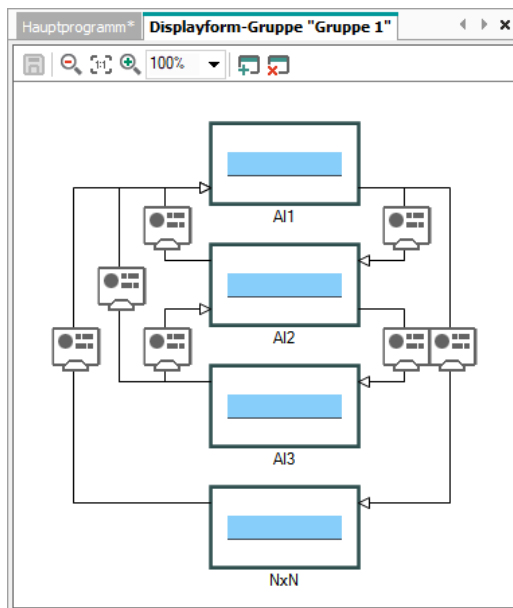








Abb. 2.17 Struktur-Editor

Tabelle 2.8 Editor-Symboleiste

Allgemein	
	Arbeitsbereich speichern
Zoom	
	Verkleinern um 10%
	Originalgröße
	Vergrößern um 10%
100% ▾	Maßstab wählen
Displayform	
	Displayform hinzufügen
	Displayform löschen

2.7 Statusleiste

Status- und Fehlermeldungen werden im linken Feld der Statusleiste angezeigt.

Außerdem enthält die Statusleiste verschiedene Statusindikatoren, die Informationen über Speicherbelegung des Zielgeräts (Ressourcen-Indikatoren), über dem Status des angeschlossenen Geräts und der Programmierschnittstelle anzeigen.

Welche Indikatoren in der Statusleiste verfügbar sind, hängt vom Typ des Zielgeräts ab.

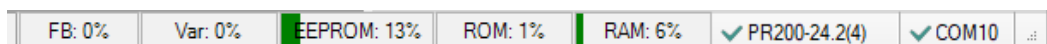


Abb. 2.18

Die Ressourcen-Indikatoren zeigen die Nutzung der Ressourcen in Prozent der gesamten verfügbaren Menge an. Um die absoluten Betrag der Ressource zu sehen, bewegen Sie die Maus über dem Indikator.

Wenn ein Gerät angeschlossen ist, enthält die Statusleiste folgende Informationen:

- **FB** – die Anzahl der verfügbaren und verwendeten Funktionsbausteine
- **Var** – die Anzahl der verfügbaren und verwendeten Variablen
Hinweis: Einige Variablen können von der Software automatisch erstellt werden, wenn Elemente wie Verzögerungslinien oder Mehrfachverknüpfungen mit gemeinsamen Knoten in einem Projekt verwendet werden.
- **Stack** – die Anzahl der verfügbaren und verwendeten Stack-Ebenen
- **EEPROM** – der verfügbare und verwendete Retain-Speicher
- **ROM** – der verfügbare und verwendete ROM-Speicher
- **RAM** – der verfügbare und verwendete RAM-Speicher
Hinweis: Die ALP-Software berechnet automatisch die verfügbaren Ressourcen eines Geräts und zeigt eine Warnung an, wenn der kritische Wert erreicht ist.
- **Gerät** – der Typ des angeschlossenen Geräts
Hinweis: Klicken Sie auf den Indikator, um zum OFFLINE-Modus zu wechseln. In diesem Modus ist die Verbindung zum Gerät unterbrochen. Mit dem nächsten Klick deaktivieren Sie den OFFLINE-Modus (Abschn. 3.4.1).

Benutzeroberfläche

- **Port** – der ausgewählte Portnummer (Programmierschnittstelle)

Hinweis: Klicken Sie auf den Indikator, um das Fenster **Porteinstellungen** zu öffnen

2.8 Variable Box

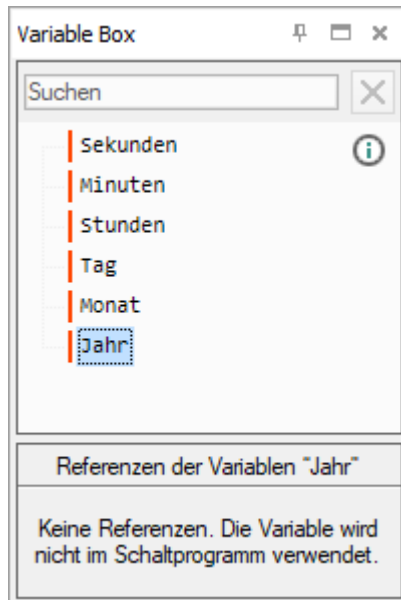


Abb. 2.19

Das Panel **Variable Box** zeigt die Projektvariablen aus der Variablen-tabelle.

Die Standardposition des Panels in der oberen linken Fensterecke kann geändert werden.

Sie können die Informationen über die Variable in einem QuickInfo-Text ansehen.

Die Variablen-Verknüpfungen werden als Links im unteren Bereich des Panels angezeigt. Wenn Sie auf den Link klicken, wird der Baustein, auf den sich die Variable bezieht, im Arbeitsbereich markiert.

Ziehen Sie eine Variable per Drag-and-Drop, um sie als Eingangsbaustein im Schaltprogramm zu platzieren.

Um eine Variable als Ausgangsbaustein zu platzieren, halten Sie die Umschalt-taste gedrückt, während Sie die Variable ziehen und ablegen.

Wenn eine Variable per Drag-and-Drop auf einen Anschlusspin eines Bausteins gezogen wird, wird der erstellte Variablenbaustein mit diesem Anschlusspin verbunden.

Wenn die Variable an mehreren Stellen im Projekt verwendet wird, können alle Verknüpfungen mit dem Punkt **Referenzen anzeigen** im Kontextmenü des Variablenbausteins angezeigt werden. Klicken Sie auf den Link, um die Verknüpfung anzuzeigen (Abb. 2.20)

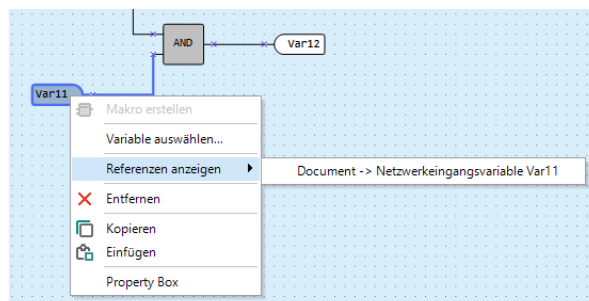


Abb. 2.20

2.9 Komponenten-Manager

Neue Makros und Gerätevorlagen können von der akYtec Online-Datenbank heruntergeladen werden. Der Komponenten-Manager ist das Werkzeug für alle Interaktionen mit

Benutzeroberfläche

dieser Datenbank. Wählen Sie den Menüpunkt **Datei > Komponenten-Manager** aus, um ihn in einem separaten Fenster zu öffnen.

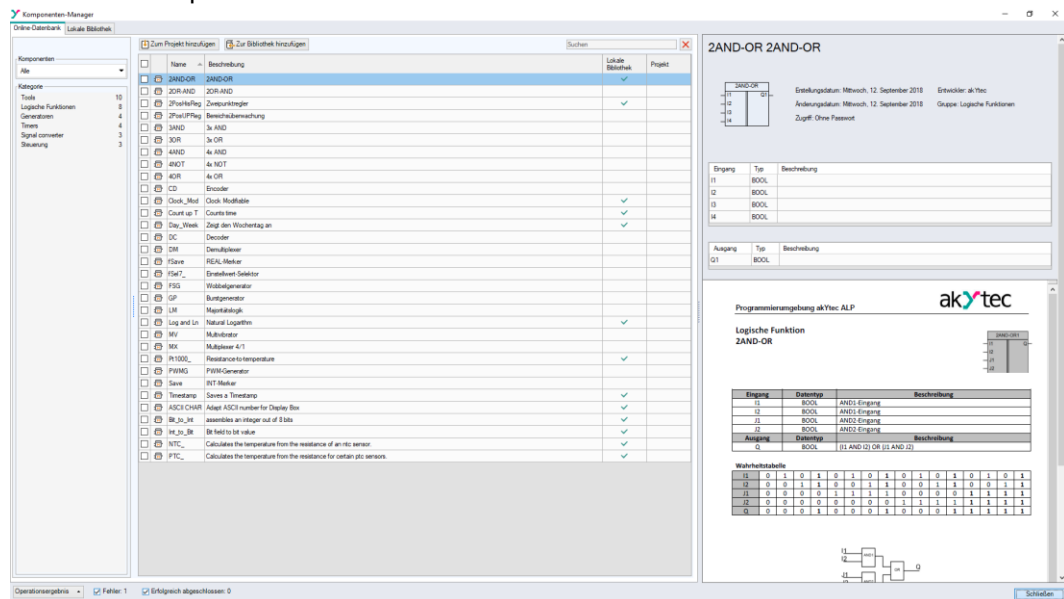


Abb. 2.21

Der Internetzugang ist erforderlich, damit der Komponenten-Manager mit der Online-Datenbank interagieren kann.

Die Benutzeroberfläche verfügt über zwei Registerkarten: **Online-Datenbank** (Abschn. 2.9.1) und **Lokale Bibliothek** (Abschn. 2.9.2). Die Bausteine werden in Kategorien angezeigt und können nach Namen gefiltert werden.

2.9.1 Online-Datenbank

- Schaltfläche **Zu Projekt hinzufügen** – die ausgewählten Bausteine (Makros oder Gerätevorlagen) aus der Online-Datenbank werden zur Projektbibliothek hinzugefügt. Die Bausteine werden in der Projektdatei gespeichert und können in der Library Box (Abschn. 2.4) im Bereich **Projektmakros** eingesehen werden.
- Schaltfläche **Zur Bibliothek hinzufügen** – die ausgewählten Bausteine aus der Online-Datenbank werden in die lokale Bibliothek heruntergeladen und können dann offline verwendet werden.

Ein Häkchen in der Spalte **Projekt** oder **Bibliothek** zeigt an, dass den Baustein erfolgreich heruntergeladen / hinzugefügt wurde.

Die Bibliotheksdateien sind unter der lokalen Adresse gespeichert:

**C: \Benutzer \[Benutzername] \Dokumente \akYtec ALP \Library **

Im oberen rechten Feld wird die kurze Beschreibung des ausgewählten Bausteins und im rechten unteren Feld die vollständige Beschreibung angezeigt. Die vollständige Beschreibung ist ein PDF-Dokument. Scrollen Sie das Dokument bis zum Ende, um die PDF-Symbolleiste anzuzeigen. Damit können Sie das Dokument herunterladen oder ausdrucken.

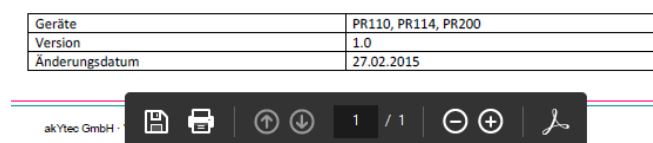


Abb. 2.22

Klicken Sie auf die Schaltfläche **Operationsergebnis** unten im Fenster, um die Programmierungen zu den ausgeführten Vorgängen anzuzeigen.

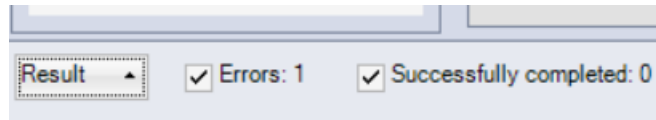


Abb. 2.23

2.9.2 Lokale Bibliothek

- Schaltfläche **Zu Projekt hinzufügen** – die ausgewählten Bausteine (Makros oder Gerätevorlagen) aus der Online-Datenbank werden zur Projektbibliothek hinzugefügt. Die Bausteine werden in der Projektdatei gespeichert und können in der Library Box (Abschn. 2.4) im Bereich **Projektmakros** (Abschn. 6.3) eingesehen werden.
- Schaltfläche **Importieren** – die ausgewählten Bausteine werden aus einer Datei zur Projektbibliothek hinzugefügt.
- Schaltfläche **Exportieren** – die ausgewählten Bausteine aus der Projektbibliothek werden unter dem angegebenen Pfad als Dateien gespeichert.
- Schaltfläche **Entfernen** – die ausgewählten Bausteine werden aus der lokale Bibliothek entfernt.

3 Arbeitsgrundlagen

Um ein Gerät zu programmieren, gehen Sie wie folgt vor:

- ALP starten
- ein neues Projekt für das ausgewählte Zielgerät erstellen oder ein vorhandenes Projekt öffnen (Abschn. 3.3)
- das Projekt auf dem PC speichern
- das Programm im Simulationsmodus überprüfen und Fehler beseitigen (Abschn. 7.11)
- das Projekt auf das angeschlossene Gerät übertragen (Abschn. 3.7)

3.1 Programmausführung

Das ausgewählte Zielgerät bestimmt die Anzahl der verfügbaren Ein- und Ausgänge und die Verfügbarkeit einer Echtzeituhr. Die allgemeine Struktur eines programmierbaren Relais wird in Abb. 3.1 dargestellt.

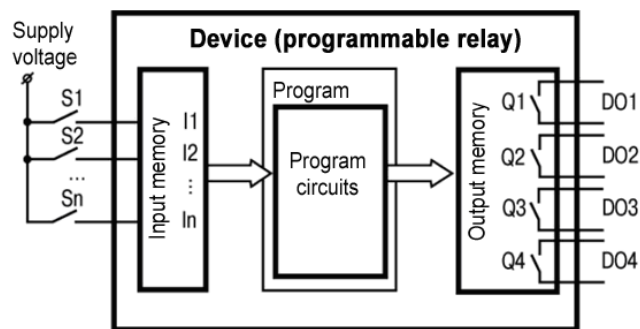


Abb. 3.1 PR-Arbeitsdiagramm

Ein programmierbares Relais ist eine Art der SPS mit einem zyklisch ausgeführten Programm:

Schritt 1 – Der Status der physikalischen Eingänge wird in die Eingangsspeicherzellen (Input Image Table) gespeichert.

Schritt 2 – Die Eingangsspeicherzellen werden ausgelesen und das Programm wird von der ersten bis zur letzten Anweisung ausgeführt.

Schritt 3 – Die Ergebnisse werden in den Ausgangsspeicherzellen (Output Image Table) gespeichert und an die Ausgänge angelegt.

Wenn der letzte Schritt abgeschlossen ist, wird das Programm ab dem ersten Schritt erneut ausgeführt.

3.2 Programmzykluszeit

Die Zykluszeit wird vom Gerät berechnet und ist von der Programmkomplexität abhängig. Die folgenden Bausteine sind besonders ressourcenintensiv.

- FBs (Abschn. 6.2)
- Makros (Abschn. 6.3)
- Netzwerkvariable (Abschn. 5.5)
- Display-Elemente (Abschn. 6.4).

Es ist ein schreibgeschützter Parameter und kann auf dem Gerätedisplay (falls vorhanden) über das Systemmenü angezeigt werden (siehe Bedienungsanleitung des Geräts). Die minimale Zykluszeit beträgt 1 ms.

Arbeitsgrundlagen

Hinweis: Die Parameter **Zykluszeit** im Gerät und im ALP-Simulationsmodus (Abschn. 7.11) sind trotz des gleichen Namens unterschiedlich.

3.3 Projekterstellung

Um ein neues Projekt zu erstellen, wählen Sie im Hauptmenü **Datei > Neues Projekt** aus oder verwenden Sie das entsprechende Symbol in der Taskleiste. Wählen Sie das Zielgerät im Dialogfenster **Geräteauswahl** und bestätigen mit **OK** (Abb. 3.3).

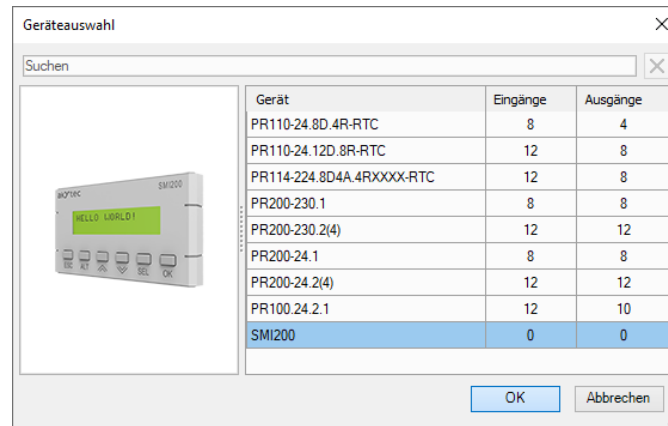


Abb. 3.2

Aussehen des neuen Projekts:

- Arbeitsbereich – ein leeres Schaltprogramm
- Statusleiste – die Informationen zu verfügbaren Ressourcen
- Library Box – die verfügbare Programmbausteine
- Property Box – die Eigenschaften des Arbeitsbereiches


Schaltprogramm

Jetzt kann das Hauptprogramm in FBD-Sprache mit den allgemeinen Programmbausteinen aus der Werkzengleiste **Einfügen** und den spezifischen Programmbausteinen aus der Library Box im Arbeitsbereich erstellt werden. Zeichnen Sie Verbindungslinien zwischen Eingängen, Ausgängen und Bausteinen, um logische Verbindungen im Programm herzustellen. Einzelheiten zu den einzelnen Programmbausteinen und Verbindungslinien finden Sie im Abschn. 7.

Display-Manager

Wenn das ausgewählte Gerät ein Display besitzt, die Registerkarte **Display-Manager** erscheint links vom Arbeitsbereich. Mit diesem Tool können Sie die angezeigten Informationen programmieren.

Simulation

Das Programm kann offline simuliert werden. Starten Sie den Simulationsmodus über den Menüpunkt **Service > Simulationsmodus** oder das Symbol  in der Symbolleiste, ändern Sie den Zustand der Eingänge und notieren Sie den Zustand der Ausgänge, um die Korrektheit des Programms zu überprüfen (Abschn. 7.11).

Online-Debugging

Wenn das Gerät angeschlossen ist und das Programm im Gerät und in ALP identisch ist, können Sie das Online-Debugging verwenden, um die Richtigkeit des Programms im Gerät zu überprüfen (Abschn. 7.12).

3.4 Verbindung zum Gerät



Das Gerät muss vor dem Anschließen an den PC ausgeschaltet werden.

Geräte können direkt an einen PC (PR200, SMI200) oder über einen PR-KP20-Adapter (PR110, PR114) angeschlossen werden. Das erforderliche Verbindungskabel ist im Lieferumfang des PR200 oder des Adapters enthalten.

Das Gerät kann direkt an den PC (für PR200) oder über den Adapter PR-KP20 (für PR110, PR114) angeschlossen werden. Die benötigten Anschlusskabel sind im Lieferumfang des Adapters oder des PR200 enthalten.

Verbinden Sie das Gerät mit einem USB-Port des PCs, schalten Sie das Gerät ein und wählen Sie den seriellen Anschluss im Menü **Gerät > Porteinstellungen** aus. Die Nummer des emulierten COM-Anschlusses können Sie im Windows-Gerätemanager unter "Anschlüsse (COM und LPT)" finden.

Wenn das Betriebssystem den richtigen Treiber nicht findet, installieren Sie den Treiber für PR200 oder für den Adapter PR-KP20. Dieser kann unter akytec.de heruntergeladen werden.

Wählen Sie im Dialogfenster **Porteinstellungen** die Portnummer und bestätigen Sie mit **OK**. Alle anderen Einstellungen sind nicht änderbar und werden nur zu Ihrer Information angezeigt.

Wenn die Verbindung hergestellt ist, werden Informationen über das angeschlossene Gerät und den seriellen Port in den Statusindikatoren angezeigt.

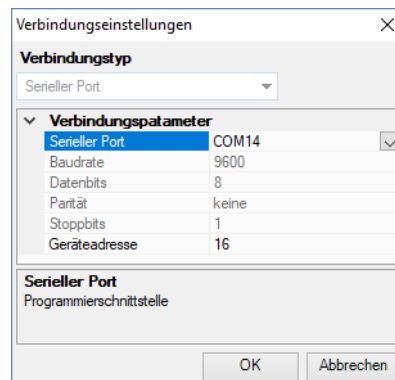


Abb. 3.3

3.4.1 OFFLINE-Modus

In diesem Modus wird die Verbindung zwischen ALP und dem Gerät unterbrochen. Der Modus ist hilfreich, wenn Sie mit zwei ALP-Instanzen arbeiten, die auf einem PC ausgeführt werden und versuchen, mit demselben Gerät zu kommunizieren. Beide Anwendungen werden den Port abwechselnd belegen und die Verbindung zum Gerät wird ständig unterbrochen.

Die Anwendung, die nicht mit dem Gerät kommunizieren soll, sollte in den OFFLINE-Modus versetzt werden.

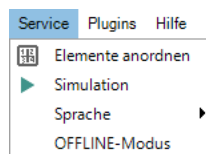


Abb. 3.4

Arbeitsgrundlagen

Der OFFLINE-Modus kann über den Menüpunkt **Service > OFFLINE-Modus** oder durch Klicken auf die Statusindikator **Gerät** aktiviert / deaktiviert werden (Abschn. 2.7). Mit dem nächsten Klick ist der OFFLINE-Modus deaktiviert.

3.5 Geräteinformationen


Um Informationen über die Software, das Zielgerät und das angeschlossene Gerät zu erhalten, nutzen Sie den Menüpunkt **Gerät > Informationen...** oder das Symbol  in der Symbolleiste.



Abb. 3.5

Das Fenster **Geräteinformationen** enthält folgende Informationen:

Zielgerät – das Gerät, für das das Projekt erstellt wurde

Angeschlossenenes Gerät – die Informationen über das an den PC angeschlossenen Gerät

Ins Projekt exportieren – die Schaltfläche ist nur aktiv, wenn ein Gerät PR114-224.8D4A.4RXXXX angeschlossen ist. Damit können die realen Ausgangstypen des angeschlossenen Geräts in das Projekt exportiert werden. Alternativ kann der Typ jedes Ausgangs manuell in der Property Box entsprechend der Hardware geändert werden.

3.6 Projektinformationen

Verwenden Sie den Menüpunkt **Datei > Projektinformationen**, um die Informationen zum Projekt anzusehen und zu ändern. Die Registerkarte **Allgemein** enthält Informationen über die Version der Software.

Softwareversion bei der Projekterstellung – die Version der Software, in der das Projekt erstellt wurde

Softwareversion bei der Projektänderung – die Version der Software, in der das Projekt geändert wurde

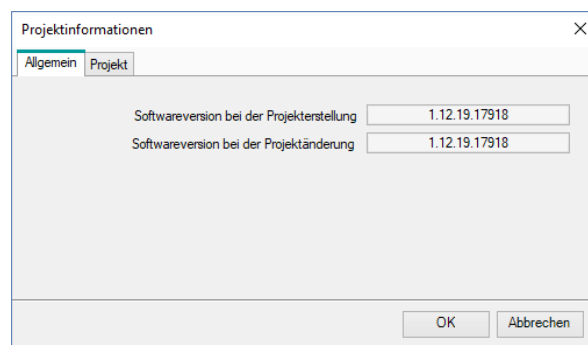


Abb. 3.6

Arbeitsgrundlagen

Die Registerkarte **Projekt** enthält Informationen zur Projektversion und ist nur für PR200 verfügbar.

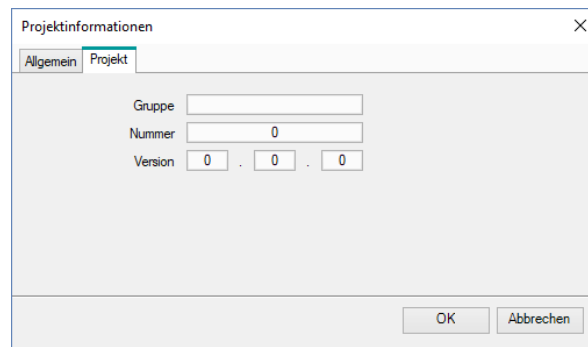


Abb. 3.7

Gruppe – Name der Projektgruppe

Nummer – Projektnummer innerhalb einer Gruppe

Version – Projektversion


Nachdem Sie die gewünschten Einträge vorgenommen haben, bestätigen Sie sie mit **OK** oder klicken Sie auf **Abbrechen**, um sie zu verwerfen.

3.7 Projekt auf das Gerät hochladen

Achtung! Das bereits im angeschlossenen Gerät gespeicherte Programm wird durch das neue ersetzt.

Gehen Sie wie folgt vor:

- verbinden Sie das Gerät mit dem PC
- schalten Sie das Gerät ein
- passen Sie ggf. die Port-Einstellungen an
- laden Sie das Projekt auf das Gerät hoch

Das Projekt kann über den Menüpunkt **Gerät > Programm auf Gerät übertragen** oder mit dem Klick auf das Symbol  in der Symbolleiste auf das Gerät hochgeladen werden. Nach dem Abschluss der Übertragung kann die Stromversorgung ausgeschaltet und das Gerät vom PC getrennt werden.

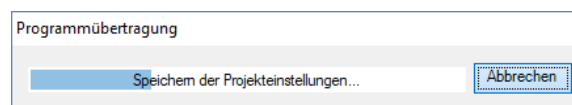


Abb. 3.8

Wenn das Zielgerät nicht mit dem angeschlossenen Gerät übereinstimmt, wird eine Warnmeldung angezeigt.

Hinweis: Nach Abschluss der Programmübertragung wechselt das Gerät in den Betriebsmodus und die Anwendung startet automatisch.

3.8 Firmware-Update / -Reparatur

Wenn eine neue ALP-Version eine neue Version der Firmware für das angeschlossene Gerät oder das Erweiterungsmodul enthält, werden Sie aufgefordert, die Firmware zu aktualisieren, bevor Sie ein Benutzerprogramm auf das Gerät hochladen. Es ist keine Internetverbindung erforderlich. Klicken Sie auf **Ja**, um das Update zu starten.

Hinweis: Stellen Sie sicher, dass die Stromversorgung des Geräts und der Erweiterungsmodule (falls vorhanden) sowie die sichere Verbindung zwischen dem

Arbeitsgrundlagen

PC, dem Gerät und den Erweiterungsmodulen (falls vorhanden) während des Updatevorgangs vorhanden sind.

Sie können die Firmware auch manuell über den Menüpunkt **Gerät > Firmware-Update** aktualisieren. Auf diese Weise kann die Firmware repariert werden, wenn der Firmware-Schaden erkannt wird (siehe entsprechende Bedienungsanleitung, Tabelle „Fehleranzeige“).

Hinweis: Das Anwenderprogramm wird nicht durch Firmware-Update betroffen sein.

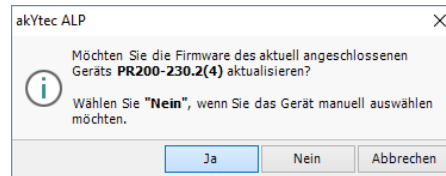


Abb. 3.9

Wenn Sie **Ja** auswählen, wird die Firmware des aktuell verbundenen und erkannten Geräts aktualisiert (repariert).

Wenn Sie **Nein** auswählen, werden Listen mit Geräten und Erweiterungsmodulen zur Auswahl angeboten (Abb. 3.10). Das geöffnete Fenster hat zwei Registerkarten: **Gerät** und **Erweiterungsmodul**. Auf diese Weise kann ein erzwungenes Firmware-Update durchgeführt werden (Abschn. 3.8.1).

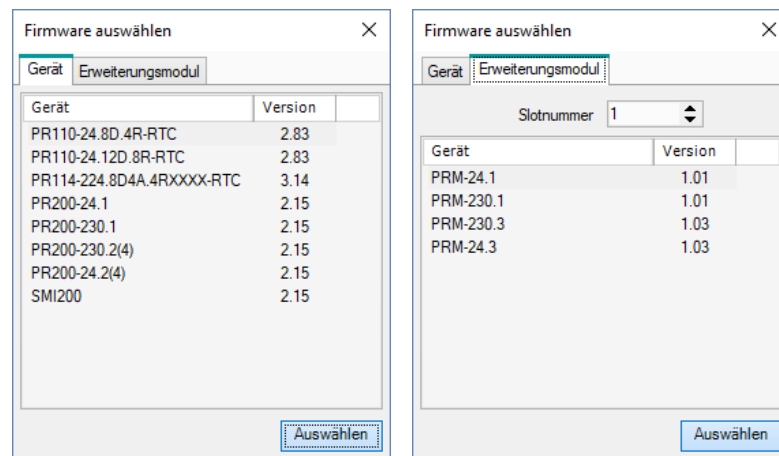


Abb. 3.10

Klicken Sie auf **Auswählen**, um die Auswahl zu bestätigen und die Aktualisierung (Reparatur) zu starten. Die Nachricht über das Update-Ergebnis wird nach Abschluss des Updates angezeigt.

3.8.1 Erzwungenes Firmware-Update / -Reparatur

Wenn die Firmware beschädigt ist (siehe entsprechende Bedienungsanleitung, Tabelle „Fehleranzeige“) und die automatische Erkennung des Geräts nicht möglich ist, sollte ein erzwungenes Firmware-Update verwendet werden.

Gehen Sie wie folgt vor:

1. Setzen Sie das Gerät in den Modus des erzwungenen Downloads (siehe Bedienungsanleitung des Geräts).
2. Wählen Sie den Menüpunkt **Gerät > Firmware-Update** aus. Es werden Listen mit Geräten und Erweiterungsmodulen zur Auswahl angeboten.
3. Wählen Sie das Gerät (Erweiterungsmodul).
4. Klicken Sie auf **Auswählen**, um die Auswahl zu bestätigen und den Update-/Reparatur-Prozess zu starten.

Arbeitsgrundlagen

Die Meldung über das Update-Ergebnis wird nach Abschluss des Updates angezeigt. Wenn das Basisgerät und das Erweiterungsmodul inkompatible Firmware-Versionen haben und das Benutzerprogramm ohne angeschlossenes Erweiterungsmodul in das Basisgerät hochgeladen wird, kann dies zu einem Fehler des Erweiterungsmoduls führen. Um diesen Fehler zu beheben, verwenden Sie ein erzwungenes Firmware-Update für das Erweiterungsmodul wie beschrieben, wobei Schritt 1 übersprungen werden soll.

4 Gerätekonfiguration

Die Konfiguration des Geräts ist Teil eines Projekts und kann über den Menüpunkt **Gerät > Konfiguration** festgelegt werden. Das Dialogfenster **Gerätekonfiguration** besteht aus zwei Teilen. Die konfigurierbaren Parameter des Geräts werden im Strukturbaum im linken Teil des Fensters und der Inhalt einer Gruppe wird im rechten Teil dargestellt.

Der Inhalt des Strukturbaums hängt vom Zielgerät ab und kann folgende Gruppen enthalten:

- Display
- Uhr
- Schnittstellen
- Erweiterungsmodule
- Eingänge
- Ausgänge

Alle Einstellungen werden im Projekt gespeichert, mit Ausnahme der Uhrzeiteinstellungen. Die Konfiguration ist auch ohne Anschluss des Geräts möglich.

4.1 Display

Wenn das Zielgerät ein Display hat, können folgende Parameter festgelegt werden:

Hintergrundbeleuchtung – die Dauer der Hintergrundbeleuchtung seit der letzten Benutzeraktivität

Helligkeit – 0...100 %

Kontrast – 0...100 %

Über die Schaltfläche **Lesen** können Sie die aktuellen Display-Einstellungen vom angeschlossenen Gerät auslesen werden.

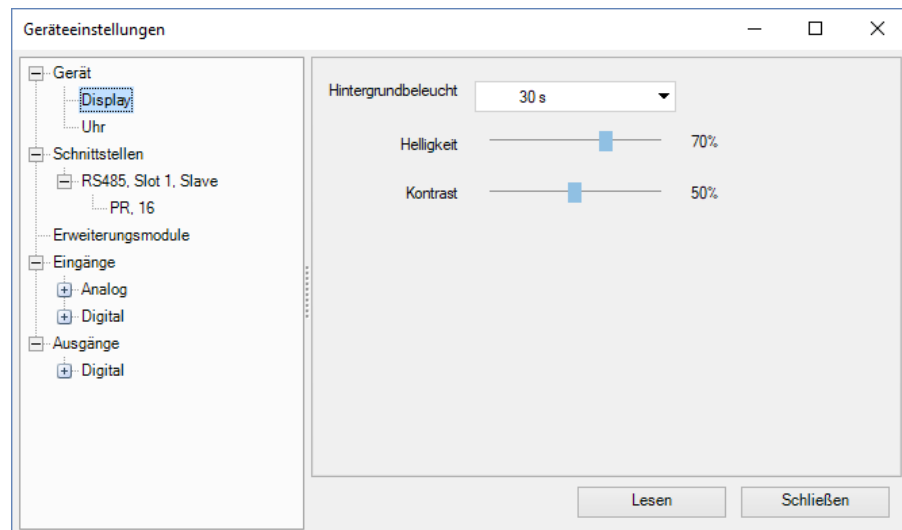


Abb. 4.1

4.2 Uhr

Wenn das Zielgerät über eine integrierte Echtzeituhr verfügt, können Datum und Uhrzeit in der Gruppe **Uhr** festgelegt werden.

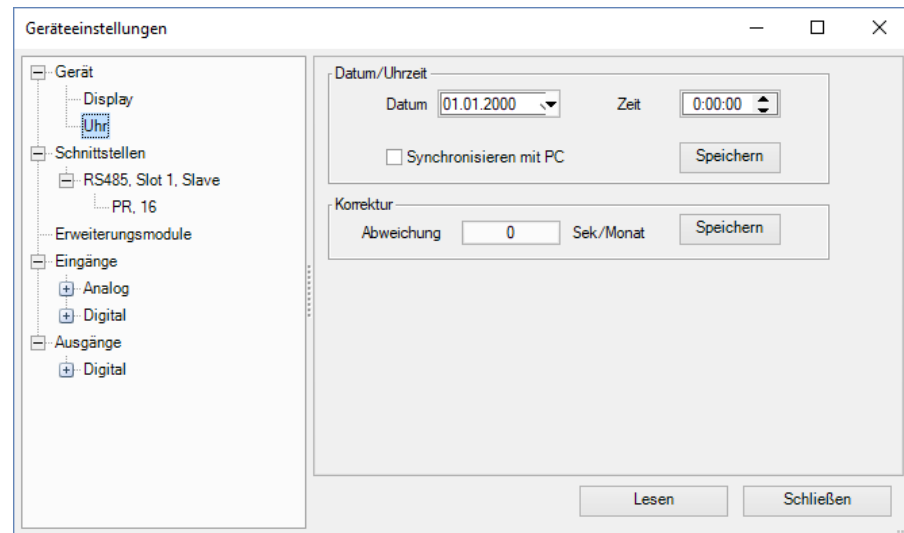


Abb. 4.2

Um die Geräteuhr mit der PC-Uhr zu synchronisieren, markieren Sie das Kontrollkästchen **Synchronisieren mit PC**. In diesem Fall sind die Felder **Datum** und **Uhrzeit** inaktiv. Um die Geräteuhr auf die neuen Werte einzustellen, klicken Sie auf die Schaltfläche **Speichern** im Bereich **Datum/Uhrzeit**.

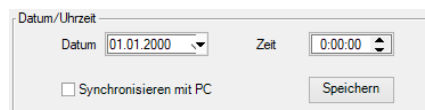


Abb. 4.3

Geben Sie die Zeitabweichung in Sekunden pro Monat im Feld **Abweichung** an, um die Zeitkorrektur einzustellen. Geben Sie einen negativen Wert ein, wenn die Geräteuhr zu schnell ist.

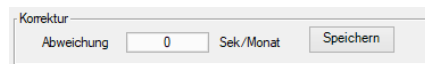


Abb. 4.4

Um die Uhrenkorrektur im Gerät zu speichern, klicken Sie im Bereich **Korrektur** auf die Schaltfläche **Speichern**.

Mit der Schaltfläche **Lesen** können die aktuellen Zeiteinstellungen vom angeschlossenen Gerät ausgelesen werden.

4.3 Schnittstellen

Wenn das Zielgerät über eine serielle Netzwerkschnittstelle RS485 verfügt, können deren Parameter in der Gruppe **Schnittstellen** festgelegt werden.

Standardmäßig ist eine Schnittstelle als Slave konfiguriert und dem Hardware-Slot 1 mit den folgenden Einstellungen zugeordnet: Mastergerät mit dem Namen PR und der Netzwerkadresse 16.

Wenn die Anzahl der Schnittstellen am Zielgerät geändert werden kann, können Schnittstellen in der Konfiguration hinzugefügt oder gelöscht werden, aber ihre Anzahl kann nicht die Anzahl der vorhandenen Slots übersteigen (Abschn. 4.3.2).

Wenn eine Schnittstelle als Master konfiguriert ist, können Slaves in der Konfiguration hinzugefügt oder entfernt werden, ihre Anzahl darf jedoch 16 nicht überschreiten.

Gerätekonfiguration

4.3.1 Arbeit mit Modbus

Mit akYtec ALP können die Geräte programmiert werden, die die Protokolle Modbus-RTU oder Modbus-ASCII (Master / Slave) unterstützen.

Hinweis: Die Geräte PR110 und PR114 können nur im Slave-Modus und nur mit dem Schnittstellenadapter PR-MI485 betrieben werden.

Um den Datenaustausch im Netzwerk über die RS485-Schnittstelle zu organisieren, ist ein Master-Gerät erforderlich. Es kann nur einen Master im Netzwerk geben.

Zykluszeit

Die Programmausführungszeit (Zykluszeit) wird abhängig von der Programmkomplexität automatisch angepasst (Auto-Tuning). Das Auto-Tuning beeinflusst den Datenaustausch über Modbus, da die Programmausführung eine höhere Priorität als die Abfrageverarbeitung hat. Wenn das Programm groß ist, kann dies die gesamte CPU-Zeit in Anspruch nehmen, und der Modbus-Datenaustausch wird nicht korrekt ausgeführt.

Um dieses Problem zu vermeiden, ist die Untergrenze für das Volumen des Modbus-Datenaustauschs reserviert: 50 Abfragen pro Sekunde. Somit können mindestens 50 Abfragen pro Sekunde ausgeführt werden, selbst wenn das Benutzerprogramm groß ist, und noch mehr, wenn das Programm klein ist und die Prozessorkapazität ausreichend ist. Wenn nicht genügend Zeit zum Abfragen aller Geräte zur Verfügung steht, sollte die Anzahl der Abfragen im Anwenderprogramm optimiert werden.

Die **Abfragezyklus**-Einstellung hängt von der Anzahl der abgefragten Variablen und der Abfragehäufigkeit im Programm ab. Es wird empfohlen, den **Abfragezyklus** auf 1 s einzustellen. In diesem Fall kann das Gerät bis zu 50 Variablen abfragen.

Abfragezeit

Die Abfragezeit ist die tatsächliche Zeit, die das Gerät benötigt, um alle Abfragen in einer Warteschlange auszuführen. Wenn die Warteschlange kurz ist, führt das Gerät alle Anfrage-Antwort-Zyklen aus und wartet, bis der angegebene Abfragezyklus abgelaufen ist (Abb. 4.5). Wenn die Warteschlange lang ist und die Abfrage länger dauert als der angegebene Abfragezyklus, fragt das Gerät alle Slaves in der kürzestmögliche Zeit ab.

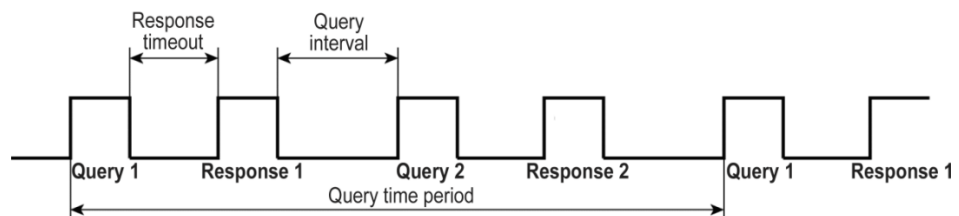


Abb. 4.5 Abfragezeitdiagramm

Um die Abfragezeit zu minimieren, wird Folgendes empfohlen:

Wenn eines oder mehrere der Slave-Geräte nicht angeschlossen oder vorübergehend nicht verfügbar sind, sollten Sie die Abfrage im Programm blockieren oder den Parameter **Timeout** für diese Geräte minimieren.

Beim Einstellen des Parameters **Abfragezyklus**, müssen die Anzahl der Slave-Geräte und die Gesamtzahl der Abfragen berücksichtigt werden. Wenn die Bearbeitungszeit aller Abfragen (Abfragezeit) länger als der **Abfragezyklus** ist, wird der Parameter ignoriert.

Mehrere Geräte in Netzwerk Abfragen

Die Slaves werden entsprechend der generierten Warteschlange von der kleinsten bis zur größten Adresse abgefragt. Im folgenden Beispiel wird der Slave mit der Adresse 8 zuerst, mit der Adresse 32 zuletzt abgefragt.



Abb. 4.6

Der **Abfragezyklus** kann für jeden Slave individuell eingestellt werden.

4.3.2 Schnittstelle hinzufügen / entfernen

Wenn das Gerät über einen Slot verfügt, für den keine Schnittstelle konfiguriert ist, kann eine entsprechende Schnittstelle über den Menüpunkt **Schnittstelle hinzufügen** im Kontextmenü hinzugefügt werden.

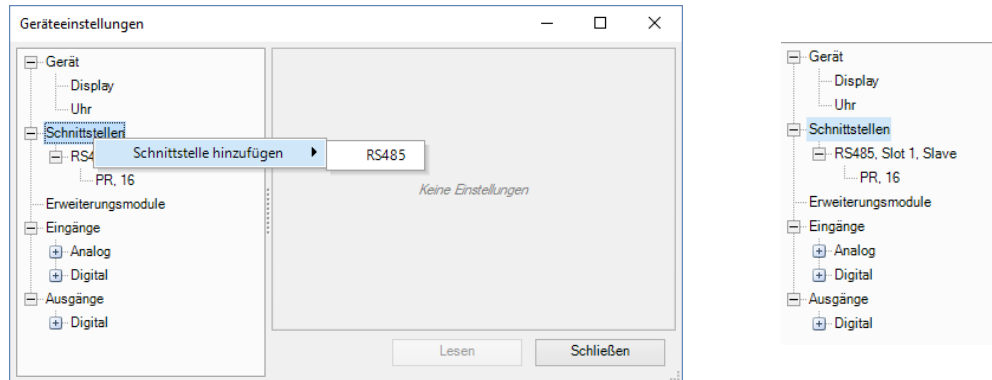


Abb. 4.7

Eine Schnittstelle des gewählten Typs wird mit den Standardeinstellungen hinzugefügt. Je nach Gerät kann eine Schnittstelle durch eine andere Art von Schnittstelle im Kontextmenü ersetzt oder entfernt werden.

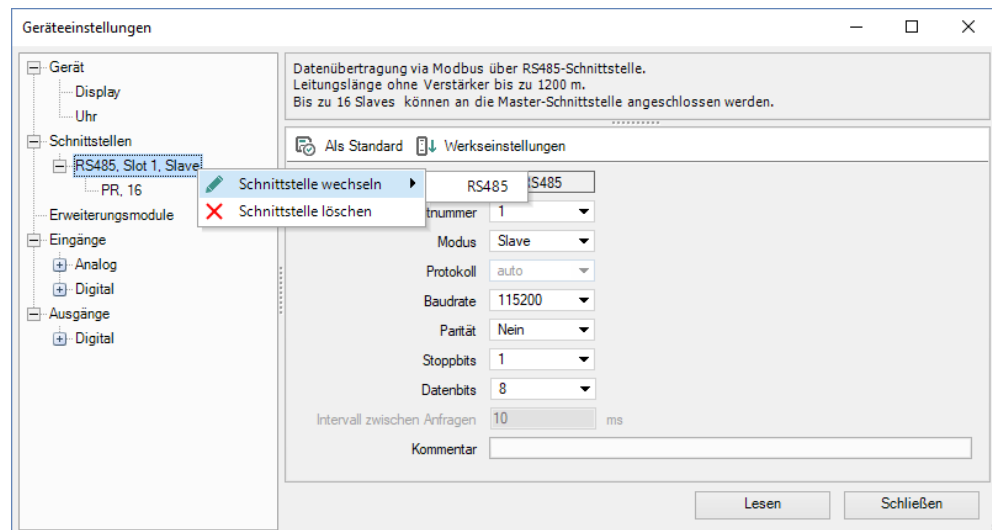


Abb. 4.8

4.3.3 RS485-Schnittstellenkonfiguration

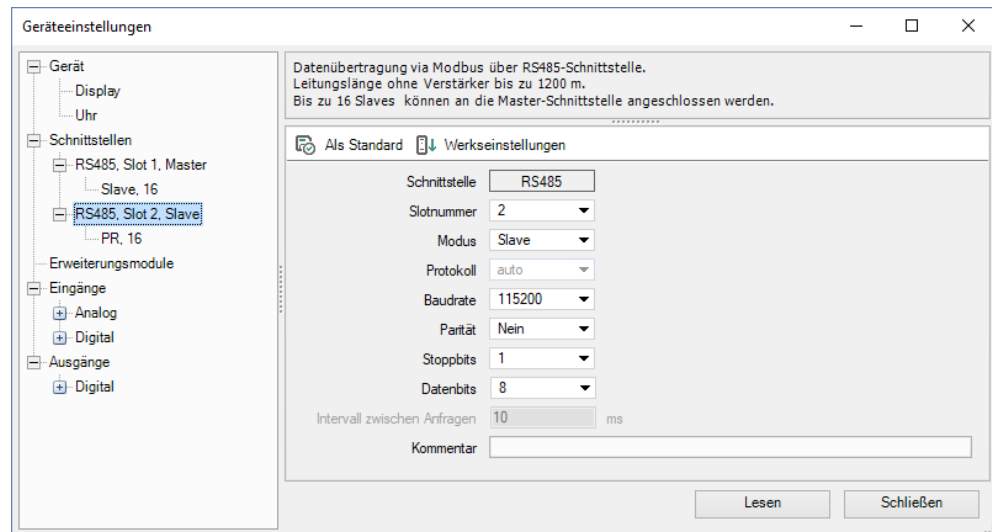


Abb. 4.9 Schnittstellenkonfiguration

Der Typ der Schnittstelle (RS485), die Nummer des zugewiesenen Slots und der Modus (Masters / Slave) werden im Strukturbaum angezeigt.

Um eine Verbindung über die Schnittstelle herzustellen, muss sie konfiguriert werden. Die Parameter der Schnittstelle werden im rechten Teil des Fensters angezeigt. Der Standardwert hängt vom Zielgerät ab. Die Parameter **Protokoll** und **Intervall zwischen Anfragen** sind nur im Master-Modus verfügbar. Im Slave-Modus sind sie inaktiv und grau dargestellt.

Das Symbol **Als Standard** wird verwendet, um die Einstellungen als Standardwerte für zukünftige Projekte zu speichern.

Das Symbol **Werkseinstellungen** wird verwendet, um die unveränderlichen Werkseinstellungen anzuwenden.

Die Schaltfläche **Lesen** wird verwendet, um die aktuellen Einstellungen vom angeschlossenen Gerät auszulesen.

Verwenden Sie die Schaltfläche **Schließen**, um die Einstellungen im Projekt zu speichern und den Dialog zu schließen.

4.3.3.1 Master-Modus

Jede Schnittstelle kann bis zu 16 Slaves steuern. Jeder Slave unterstützt bis zu 256 Variablen. Die Adressen und Namen der Variablen müssen nur dann eindeutig sein, wenn sie zum selben Slave gehören.

Im Master-Modus werden alle an die Schnittstelle angeschlossenen Slaves sequenziell abgefragt. Wählen Sie in der Parameterliste den Modus **Master** aus, legen Sie weitere Verbindungsparameter fest und fügen Sie die gewünschte Anzahl von Slaves über den Punkt **Slave hinzufügen** der Schnittstelle-Kontextmenü hinzu.

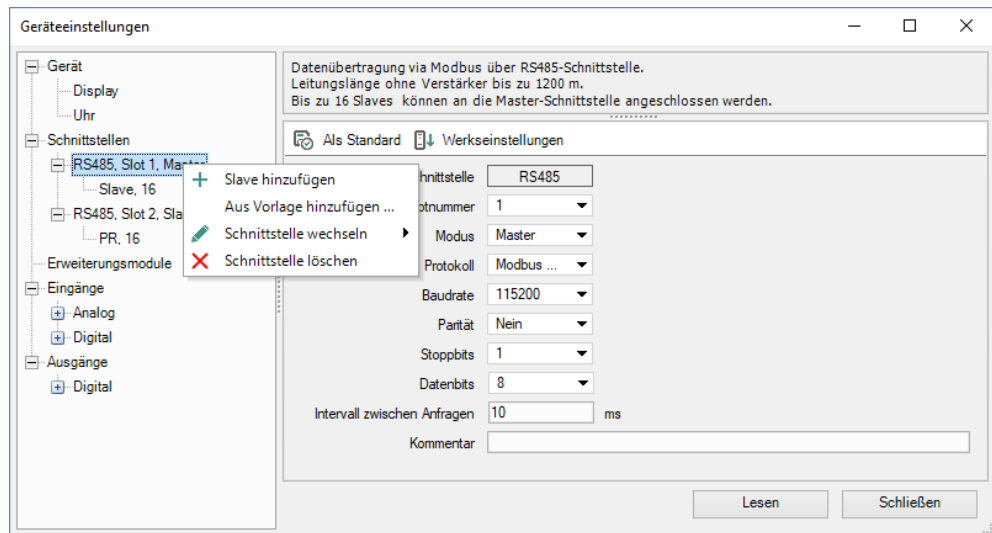


Abb. 4.10 Master-Konfiguration im Master-Modus

Das hinzugefügte Slave-Gerät wird mit seinem Namen und seiner Adresse im Strukturbaum unterhalb der Schnittstelle angezeigt. Wählen Sie einen Slave aus, um ihn im rechten Teil des Fensters zu konfigurieren.

Die Parameter des Slaves können im oberen Teil des Fensters angegeben werden (Abb. 4.11).

Um ein Slave zu löschen, verwenden Sie das Kontextmenü oder das Symbol **X** *Slave entfernen*.

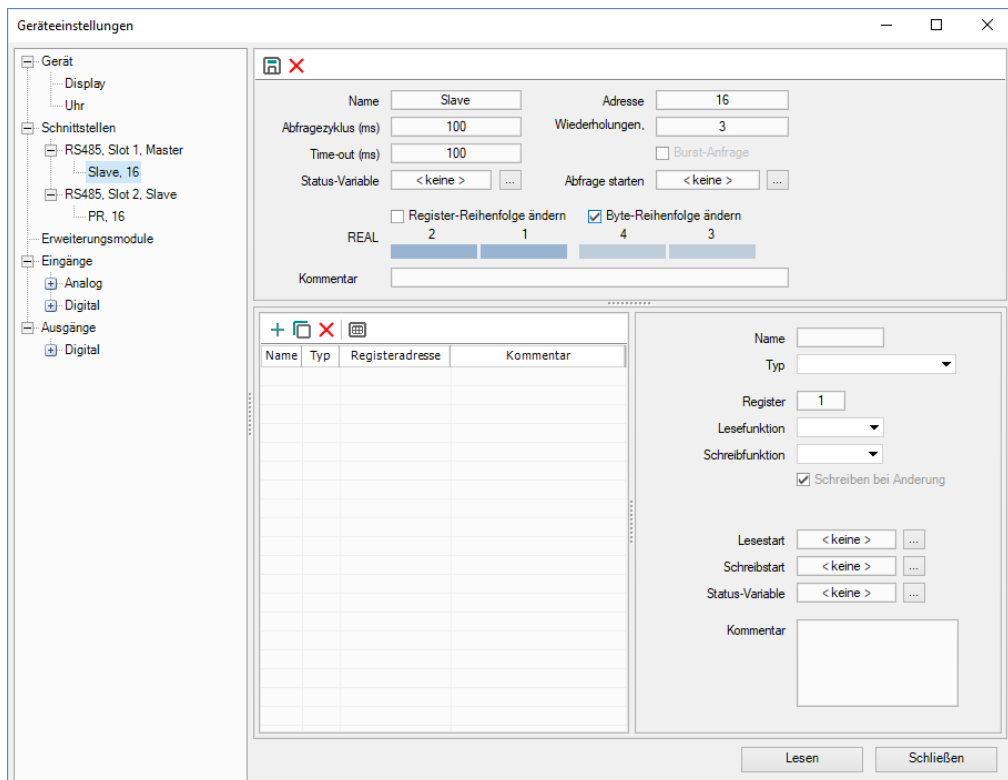




Abb. 4.11 Slave-Konfiguration im Master-Modus

- **Name** – der Name des Slaves, der im Strukturbaum angezeigt wird
- **Adresse** – die Netzwerkadresse des Slaves
- **Abfragezyklus (ms)** – das Zeitintervall zwischen Abfragen. Der gültige Bereich ist 0...65535 ms.

Gerätekonfiguration

- **Timeout (ms)** – die Zeit, die eine Abfrage dauern kann, bevor der Versuch als fehlgeschlagen gilt. Der gültige Bereich ist 0...65535 ms.
- **Wiederholungen, max.** – die Anzahl der fehlgeschlagenen Abfragen, bis die Abfrage gestoppt wird und der Status des Geräts sich ändert. Der gültige Bereich ist 0...255.
- **Burst-Anfrage** – eine Gruppenabfrage von aufeinanderfolgenden Registern zur Erhöhung des Datendurchsatzes
- **Statusvariable** – zur Aufzeichnung des Gerätestatus wählen Sie mit dem Symbol  eine BOOL-Variable aus:
 - 1 – das Gerät funktioniert ordnungsgemäß
 - 0 – das Gerät ist nicht angeschlossen
- **Abfrage starten** – zur Steuerung der Abfrage wählen Sie mit dem Symbol  eine BOOL-Variable aus:
 - 0 – Abfrage deaktiviert
 - 1 – Abfrage aktiviert
- **Register-Reihenfolge ändern** – bestimmt die Register-Reihenfolge in den Doppelregister-Variablen
- **Byte-Reihenfolge ändern** – bestimmt die Byte-Reihenfolge im Register
- **Kommentar** – ein Textkommentar, um das Gerät zu beschreiben

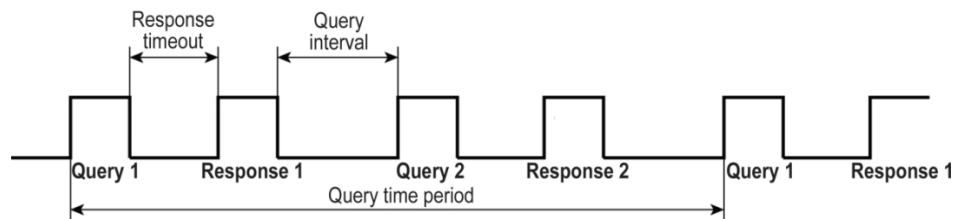
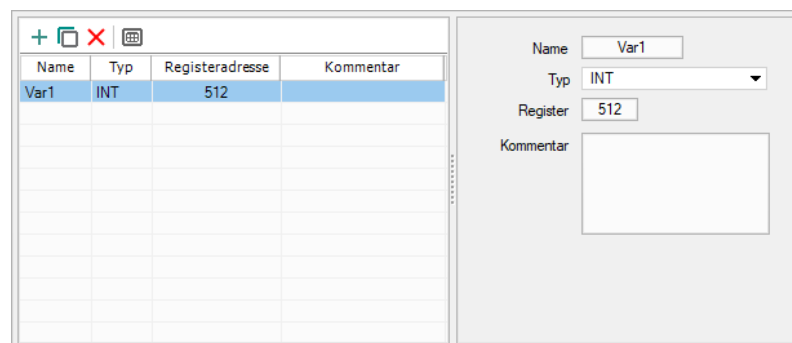


Abb. 4.12 Abfragezeit-Diagramm

Die Liste der Variablen, die von diesem Slave abgefragt werden sollen, befindet sich im unteren Teil des Fensters. Jede Variable, die in dieser Liste erstellt wurde, finden Sie in der Variablen-tabelle unter der Registerkarte **Netzwerk, Slot X** mit einer separaten Liste von Variablen für jedes Slave-Gerät (Abschn. 5.5).

Fügen Sie eine Variable hinzu, indem Sie auf das Symbol **+ Neue Variable** klicken, und legen Sie deren Eigenschaften fest.




Name	Typ	Registeradresse	Kommentar
Var1	INT	512	

Abb. 4.13 Konfiguration von Variablen im Master-Modus

- **Name** – der Name der Variablen
- **Typ** – der Datentyp der Variablen: BOOL, INT oder REAL
- **Register** – die Registeradresse
- **Bit** – die Nummer des Bits des Registers (0...15) (nur für die BOOL-Variablen)

Gerätekonfiguration

- **Lese-Funktion / Schreib-Funktion** – Auswahl der Lese-/Schreibfunktion oder Deaktivieren des Lesens/Schreibens.
- **Anzahl der Register** – die Anzahl der von der Variablen belegten Register (nur für die INT-Variablen)
- **Lesevorgang starten** – eine BOOL-Variable für das erzwungene Lesen der abgefragten Variablen zuweisen
- **Schreibvorgang starten** – eine BOOL-Variable für das erzwungene Schreiben der abgefragten Variablen zuweisen
- **Statusvariable** – eine INT-Variable zur Aufzeichnung des Fehlercodes zuweisen
- **Kommentar** – ein Textkommentar zur Beschreibung der abgefragten Variablen

Um mehrere Variable mit den gleichen Einstellungen zu erstellen, markieren Sie eine Variable und klicken auf das Symbol  **Duplizieren**.

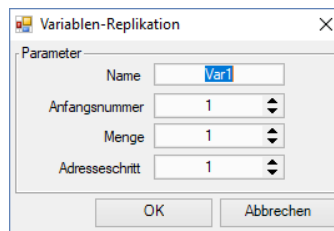



Abb. 4.14 Variablen duplizieren

- **Name** – der Name der duplizierten Variablen
- **Startnummer** – die Anfangsnummer, die zum Namen der duplizierten Variablen hinzugefügt wird
- **Menge** – die Anzahl der duplizierten Variablen
- **Adressschritt** – die Adresserhöhung

Klicken Sie auf **OK**, um die replizierten Variablen zur Liste der Variablen hinzuzufügen. Die Variablen werden in benachbarten Registerzellen mit aufeinanderfolgenden Adressen gespeichert.

Um eine Variable aus der Liste zu entfernen, nutzen Sie das Symbol  **Löschen**.

4.3.3.2 Gerätevorlage

Ein Slave-Gerät in der Konfigurationsmaske kann mit seinen Parametern und Variablen als Vorlage gespeichert und in weiteren Projekten verwendet werden. Verwenden Sie das Kontextmenüpunkt oder das Symbol  **Slave als Vorlage speichern** (Abb. 4.15). Die Vorlage wird als eine Datei mit der Erweiterung *.dvtp gespeichert.

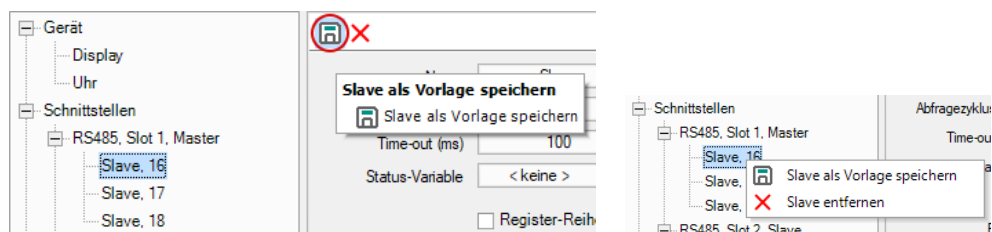


Abb. 4.15 Slave-Kontextmenü

Ein Slave als Vorlage kann einem Master mit dem Kontextmenüpunkt **Aus Vorlagen hinzufügen...** hinzugefügt werden (Abb. 4.16)

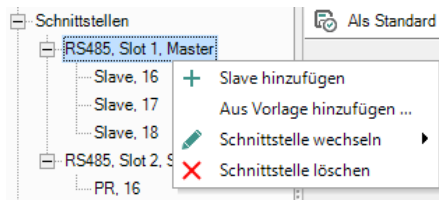


Abb. 4.16 Master-Kontextmenü

4.3.3.3 Slave-Modus

Eine RS485-Schnittstelle, die dem Strukturelement **Schnittstellen** hinzugefügt ist, hat den Standardmodus Slave und den Standard-Master mit dem Namen PR und der Adresse 16 unten hinzugefügt. Wählen Sie die Schnittstelle aus, um die Verbindungsparameter festzulegen.

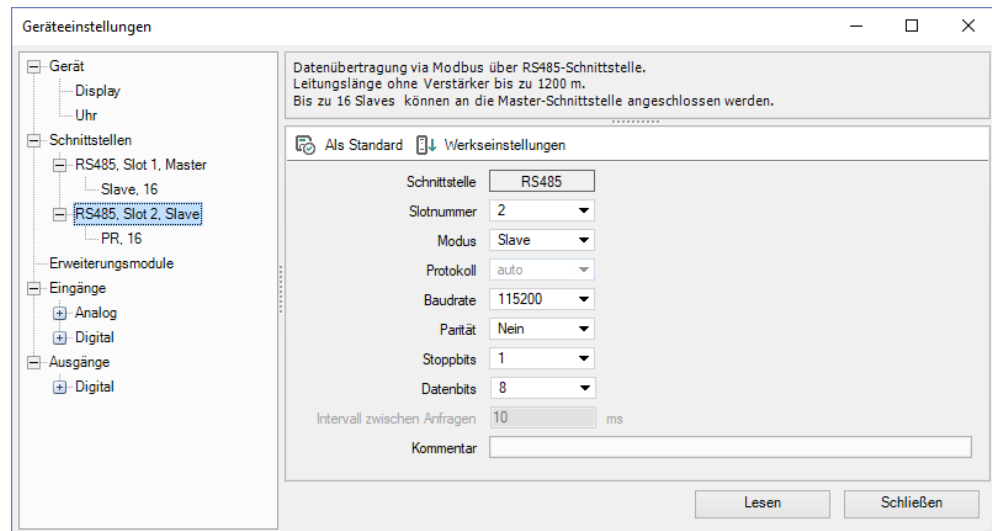


Abb. 4.17 Slave-Konfiguration im Slave-Modus

Wählen Sie im Baum den Master aus, um die Parameter für den Datenaustausch festzulegen.

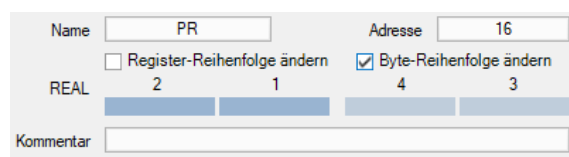


Abb. 4.18 Master-Konfiguration im Slave-Modus

Die gemeinsamen Parameter für den Datenaustausch können im oberen Fensterteil festgelegt werden.

- **Name** – der im Strukturbaum angezeigte Name des Masters
- **Adresse** – die Netzwerkadresse des Masters
- **Register-Reihenfolge ändern** – die Register-Reihenfolge in den Doppelregister-Variablen
- **Byte-Reihenfolge ändern** – die Byte-Reihenfolge im Register
- **Kommentar** – ein Textkommentar, um das Gerät zu beschreiben

Die Liste der vom Master abgefragten Variablen befindet sich im unteren Teil des Fensters. Jede Variable, die in dieser Liste erstellt wurde, finden Sie in der Variablentabelle unter der Registerkarte **Netzwerk, Slot X** (Abschn. 5.5).

Gerätekonfiguration

Fügen Sie eine Variable hinzu, indem Sie auf das Symbol **+** *Neue Variable* klicken, und legen Sie deren Eigenschaften fest.

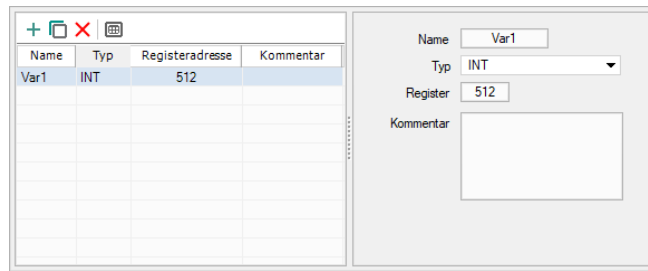


Abb. 4.19 Konfiguration von Variablen im Slave-Modus

- **Name** – der Name der Variablen
- **Typ** – der Datentyp der Variablen: BOOL, INT oder REAL
- **Register** – die Registeradresse. Der Bereich der verfügbaren Adressen ist in der Bedienungsanleitung des Geräts angegeben.
- **Kommentar** – ein Textkommentar zur Beschreibung der abgefragten Variablen

Um mehrere Variable mit den gleichen Einstellungen zu erstellen, markieren Sie eine Variable und klicken auf das Symbol **☐** *Duplizieren*.

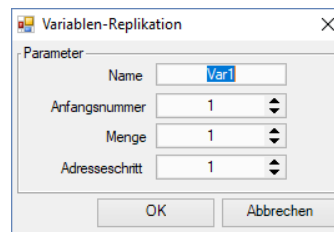


Abb. 4.20 Variablen duplizieren

- **Name** – der Name der duplizierten Variablen
- **Startnummer** – die Anfangsnummer, die zum Namen der duplizierten Variablen hinzugefügt wird
- **Menge** – die Anzahl der duplizierten Variablen
- **Adressschritt** – die Adresserhöhung

Klicken Sie auf **OK**, um die duplizierten Variablen zur Liste der Variablen hinzuzufügen.

Um eine Variable aus der Liste zu entfernen, nutzen Sie das Symbol **✕** *Löschen*.

4.4 Erweiterungsmodule

Es können bis zu zwei E/A-Erweiterungsmodule vom Typ PRM an PR200 angeschlossen werden. Weitere Informationen zu Erweiterungsmodulen finden Sie in der PRM-Bedienungsanleitung.

Um die E/A-Punkte des Moduls im Schaltprogramm zu verwenden, fügen Sie das Modul über das Kontextmenü der Gruppe **Erweiterungsmodule** hinzu (Abb. 4.21).

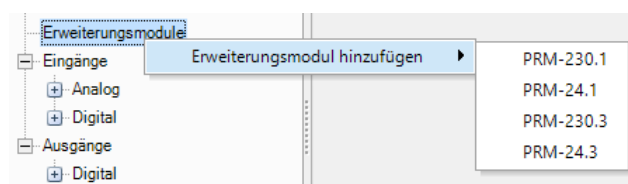


Abb. 4.21

Gerätekonfiguration

Die zusätzlichen E/A-Punkte der hinzugefügten Module können in den Zweigen **Eingänge** und **Ausgänge** konfiguriert werden (Abschn. 4.5, 4.6). Sie werden im Strukturbaum als **Ix(y)** und **Qx(y)** angezeigt, wobei **x** die Ordnungsnummer des E/A-Punkte des Moduls und **y** die Ordnungsnummer des Moduls ist, die vom Basisgerät aus gezählt wird.

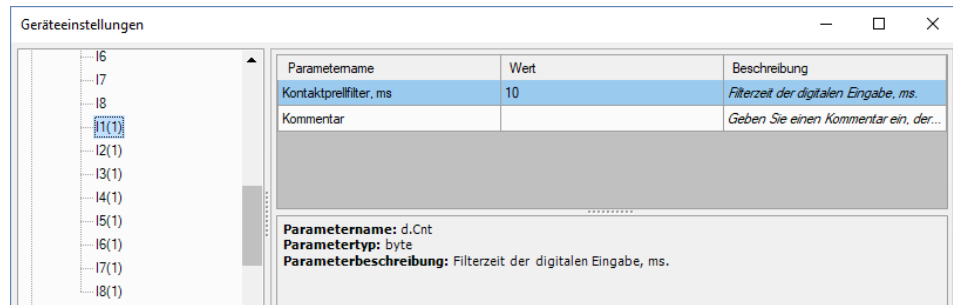


Abb. 4.22

Vor dem Hochladen des Projekts in das Basisgerät müssen alle Module über den internen Bus an PR200 angeschlossen und eingeschaltet sein. Die Firmware des Moduls wird beim Hochladen des Projekts mit der aktuellen ALP-Version synchronisiert.

4.5 Eingänge

Der Inhalt des Zweigs **Eingänge** ist abhängig von den Ressourcen des Zielgeräts. Es können analoge und/oder digitale Eingänge sein (Abb. 4.23, 4.24).

Der Parameter **Kommentar** ist eine allgemeine Einstellung für alle Eingangstypen. Der Text in diesem Feld wird in einem QuickInfo angezeigt, wenn sich die Maus über dem Eingang im Arbeitsbereich befindet. Der Text kann auch in der Property Box eingegeben werden.

Weitere Informationen zur Konfiguration der Eingänge finden Sie in der Bedienungsanleitung des Geräts.

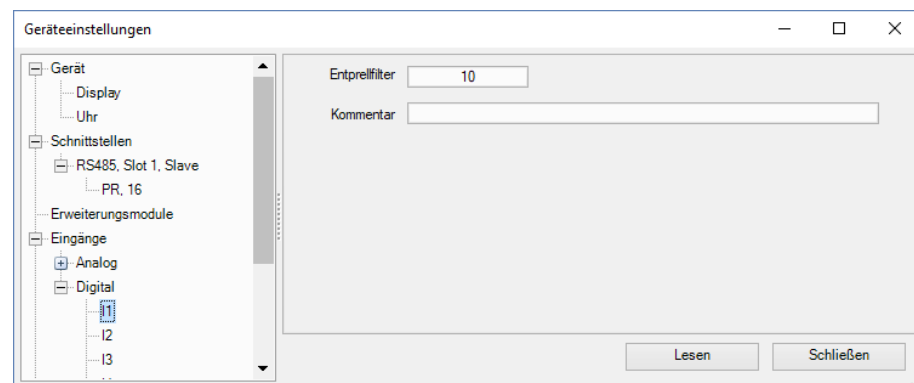


Abb. 4.23 PR200 Konfiguration eines digitalen Eingangs

Andere Eingangsparameter hängen von dem Typ des Eingangs und des Gerät ab.

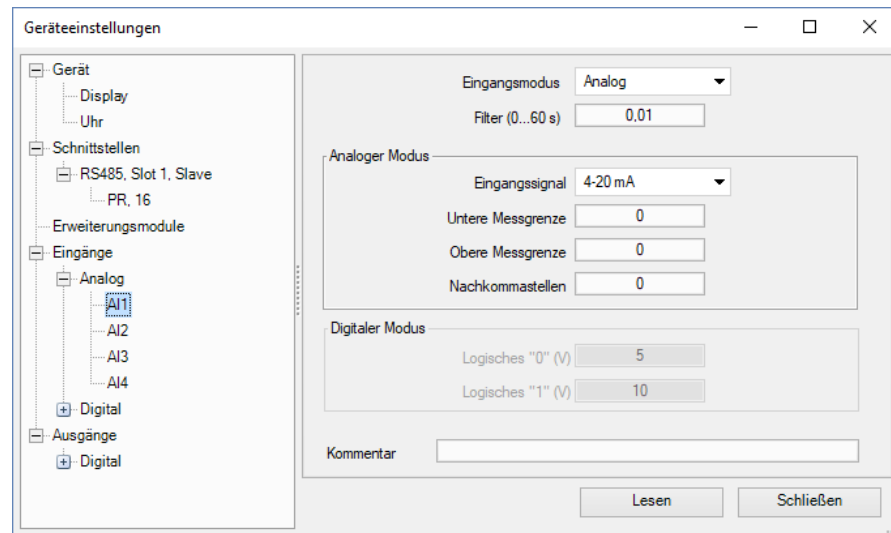


Abb. 4.24 PR200 Konfiguration eines analogen Eingangs

4.6 Ausgänge

Der Inhalt des Zweigs **Ausgänge** ist abhängig von den Ressourcen des Zielgeräts. Es können analoge und/oder digitale Ausgänge sein (Abb. 4.24).

Der Parameter **Kommentar** ist eine allgemeine Einstellung für alle Ausgangstypen. Der Text in diesem Feld wird in einem QuickInfo angezeigt, wenn sich die Maus über dem Ausgang im Arbeitsbereich befindet. Der Text kann auch in der Property Box eingegeben werden.

Weitere Informationen zur Konfiguration der Ausgänge finden Sie in der Bedienungsanleitung des Geräts.

Die digitalen Ausgänge des Erweiterungsmoduls haben einen zusätzlichen Parameter **Sicherer Zustand**. Der Parameter bestimmt den Ausgangszustand, falls die Verbindung zwischen dem Modul und dem Basisgerät unterbrochen wird.

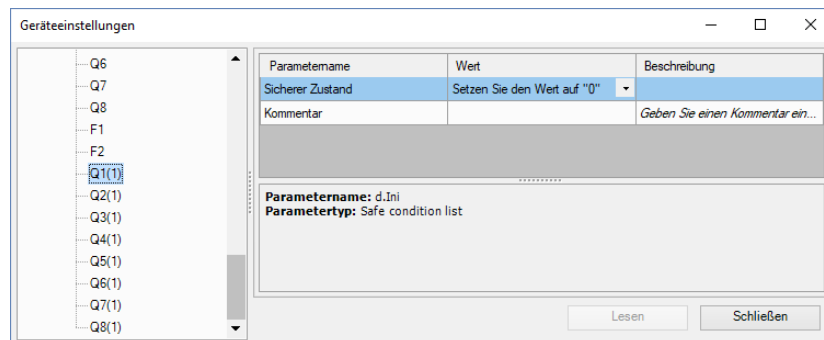


Abb. 4.25 PRM Ausgangs-Konfiguration

4.7 Kalibrierung

In diesem Abschnitt finden Sie nur allgemeine Informationen zur Kalibrierung von analogen Ein- oder Ausgängen. Detaillierte Informationen zur Kalibrierung finden Sie in der Bedienungsanleitung des Geräts.

Wenn eine Kalibrierung von analogen Ein- oder Ausgängen erforderlich ist, verwenden Sie den Menüpunkt **Gerät > Kalibrierung....**

Der Menüpunkt ist nur aktiv, wenn ein Gerät angeschlossen ist. Wählen Sie im Dialogfeld das Kalibrierungsziel (Ein- oder Ausgänge) aus.



Abb. 4.26

Nach der Auswahl des Kalibrierungsziels wird die Ausführung des Programms auf dem Gerät gestoppt. Das Programm wird nach erfolgreichem Abschluss der Kalibrierung erneut gestartet.

4.7.1 Eingangs-Kalibrierung

Verbinden Sie eine Referenzsignalquelle mit den zu kalibrierenden Eingängen. Starten Sie die Kalibrierung, wählen Sie den Typ des am Eingang angeschlossenen Signals und stellen Sie die Kalibrierungsparameter im geöffneten Dialog ein.

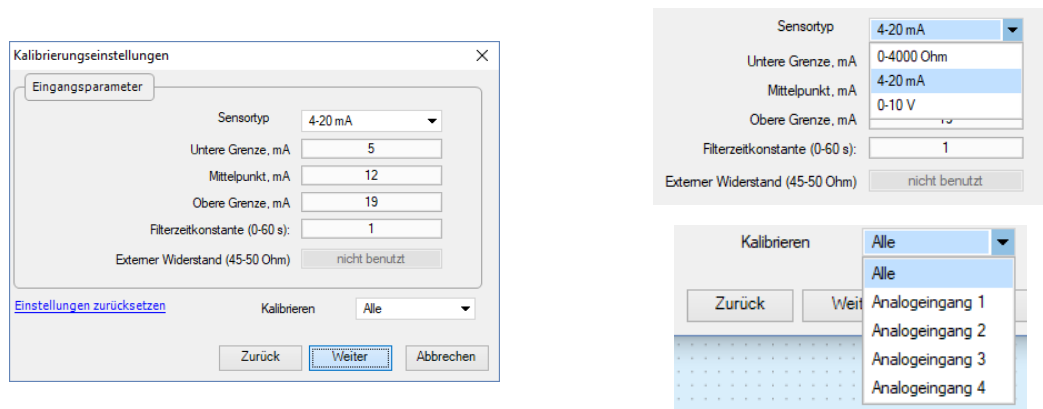


Abb. 4.27 PR200 Eingangs-Kalibrierung

Verwenden Sie die Option **Einstellungen zurücksetzen**, um die Standardeinstellungen für die Kalibrierung anzuwenden.

Verwenden Sie die Liste **Eingang auswählen**, um den zu kalibrierenden Eingang auszuwählen, klicken Sie auf die Schaltfläche **Weiter** und folgen Sie den Anweisungen.

4.7.2 Ausgangs-Kalibrierung

Vor der Kalibrierung eines analogen Ausgangs, bereiten Sie das entsprechende Messgerät vor, starten Sie die Kalibrierung und befolgen Sie die Anweisungen. Messen Sie das Signal an dem oben rechts im Fenster angegebenen Ausgang und geben Sie den Wert in das Eingabefeld ein.

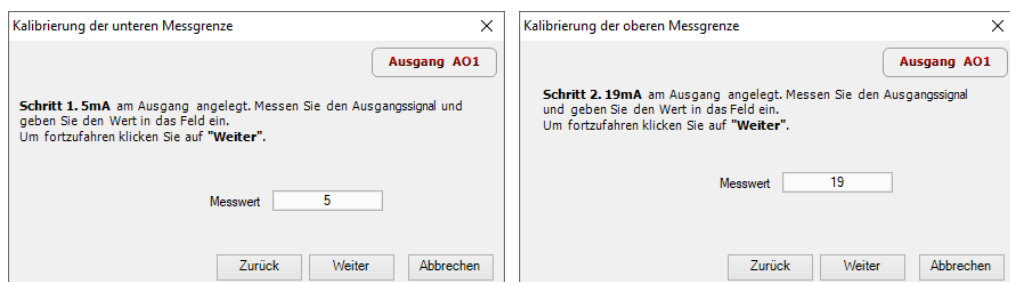


Abb. 4.28 PR200 Ausgangs-Kalibrierung

Gerätekonfiguration

Bei Bedarf, gehen Sie auf die gleiche Weise mit den anderen Ausgängen vor. Nach Abschluss der Kalibrierung erscheint eine Meldung über die Kalibrierergebnisse.

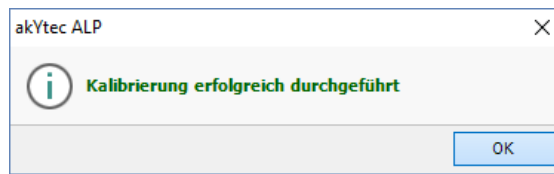


Abb. 4.29

4.8 Zielgerät ändern

Das Zielgerät eines Projekts kann über den Menüpunkt **Datei > Zielgerät ändern** geändert werden. Eine Liste der Geräte, auf die Sie das Projekt übertragen können, wird angezeigt. Wählen Sie das Gerät aus und bestätigen Sie mit OK. Überprüfen und reparieren Sie defekte Links, falls vorhanden. Das Programm kann durch Simulation überprüft werden (Abschn. 7.11). Speichern Sie das geänderte Projekt.

Beachten Sie die Ersetzungsregeln:

1. Die Größe des Arbeitsbereichs wird automatisch an die geänderte Anzahl der E/A-Punkte angepasst.
2. Das vom Benutzer konfigurierte Layout der E/A-Punkte bleibt erhalten. Neue E/A-Punkte werden nach vorhandenen E/A-Punkten des ursprünglichen Projekts platziert.
3. Die Verbindungen von E/A-Punkten, deren Datentyp geändert wurde, werden getrennt.
4. Wenn die Anzahl der E/A-Punkte geringer wird als im ursprünglichen Projekt, werden auch die Verbindungen der entfernten E/A-Punkte getrennt.
5. Wenn im ursprünglichen Projekt Erweiterungsmodule vorhanden waren, werden diese mit ihren Verbindungen in die neue Konfiguration übernommen.
6. Einstellungen von analogen E/A-Punkten werden übernommen, wenn auf dem neuen Gerät analoge E/A-Punkte vorhanden sind.
7. Netzwerkschnittstellen werden unverändert übernommen.
8. Display-Einstellungen werden unverändert übernommen.
9. Variablen werden unverändert übernommen.

5 Variable

Um alle im Projekt angelegten Variablen zu sehen, klicken Sie auf das Symbol in der Symbolleiste oder verwenden Sie den Menüpunkt **Gerät > Variablentabelle**.

Die Variablen sind in drei Gruppen unterteilt, von denen jede eine eigene Registerkarte in der Tabelle hat:

- Standard (Abschn. 5.3)
- Service (Abschn. 5.4)
- Netzwerk (Abschn. 5.5)

Standard- und Netzwerkvariablen können erstellt, dupliziert oder gelöscht werden. Die duplizierte Variable ist eine Kopie der ausgewählten Variablen, die in der nächsten freien Registerzelle gespeichert ist. Aufeinanderfolgende Indexe, beginnend mit 2, werden zum Namen jeder duplizierten Variablen hinzugefügt.

Zum Erstellen einer neuen Variablen können Sie

- Symbolleistensymbol
- Tastenkombination **Strg+N**

verwenden, oder einen neuen Variablennamen in der letzten Zeile schreiben.

Wenn Sie nach einer erfolglosen Suche eine neue Variable erstellen, wird der im Suchfeld eingegebene Name als Name der neuen Variablen vorgeschlagen.

Um eine vorhandene Variable zu duplizieren, können Sie

- Symbolleistensymbol
- Tastenkombination **Strg+D**
- Kontextmenüpunkt **Duplizieren**

verwenden.

Zum Löschen einer Variablen können Sie

- Symbolleistensymbol
- Taste **Entf**
- Kontextmenüpunkt **Löschen**

verwenden.

Servicevariablen können weder erstellt noch gelöscht werden.

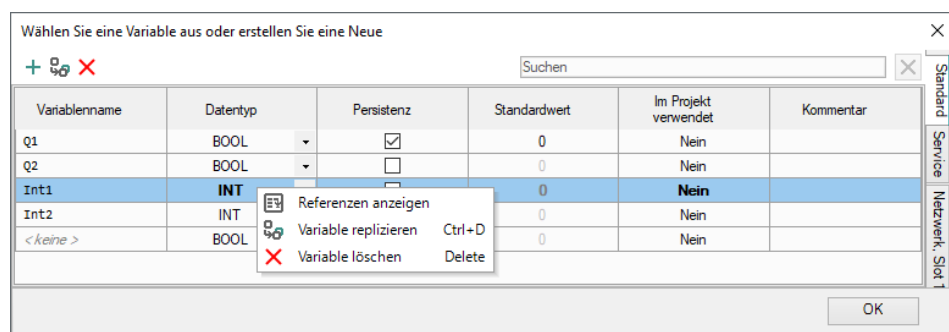


Abb. 5.1

Die Zeilen in der Variablentabelle können nach jeder Spalte sortiert werden.

5.1 Eigenschaften

Tabelle 5.1 Variablen-Eigenschaften

Name	Der Name der Variablen
Datentyp	BOOL, INT oder REAL (Abschn. 5.2)

Variable

Persistenz	Nur für Standardvariablen verfügbar. Die Variable wird im nicht-flüchtigen Speicher des Geräts gespeichert und ist somit eine Retain-Variable . Detaillierte Informationen zur Speicherzeit und zur Speichergröße finden Sie in der Bedienungsanleitung des Geräts.
Standardwert	Nur für Retain-Variablen verfügbar. Der Wert zum ersten Start des Programms, bis ein neuer Wert zugewiesen wird.
Verwendet im Projekt	Die Variable ist mit einem Baustein im Programm verknüpft
Kommentar	Text, der in einem QuickInfo im Schaltprogramm angezeigt wird, wenn sich die Maus über der Variablen befindet

5.2 Datentyp

Im Programm werden folgende Typen von Variablen verwendet:

- Boolesche (**BOOL**)
- Ganzzahlige (**INT**)
- Reelle (**REAL**)

Hinweis: *Verschiedene Geräte können Einschränkungen hinsichtlich der Unterstützung bestimmter Typen von Variablen haben.*

1. BOOL

Eine Variable dieses Typs hat nur zwei mögliche Werte: 1 (**Wahr**) oder 0 (**Falsch**).

Die Verbindungslinien zwischen BOOL-Variablen werden im Schaltprogramm grau dargestellt.

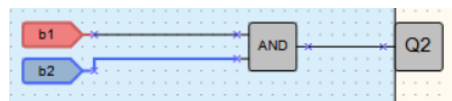


Abb. 5.2 BOOL-Verbindungen

2. INT

Eine Variable dieses Typs ist eine vorzeichenlose Ganzzahl im Bereich 0...4.294.967.295 (4 Byte).

Die Verbindungslinien zwischen INT-Variablen werden im Schaltprogramm rot dargestellt.

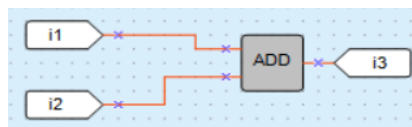


Abb. 5.3 INT-Verbindungen

3. REAL

Eine Variable dieses Typs hat einen Wert im Bereich $-3,402823e+38 \dots 3,402823e+38$.

Sie wird durch eine Gleitkommazahl von einfacher Genauigkeit (4 Byte) dargestellt.

Die Verbindungslinien zwischen REAL-Variablen werden im Schaltprogramm violett dargestellt.

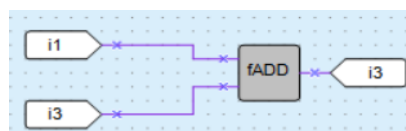


Abb. 5.4 REAL-Verbindungen

Variable

5.3 Standardvariablen

Diese sind allgemeine Variablen, die für Datenaustausch zwischen Bausteinen des Schaltprogramms, Eingängen, Ausgängen und Displayformen verwendet werden.

Die Standardvariablen sind in der Variablentabelle unter der Registerkarte **Standard** aufgelistet.

Um eine Variable zu erstellen, wählen Sie eine leere Zeile in der Tabelle aus, geben Sie den Variablennamen ein und wählen Sie den Datentyp. Weitere Parameter sind optional. Die erstellte Variable kann im Projekt verwendet werden.

Verwenden Sie den Punkt **Referenzen anzeigen** im Variablen-Kontextmenü, um zu sehen, wo die Variable im Projekt verwendet ist.

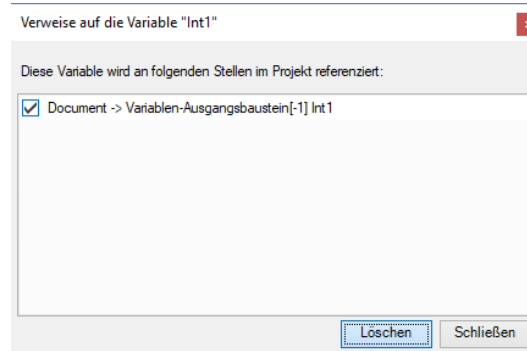


Abb. 5.5

Im Dialogfenster **Verweise auf die Variable** wählen Sie die Verknüpfung, die Sie löschen möchten, und klicken auf **Löschen**.

Um eine Variable aus der Tabelle zu entfernen, verwenden Sie im Kontextmenü die Option **Variable löschen**.

5.4 Servicevariablen

Die Servicevariablen sind mit den Geräteeinstellungen verknüpft und können je nach Gerät unterschiedlich sein. Servicevariable beziehen sich auf Hardwarefunktionen wie Echtzeituhr, Schnittstellenkarte im Slot usw. und können nicht gelöscht werden. Die Zugriffsberechtigungen auf Servicevariablen können eingeschränkt sein.

Die Servicevariablen sind in der Variablentabelle unter der Registerkarte **Service** aufgeführt.

Die Bausteine der Servicevariablen sind im Schaltprogramm grau hinterlegt.

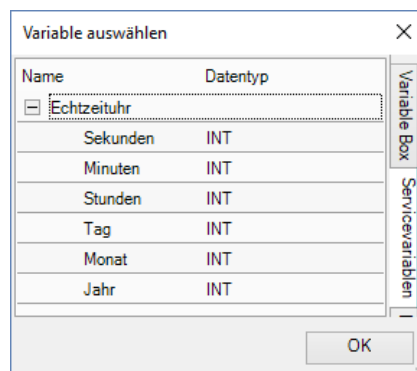


Abb. 5.6 Servicevariablen in der Tabelle



Abb. 5.7 Servicevariablen im Schaltbild

5.5 Netzwerkvariablen

Jeder Schnittstellen-Slot hat eine separate Registerkarte in der Tabelle.

Variable

Wenn die Schnittstelle als Master konfiguriert ist, gibt es für jedes Slave-Gerät separate Registerkarten innerhalb der Slot-Registerkarte (Abb. 5.8). Die Registerkarte Slave enthält die Variablen, die für dieses Slave-Gerät abgefragt werden sollen.

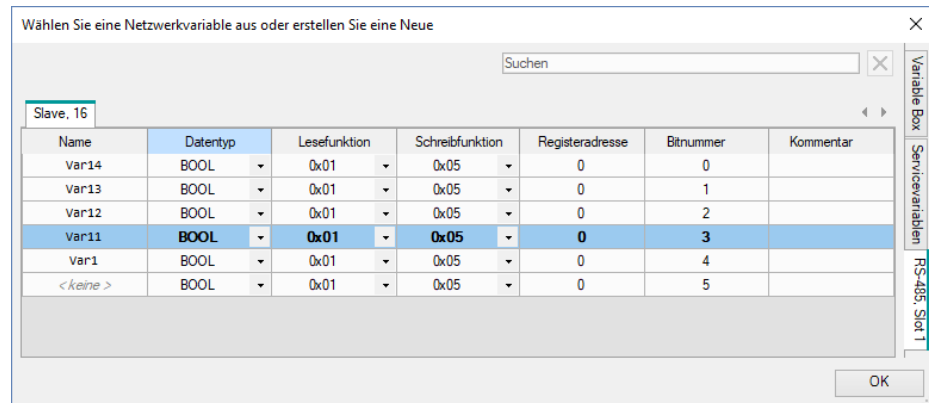


Abb. 5.8 Netzwerkvariablen für eine Master-Schnittstelle

Netzwerkvariablen und ihre Verknüpfungen werden auf dieselbe Weise gelöscht wie Standardvariablen (Abschn. 5.3).

Weitere Details zu den Netzwerkvariablen für eine Master-Schnittstelle finden Sie im Abschn. 4.3.3.1.

Wenn die Schnittstelle als ein Slave konfiguriert ist, werden alle Netzwerkvariablen, die vom Master abgefragt werden, in einer Liste angezeigt (Abb. 5.9). Weitere Details zu den Netzwerkvariablen für eine Slave-Schnittstelle finden Sie im Abschn. 4.3.3.3.

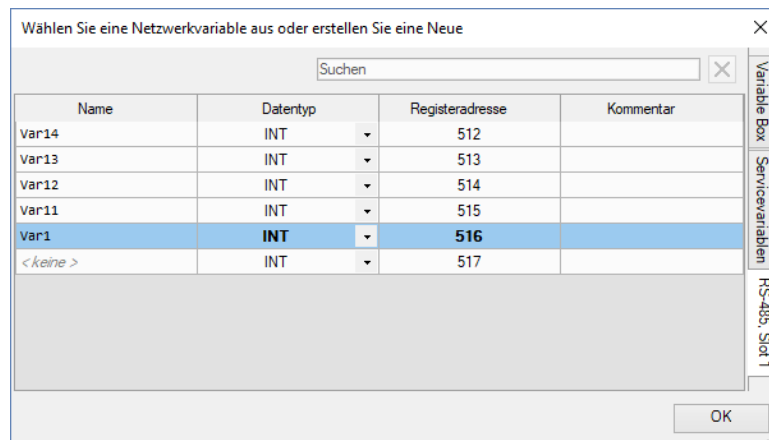




Abb. 5.9 Netzwerkvariablen für eine Slave-Schnittstelle

5.6 Variablenbaustein kopieren / einfügen

Die Variablenbausteine können kopiert und in ein anderes Projekt eingefügt werden.

Um einen Variablenbaustein zu kopieren, wählen Sie ihn im Arbeitsbereich aus und verwenden Sie das Symbol  in der Symbolleiste oder den Punkt **Kopieren** im Baustein-Kontextmenü.

Um einen Variablenbaustein in ein anderes Projekt einzufügen, öffnen Sie ihn in der zweiten ALP-Instanz und verwenden Sie das Symbol  in der Symbolleiste oder den Punkt **Einfügen** im Kontextmenü des Arbeitsbereichs.

Kopierregeln für alle mit dem Baustein verknüpften Variablen:

- Wenn die mit dem Baustein verknüpfte Variable im zweiten Projekt eindeutig ist, wird sie mit allen Eigenschaften zur Variablenliste hinzugefügt.

Variable

- Wenn das zweite Projekt eine identische Variable enthält, wird diese dem eingefügten Baustein zugeordnet. Der Variablen-tabelle werden keine neuen Variablen hinzugefügt.
- Wenn das zweite Projekt eine Variable mit demselben Namen, aber unterschiedlichen Parametern enthält, wird eine neue Variable erstellt. Um den Namenskonflikt zu lösen, sollte der Name einer der Variablen manuell geändert werden.
- Es ist nicht möglich, Variablen vom Typ REAL in ein Projekt für ein Zielgerät einzufügen, das den Datentyp REAL nicht unterstützt.
- Retain-Variablen (dauerhafte) können nicht in ein Projekt für ein Zielgerät kopiert werden, das sie nicht unterstützt.

Kopierregeln für Servicevariable:

- Servicevariablen können nicht in ein Projekt kopiert werden, der für ein Zielgerät ohne Echtzeituhr geschrieben ist.

Kopierregeln für Netzwerkvariable:

- Nur die Variablen einer Slave-Schnittstelle können in ein anderes Projekt kopiert werden und die Schnittstellen in beiden Projekten müssen die gleichen Slotnummern haben. Die Variablen der Master-Schnittstellen sollten manuell erstellt werden.
- Ein etwaiger Registerkonflikt muss manuell behoben werden.

6 Bibliothek

Wenn ein Projekt geöffnet ist, enthält das Panel Library Box folgende Bibliotheken:

- **Funktionen**
- **Funktionsbausteine**
- **Projektmakros**

Wählen Sie im unteren Teil des Panels ein Symbol aus, um den jeweiligen Inhalt anzuzeigen.

Die Bibliothek **Projektmakros** beinhaltet die Makros, die vom Benutzer erstellt, importiert oder aus der Online-Datenbank in das Projekt aufgenommen wurden.

Die Ansichtsoptionen können über die Symbole in der Symbolleiste des Panels geändert werden.

6.1 Funktionen

Die Bibliothek enthält folgende Funktionsgruppen:

- Logische Operatoren
- Mathematische Operatoren
- Vergleichsoperatoren
- Bitshift-Operatoren
- Bit-Operatoren

6.1.1 Logische Operatoren

- UND-Verknüpfung (AND)
- ODER-Verknüpfung (OR)
- Negation (NOT)
- Exklusives ODER (XOR)

Die logischen Operatoren können BOOL- oder INT-Variable verarbeiten.

Wenn die Eingangswerte INT sind, wird die Operation für jedes Bit separat durchgeführt und der Ausgang ist ebenfalls ein INT.

6.1.1.1 UND-Verknüpfung (AND)

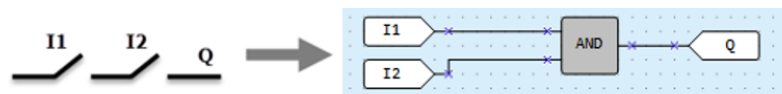


Abb. 6.1

Der Ausgang **Q** ist **Wahr**, wenn beide Eingänge **Wahr** sind. Die Funktion **UND** entspricht einer Reihenschaltung in einem elektrischen Schaltkreis.

Tabelle 6.1 Wahrheitstabelle

I1	I2	Q
0	0	0
0	1	0
1	0	0
1	1	1

Beispiel für eine bitweise Operation mit Ganzzahleingängen:

```

0101 (decimal 5)
AND 0011 (decimal 3)
= 0001 (decimal 1)

```

6.1.1.2 ODER-Verknüpfung (OR)

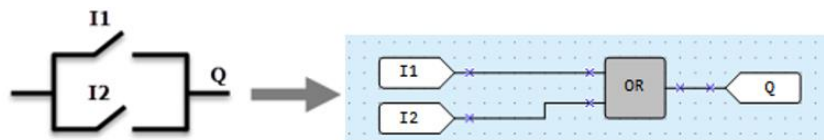


Abb. 6.2

Der Ausgang **Q** ist **Wahr**, wenn mindestens einer der Eingänge **Wahr** ist. Die Funktion **ODER** entspricht einer Parallelschaltung in einem elektrischen Schaltkreis.

Tabelle 6.2 Wahrheitstabelle

I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	1

Beispiel für eine bitweise Operation mit Ganzzahleingängen:

0101 (decimal 5)
 OR 0011 (decimal 3)
 = 0111 (decimal 1)

6.1.1.3 Negation (NOT)

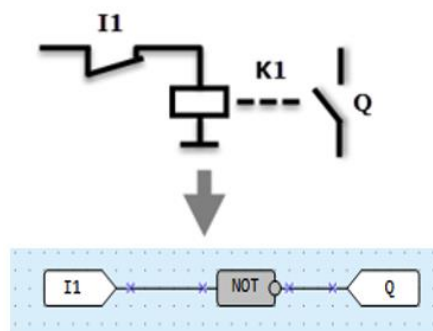


Abb. 6.3

Die Funktion **NOT** invertiert ein Signal. Der Ausgang **Q** ist **Wahr**, wenn der Eingang **Falsch** ist und umgekehrt.

Tabelle 6.3 Wahrheitstabelle

I1	Q
0	1
1	0

Beispiel für eine bitweise Operation mit Ganzzahleingängen:

NOT 0111 (decimal 7)
 = 1000 (decimal 8)

Das bitweise NOT oder Komplement ist eine unäre Operation, die für jedes Bit eine logische Negation durchführt und das Einerkomplement des gegebenen Binärwerts bildet.

6.1.1.4 Exklusives ODER (XOR)

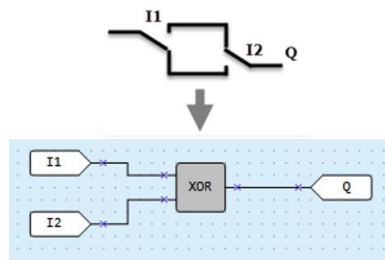


Abb. 6.4

Der Ausgang **Q** ist **Wahr**, wenn nur einer der Eingängen **Wahr** ist.

Tabelle 6.4 Wahrheitstabelle

I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	0

Beispiel für eine bitweise Operation mit Ganzzahleingängen:

0101 (decimal 5)
 OR 0011 (decimal 3)
 = 0110 (decimal 6)

6.1.2 Mathematische Operatoren

Es gibt verschiedene Operatoren für verschiedene Datentypen:

Tabelle 6.5

Operator	INT	REAL
Addition	ADD	fADD
Subtraktion	SUB	fSUB
Multiplikation	MUL	fMUL
Division	DIV	fDIV
Division mit Rest (Modulo)	MOD	-
Potenzfunktion	-	fPOW
Absoluter Wert	-	fABS

6.1.2.1 Addition (ADD, fADD)

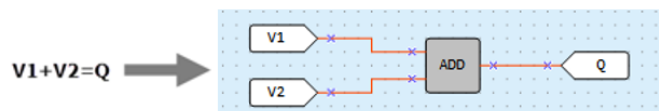


Abb. 6.5

Die Funktion **ADD** arbeitet mit INT-Variablen, die Funktion **fADD** arbeitet mit REAL-Variablen.

Der Ausgangswert **Q** ist die Summe der Eingangswerte.

Beispiel:

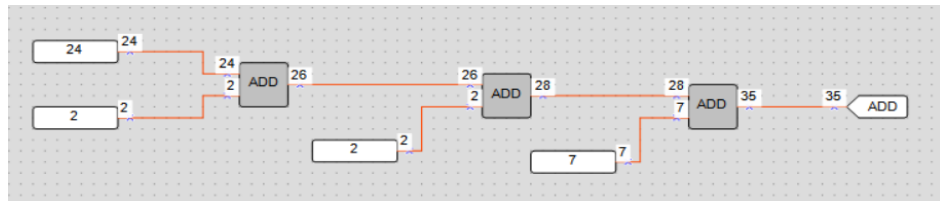


Abb. 6.6

Das Ergebnis darf 4294967295 (32 Bits) nicht überschreiten. Andernfalls werden die überschüssigen Bits abgeschnitten.

6.1.2.2 Subtraktion (SUB, fSUB)

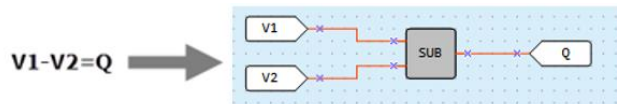


Abb. 6.7

Die Funktion **SUB** arbeitet mit den INT-Variablen, die Funktion **fSUB** arbeitet mit den REAL-Variablen.

Der Ausgangswert **Q** ist das Ergebnis der Subtraktion des Wertes **I2** vom Wert **I1**.

Beispiel 1:

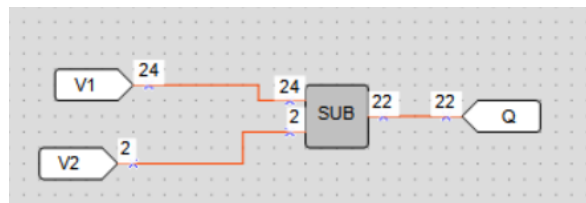


Abb. 6.8

Wenn der Wert **I1** kleiner ist als der Wert **I2**, wird das Ergebnis wie folgt berechnet:

$$Q = I1 + 0x100000000 - I2$$

$$0x100000000 = 4294967296$$

Beispiel 2:

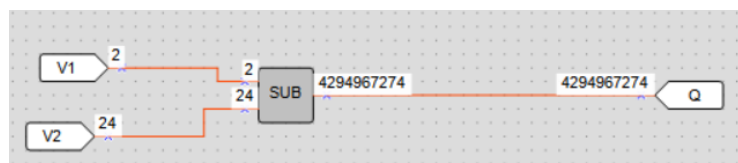


Abb. 6.9

6.1.2.3 Multiplikation (MUL, fMUL)

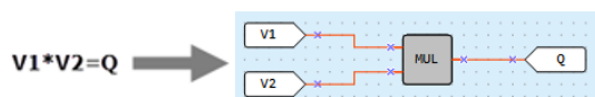


Abb. 6.10

Die Funktion **MUL** arbeitet mit den INT-Variablen, die Funktion **fMUL** arbeitet mit den REAL-Variablen.

Der Ausgangswert **Q** ist das Produkt der Eingangswerte.

Beispiel:

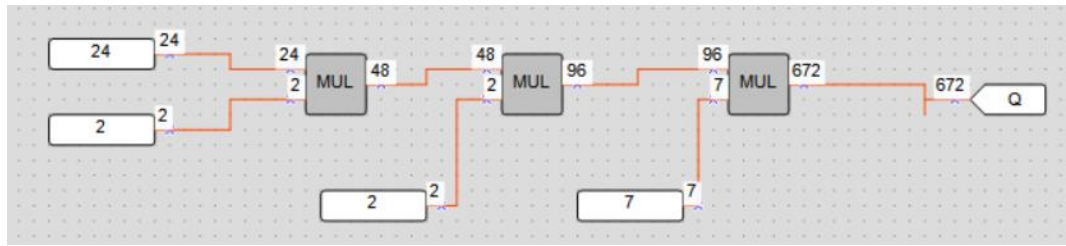


Abb. 6.11

Das Ergebnis darf 4294967295 (32 Bits) nicht überschreiten. Wenn dies auftritt, werden die zusätzlichen Bits abgeschnitten.

6.1.2.4 Division (DIV, fDIV)

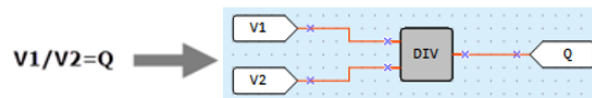


Abb. 6.12

Die Funktion **DIV** arbeitet mit den INT-Variablen, die Funktion **fDIV** arbeitet mit den REAL-Variablen.

Der Ausgangswert **Q** ist der Quotient aus der Division der Eingangswerte, wobei der Wert **I1** (Dividend) durch den Wert **I2** (Divisor) geteilt wird.

Wenn der Quotient kein INT ist, wird er auf einen INT abgerundet.

Im Falle einer Division durch 0 ist der Ausgangswert 0xFFFFFFFF.

6.1.2.5 Division mit Rest (MOD)

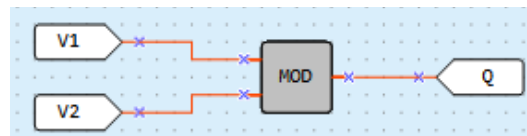


Abb. 6.13

Die Funktion **MOD** arbeitet mit den INT-Variablen. Das Ergebnis **Q** ist der Rest der Division der Eingangswerte.

$$Q = \left[\frac{V1}{V2} \right]$$

Beispiel:

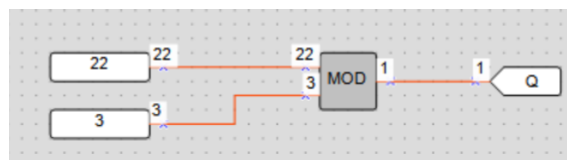


Abb. 6.14

6.1.2.6 REAL-Potenzfunktion (fPOW)

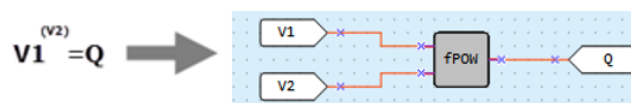


Abb. 6.15

Die Funktion **fPOW** arbeitet mit den REAL-Variablen.

Bibliothek

Das Ergebnis **Q** ist der Wert **I1**, potenziert um den Wert **I2**.

Beispiel:

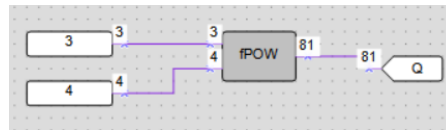


Abb. 6.16

6.1.2.7 REAL-Absolutwert-Funktion (fABS)

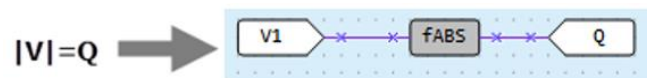


Abb. 6.17

Die Funktion **fABS** arbeitet mit den REAL-Variablen.

Das Ergebnis **Q** ist der absolute Wert des Eingangswertes.

$$Q = |V|$$

Beispiel:

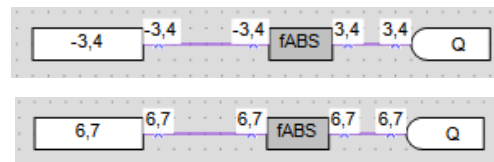


Abb. 6.18

6.1.3 Vergleichsoperatoren

Die Vergleichsoperatoren sind Funktionen, die ein bestimmtes Verhältnis zwischen zwei oder mehr Werten testen oder definieren.

6.1.3.1 Gleich (EQ)

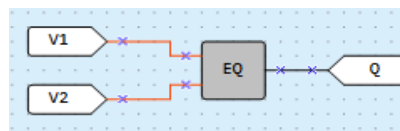


Abb. 6.19

Die Funktion **EQ** arbeitet mit den INT-Variablen.

Der Ausgangswert **Q** ist **Wahr**, wenn der Wert **I1** und der Wert **I2** gleich sind.

Tabelle 6.6 Wahrheitstabelle

I1 / I2	Q
I1 = I2	1
I1 > I2	0
I1 < I2	0

Beispiel:

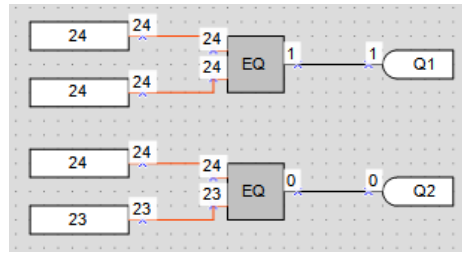


Abb. 6.20

6.1.3.2 Größer als (GT, fGT)

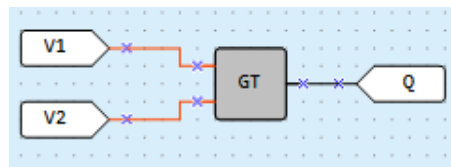


Abb. 6.21

Die Funktion **GT** arbeitet mit den INT-Variablen, die Funktion **fGT** arbeitet mit den REAL-Variablen.

Der Ausgangswert **Q** ist **Wahr**, wenn der Wert **I1** größer ist als der Wert **I2**.

Tabelle 6.7 Wahrheitstabelle

I1 / I2	Q
I1 = I2	0
I1 > I2	1
I1 < I2	0

Beispiel:

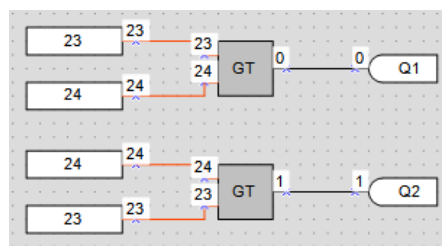


Abb. 6.22

6.1.3.3 Binäre Auswahl (SEL)

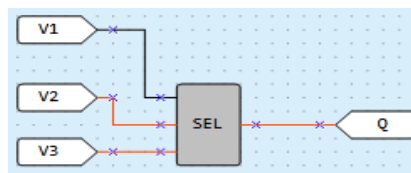


Abb. 6.23

Die Funktion **SEL** arbeitet mit den INT-Variablen, die Funktion **fSEL** arbeitet mit den REAL-Variablen.

Wenn **I1 = Falsch**, wird der Ausgangswert **Q** auf den Wert **I2** gesetzt, ansonsten auf den Wert **I3**.

Tabelle 6.8 Statustabelle

I1	Q
0	I2
1	I3

Beispiele:

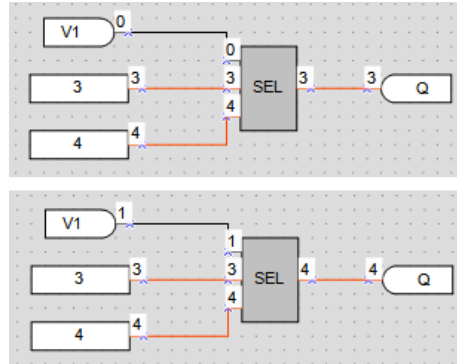


Abb. 6.24

6.1.4 Bitshift-Operatoren

Die Bitshift-Operatoren behandeln eine Variable als eine Reihe von Bits, die nach links oder rechts bewegt (verschoben) werden können.

6.1.4.1 Schieberegister links (SHL)

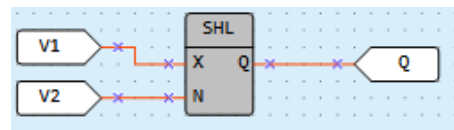


Abb. 6.25

Die Funktion **SHL** arbeitet mit den INT-Variablen. Sie wird verwendet, um alle Bits des Operanden **X** um **N** Bits nach links zu verschieben, freie Bits werden mit Nullen aufgefüllt. Das Ergebnis wird in den Ausgangswert **Q** geschrieben.

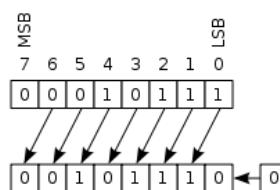


Abb. 6.26

Beispiel: Linksverschiebung der Zahl 38 (dezimal) = 00100110 (binär) um 2 Bits

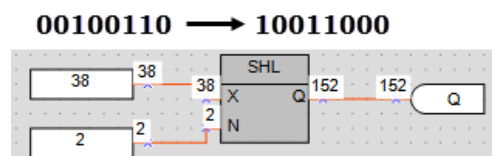


Abb. 6.27

6.1.4.2 Schieberegister rechts (SHR)

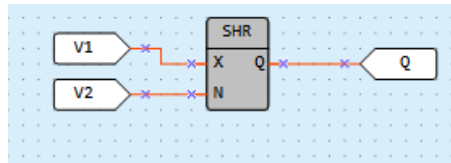


Abb. 6.28

Die Funktion **SHR** arbeitet mit den INT-Variablen. Sie wird verwendet, um alle Bits des Operanden **X** um **N** Bits nach rechts zu verschieben, freie Bits werden mit Nullen aufgefüllt. Das Ergebnis wird in den Ausgangswert **Q** geschrieben.

Beispiel: Rechtsverschiebung der Zahl 152 (dezimal) = 10011000 (binär) um 2 Bits

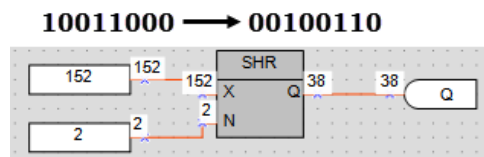


Abb. 6.29

6.1.5 Bit-Operatoren

Ein Bit-Operator behandelt einen Wert als eine Reihe von Bits, um Operationen an einem oder mehreren einzelnen Bits eines Operanden auszuführen.

6.1.5.1 Einzelnes Bit auslesen (EXTRACT)

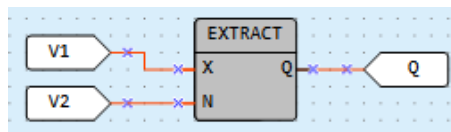


Abb. 6.30

Der Ausgangswert **Q** (BOOL) der Funktion **EXTRACT** ist der Wert des **N**-ten Bits (INT) im Operanden **X** (INT). Die Bitzählung beginnt mit Null.

Beispiel: Auslesen des 5. Bits aus der Zahl 81 (dezimal) = 1010001 (binär):

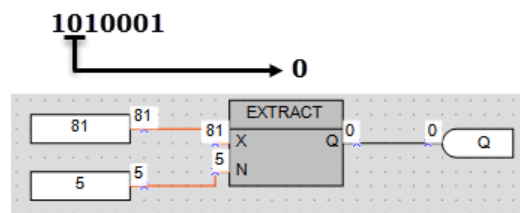


Abb. 6.31

6.1.5.2 Einzelnes Bit setzen (PUTBIT)

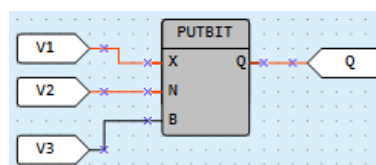


Abb. 6.32

Bibliothek

Der Ausgangswert **Q** (INT) ist der Wert des Operanden **X** (INT), wobei das **N**-te Bit (INT) auf den Eingangswert **B** (BOOL) gesetzt wird. Die Bitzählung beginnt mit Null.

Beispiel: Überschreiben des 4. Bits mit einer 1 in der Zahl 38 (dezimal) = 100110 (binär):

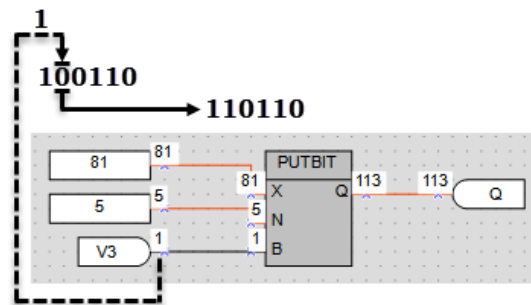


Abb. 6.33

6.1.5.3 Decoder (DC32)

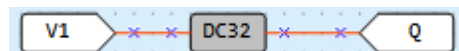


Abb. 6.34

Der Decoder konvertiert einen Binärcode am Eingang zu einem Positionscodex am Ausgang.

Die Dekodierung erfolgt bitweise durch die logische Verknüpfung UND mit dem Operanden 0x1F (11111b).

Tabelle 6.9 Wahrheitstabelle

Binärer Code					Positionscodex								
5	4	3	2	1	32	31		6	5	4	3	2	1
0	0	0	0	0	0	0		0	0	0	0	0	1
0	0	0	0	1	0	0		0	0	0	0	1	0
0	0	0	1	0	0	0		0	0	0	1	0	0
0	0	0	1	1	0	0		0	0	1	0	0	0
0	0	1	0	0	0	0	...	0	1	0	0	0	0
...									
1	1	1	0	1	0	0		0	0	0	0	0	0
1	1	1	1	0	0	1		0	0	0	0	0	0
1	1	1	1	1	1	0		0	0	0	0	0	0

Beispiel:

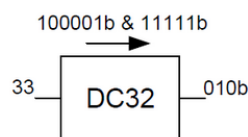


Abb. 6.35

6.1.5.4 Encoder (CD32)



Abb. 6.36

Bibliothek

Der Encoder wandelt einen Positionscode am Eingang in einen Binärcode am Ausgang um.

Wenn im Positionscode mehr als ein "1"-er-Bit vorhanden ist, arbeitet der Encoder nur mit dem höchstwertigen "1"-er-Bit.

Die Wahrheitstabelle finden Sie in der Tabelle 6.9.

6.2 Funktionsbausteine

Die Bibliothek enthält folgende FB-Gruppen:

- Trigger
- Timer
- Generatoren
- Zähler
- Steuerung

6.2.1 Trigger

- RS-Trigger mit Reset-Priorität (RS)
- SR-Trigger mit Set-Priorität (SR)
- Steigende Flanke (RTRIG)
- Fallende Flanke (FTRIG)
- D-Trigger (DTRIG)

6.2.1.1 RS-Trigger mit Reset-Priorität (RS)

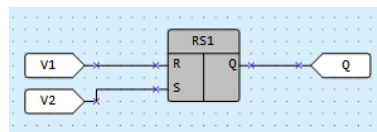


Abb. 6.37

Der Ausgang **Q** ist **Wahr** mit einer steigenden Flanke am Eingang **S** (Set) und **Falsch** mit steigender Flanke am Eingang **R** (Reset). Der Eingang **R** hat höhere Priorität.

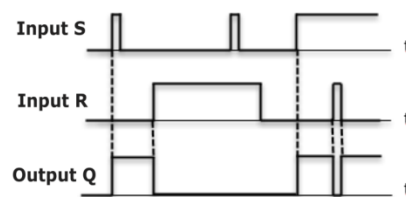


Abb. 6.38

6.2.1.2 SR-Trigger mit Set-Priorität (SR)

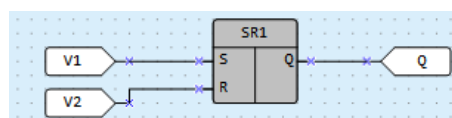


Abb. 6.39

Der Ausgang **Q** ist **Wahr** mit einer steigenden Flanke am Eingang **S** (Set) und **Falsch** mit steigender Flanke am Eingang **R** (Reset). Der Eingang **S** hat höhere Priorität.

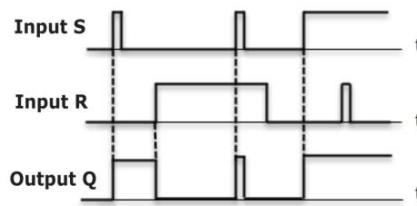


Abb. 6.40

6.2.1.3 Steigende Flanke (RTRIG)

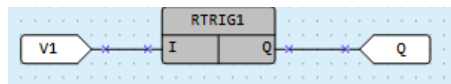


Abb. 6.41

Detektor für eine steigende Flanke

Der Ausgang **Q** bleibt **Falsch** bis zu einer steigenden Flanke am Eingang **I**. Sobald der Eingang **I Wahr** wird, wird der Ausgang **Q** auch **Wahr** und bleibt bis zum Ende des Programmzyklus erhalten.

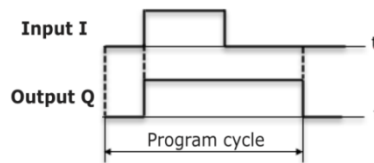


Abb. 6.42

6.2.1.4 Fallende Flanke (FTRIG)

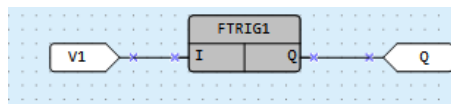


Abb. 6.43

Detektor für eine fallende Flanke

Der Ausgang **Q** bleibt **Falsch** bis zu einer fallenden Flanke am Eingang **I**. Sobald der Eingang **I Falsch** wird, wird der Ausgang **Wahr** und bleibt bis zum Ende des Programmzyklus erhalten.

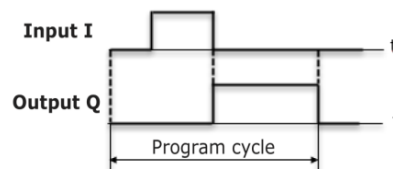


Abb. 6.44

6.2.1.5 D-Trigger (DTRIG)

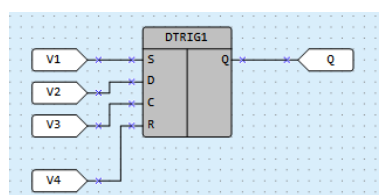


Abb. 6.45

Der D-Trigger erzeugt einen Impuls am Ausgang **Q** mit einer Impulsdauer, die am Eingang **D** festgelegt und mit der Taktfrequenz am Eingang **C** synchronisiert wird.

Wenn der Eingang **D Wahr** ist, wird der Ausgang **Q** bei steigender Flanke der Taktfrequenz am Eingang **C** auch **Wahr**.

Wenn der Eingang **D Falsch** ist, wird der Ausgang **Q** bei steigender Flanke der Taktfrequenz am Eingang **C** auch **Falsch**.

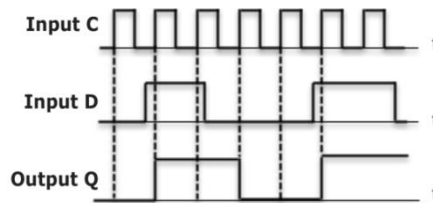


Abb. 6.46

Der Ausgang **Q** kann bei steigender Flanke am Eingang **S** (Set) zwangsweise auf **Wahr** gesetzt werden, und bei steigender Flanke am Eingang **R** (Reset) zwangsweise auf **Falsch** zurückgesetzt werden, unabhängig von den Zuständen der Eingänge **C** und **D**. Der Eingang **R** hat höhere Priorität.

6.2.2 Timer

- Impuls (TP)
- Einschaltverzögerung (TON)
- Ausschaltverzögerung (TOF)
- Schaltuhr (CLOCK)
- Wochenschaltuhr (CLOCKW)

6.2.2.1 Impuls (TP)

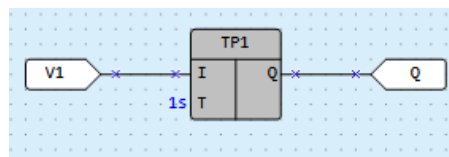


Abb. 6.47

Der Baustein **TP** wird verwendet, um einen Ausgangsimpuls mit einer vorgegebenen Impulsdauer zu erzeugen.

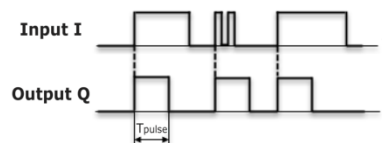


Abb. 6.48

Der Ausgang **Q** wird bei steigender Flanke am Eingang **I** für die am Eingang **T** festgelegte Zeitdauer auf **Wahr** gesetzt. Der Ausgang **Q** bleibt während dieser Zeit **Wahr**, unabhängig von der Signaländerung am Eingang **I**. Mit dem Impulsende wird der Ausgang **Q** auf **Falsch** zurückgesetzt.

Die Impulsdauer und die Zeiteinheit können in der Property Box eingestellt werden.

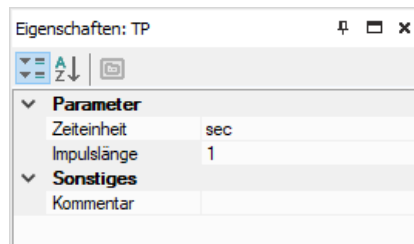


Abb. 6.49

Zeitbereich: 0...4147200000 ms oder 48 Tage.

6.2.2.2 Einschaltverzögerung (TON)

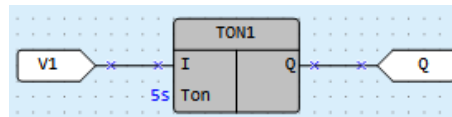


Abb. 6.50

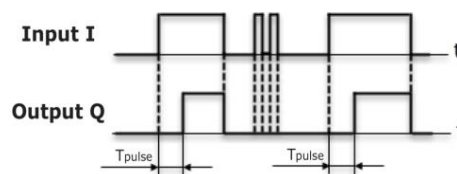


Abb. 6.51

Der Ausgang **Q = Falsch**, wenn der Eingang **I = Falsch**. Bei steigender Flanke beginnt die am Eingang T_{ON} angegebene Verzögerungszeit zu laufen. Wenn die Zeit T_{ON} verstrichen ist, wird der Ausgang **Q = Wahr** und bleibt bis zu einer fallenden Flanke am Eingang **I** erhalten. Eingangsänderungen, die kürzer als T_{ON} sind, werden ignoriert.

Die Verzögerungszeit und die Zeiteinheit können in der Property Box eingestellt werden.

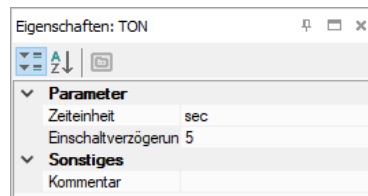


Abb. 6.52

Zeitbereich: 0...4147200000 ms oder 48 Tage.

6.2.2.3 Ausschaltverzögerung (TOF)

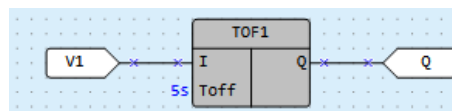


Abb. 6.53

Der Ausgang **Q = Falsch**, wenn der Eingang **I = Wahr**. Bei einer fallenden Flanke beginnt die am Eingang T_{OFF} angegebene Verzögerungszeit zu laufen. Wenn die Zeit T_{OFF} verstrichen ist, wird der Ausgang **Q Falsch** und bleibt erhalten bis zu einer steigenden Flanke am Eingang **I**. Eingangsänderungen, die kürzer als T_{OFF} sind, werden ignoriert.

Die Verzögerungszeit und die Zeiteinheit können in der Property Box eingestellt werden.

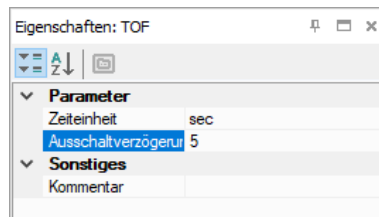


Abb. 6.54

Zeitbereich: 0...4147200000 ms oder 48 Tage.

6.2.2.4 Schaltuhr (CLOCK)

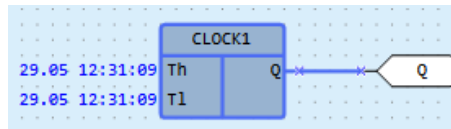


Abb. 6.55

Der Baustein **CLOCK** ist ein Intervall-Timer (Zeitschaltuhr), der von einer Echtzeituhr gesteuert wird.

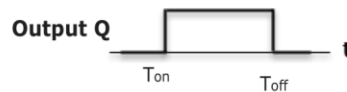


Abb. 6.56

Die Zeiten T_H und T_L können in der Property Box festgelegt werden.

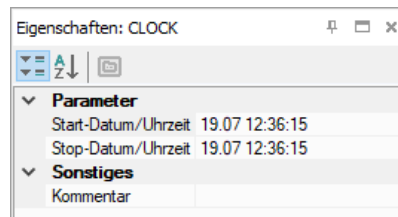


Abb. 6.57

Zeitbereich: von 0,00 Sekunden bis 24 Stunden.

Wenn $T_H < T_L$ ist, ist der Zustand des Ausgangs **Q** wie folgt:

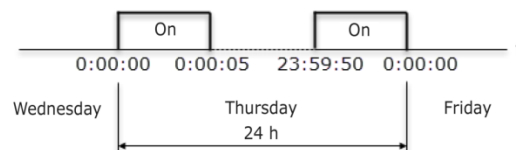


Abb. 6.58

6.2.2.5 Wochenschaltuhr (CLOCKW)

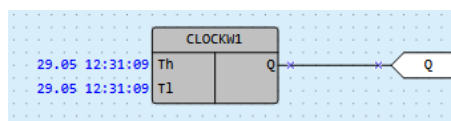


Abb. 6.59

Der Baustein **CLOCKW** ist ein Intervall-Timer (Zeitschaltuhr) mit dem Parameter **Wochentag**, der von einer Echtzeituhr gesteuert wird.

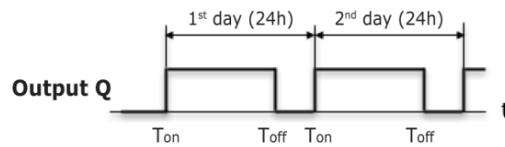


Abb. 6.60

Die Zeiten T_H und T_L können in der Property Box festgelegt werden.

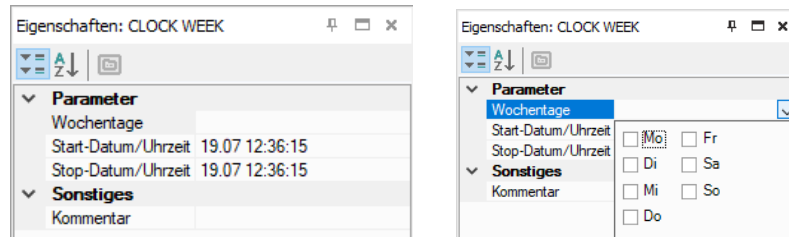


Abb. 6.61

Zeitbereich: von 0,00 Sekunden bis 24 Stunden.

6.2.3 Generatoren

- Impulsgenerator (BLINK)

6.2.3.1 Impulsgenerator (BLINK)

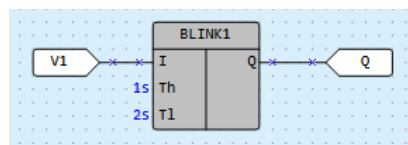


Abb. 6.62

Wenn der Eingang **I** auf **Wahr** gesetzt ist, erzeugt der Baustein **BLINK** eine Rechteckwelle am Ausgang **Q** mit einer Periode von $T_H + T_L$, beginnend mit einem Intervall der Dauer T_L , gefolgt von einem Impuls der Dauer T_H . Dies wird so lange fortgeführt, bis der Eingang **I** **Falsch** ist.

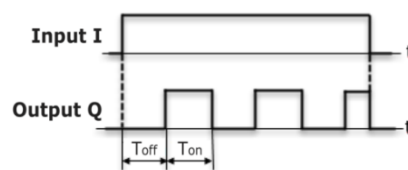


Abb. 6.63

Die Zeiten T_H und T_L und die Zeiteinheiten können in der Property Box festgelegt werden.

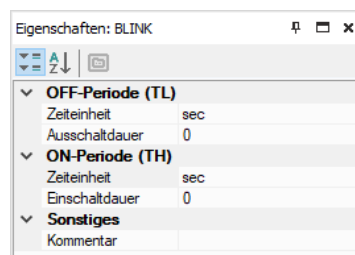


Abb. 6.64

Zeitbereich: 0...4233600000 Millisekunden oder 49 Tage.

6.2.4 Zähler

- Schwellenwertzähler mit Auto-Reset (CT)
- Universalzähler (CTN)
- Schwellenwertzähler (CTU)

6.2.4.1 Schwellenwertzähler mit Auto-Reset (CT)

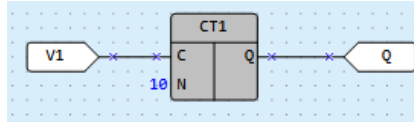


Abb. 6.65

Der Ausgang **Q** ist vom Typ BOOL. Wenn die Anzahl der am Eingang **C** gezählten Impulse den am Eingang **N** angegebenen Schwellenwert (**Einstellung**) erreicht, wird der Ausgang **Q** auf **Wahr** gesetzt und bleibt so für einen Programmzyklus.

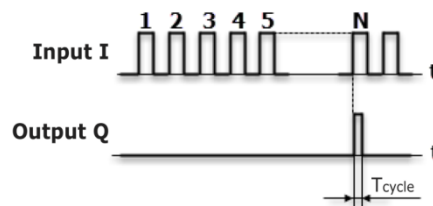


Abb. 6.66

Die Parameter **Einstellung** und **Zustand speichern** können in der Property Box festgelegt werden.

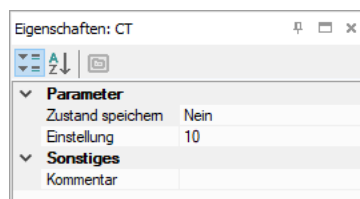


Abb. 6.67

Schwellenwertbereich: 0...65535.

Wenn **Zustand speichern = Ja**, dann wird der Zustand des Zählers im nichtflüchtigen Speicher gespeichert.

6.2.4.2 Universalzähler (CTN)

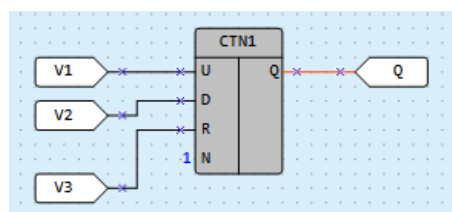


Abb. 6.68

Der Ausgang **Q** ist vom Typ INT. Eine steigende Flanke am Eingang **U** erhöht den Wert am Ausgang **Q** um 1. Eine steigende Flanke am Eingang **D** vermindert den Wert am Ausgang **Q** um 1.

Wenn der Eingang **R = Wahr**, wird der Ausgang **Q** auf den voreingestellten Wert (**Einstellung**) am Eingang **N** gesetzt.

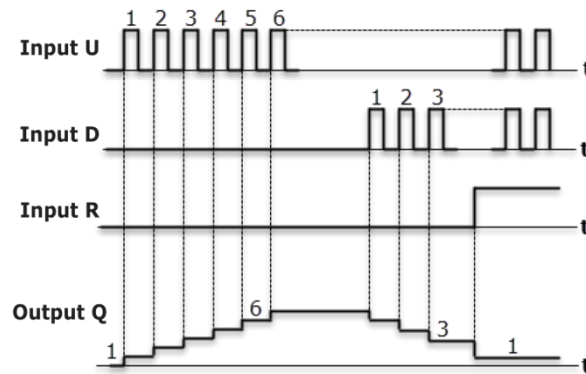


Abb. 6.69

Der Eingang **U** hat eine höhere Priorität als der Eingang **D**.

Die Parameter **Einstellung** und **Zustand speichern** können in der Property Box festgelegt werden.

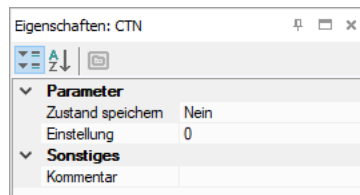


Abb. 6.70

Einstellungsbereich: 0...65535.

Wenn **Zustand speichern** = **Ja**, dann wird der Zustand des Zählers im nichtflüchtigen Speicher gespeichert.

6.2.4.3 Schwellenwertzähler (CTU)

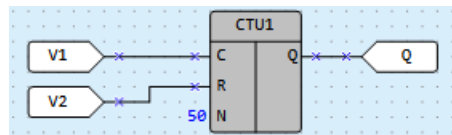


Abb. 6.71

Der Ausgang **Q** ist vom Typ BOOL. Wenn die Anzahl der am Eingang **C** gezählten Impulse den am Eingang **N** angegebenen Schwellenwert (**Einstellung**) erreicht, wird der Ausgang **Q** auf **Wahr** gesetzt und behält diesen Zustand bis zu einer ansteigenden Flanke am Eingang **R**. Der Eingang **R** hat eine höhere Priorität als der Eingang **C**.

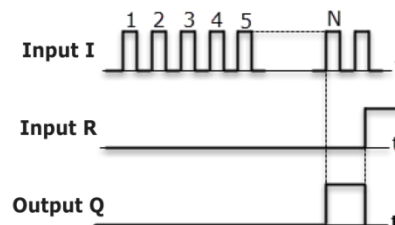


Abb. 6.72

Der Parameter **Einstellung** kann in der Property Box festgelegt werden.

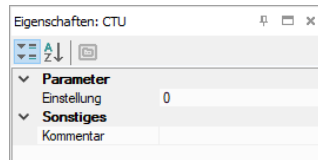


Abb. 6.73

Schwellenbereich: 0...65535.

6.2.5 Analog

- PID-Regler (PID)

6.2.5.1 PID-Regler (PID)

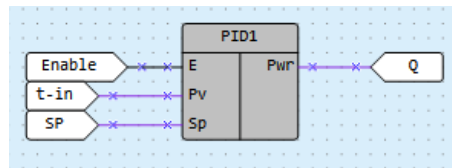


Abb. 6.74

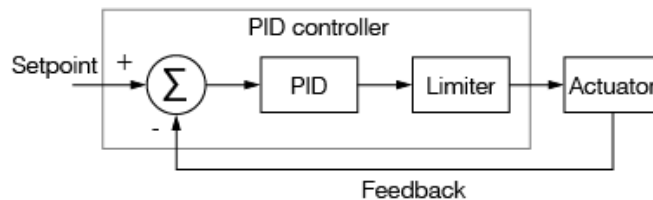


Abb. 6.75

Der Funktionsbaustein **PID** wird für die Implementierung der Proportional-Integral-Derivative-Regelung verwendet.

Tabelle 6.10 Ein-/Ausgänge des PID-Bausteins

Name	Typ	E/A	Beschreibung	Werte
E	BOOL	E	Regelung aktivieren (0 = Off, 1 = On). Wenn deaktiviert, nimmt der Parameter Pwr den Wert des Parameters Sicherer Ausgangszustand an.	0/1
Pv	REAL	E	Prozesswert	
Sp	REAL	E	Sollwert	
Pwr	REAL	A	Ausgangsleistung, %	0...100

Tabelle 6.11 Parameter des PID-Bausteins

Name	Typ	Beschreibung	Werte	Access		
				Property Box	Write ToFB	Read FromFB
Regelungsmodus	BOOL	0 – Heizung 1 – Kühlung	0/1	✓	✓	
Sicherer Ausgangszustand	REAL	Ausgangswert bei deaktivierter Steuerung, %	0...100	✓	✓	
Kp	REAL	Verstärkung, Multiplikationsfaktor für Proportionalregelung	0...100	✓	✓	
Ti (s)	REAL	Nachstellzeit, Zeitkonstante für Integralregelung in Sekunden	-3,402823e+38... 3,402823e+38	✓	✓	

Name	Typ	Beschreibung	Werte	Access		
				Property Box	Write ToFB	Read FromFB
Td (s)	REAL	Vorhaltezeit, Zeitkonstante für Differentialregelung in Sekunden	-3,402823e+38... 3,402823e+38	✓	✓	
Ausgang max.	REAL	Ausgangsobergrenze,% (Standard 80)	0...100	✓	✓	
Ausgang min.	REAL	Ausgangsuntergrenze,% (Standard 20)	0...100	✓	✓	
Start AT	BOOL	0 – stoppt Auto-Tuning 1 – startet Auto-Tuning	0/1		✓	
AT beendet	BOOL	Marker: 0 – Auto-Tuning gestoppt 1 – Auto-Tuning gestartet	0/1			✓
Kp berechnet	REAL	Berechnete Verstärkung	0...100			✓
Ti berechnet	REAL	Berechnete Nachstellzeit	-3,402823e+38... 3,402823e+38			✓
Td berechnet	REAL	Berechnete Vorhaltezeit	-3,402823e+38... 3,402823e+38			✓

Die Abstimmung (Tuning) eines Regelkreises beinhaltet die Einstellung seiner Regelparameter (**Kp**, **Ti**, **Td**) auf die optimalen Werte für das gewünschte Regelverhalten.

Programmierbare Abstimmung

Eine programmierbare Regelkreis-Abstimmung kann mit den Bausteinen **WriteToFB** ^W und **ReadFromFB** ^R durchgeführt werden (Abschn. 7.7).

Zum Schreiben der Parameter verwenden Sie den Baustein WriteToFB (Abb. 6.76) oder die Property Box (Abb. 6.77).

Zum Lesen der Parameter verwenden Sie den Baustein **ReadFromFB** (Abb. 6.76).

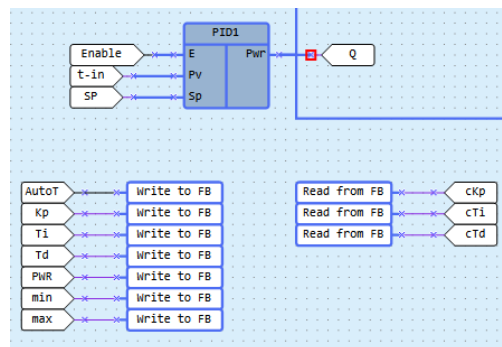


Abb. 6.76

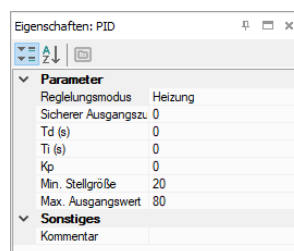


Abb. 6.77

Auto-Tuning

Um eine automatische Abstimmung (Auto-Tuning) zu verwenden, fügen Sie dem Schaltprogramm den Baustein **WriteToFB** hinzu und setzen Sie die Verknüpfung zum Parameter **Start AT** des PID-Bausteins.

Um das Auto-Tuning zu starten, aktivieren Sie die Regelung (**E = 1**) und setzen Sie den Parameter **Start AT = 1**.

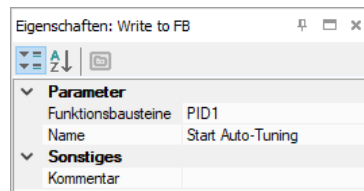


Abb. 6.78

Nach dem Abschluss des Auto-Tunings werden die neuen Werte der Parameter **Kp**, **Ti** und **Td** berechnet und der Merker **AT beendet** wird auf **1** gesetzt.

Wenn **Start AT = 0**, wird Merker **AT beendet = 0** ebenfalls.

Wenn Sie den Parameter **Start AT** vor dem Abschluss des Auto-Tunings auf **0** setzen, wird das Auto-Tuning gestoppt, der Merker **AT beendet** wird auf **0** gesetzt und keine neuen Koeffizienten werden berechnet.

Während des Auto-Tunings wird ein durch die Parameter **Ausgang max.** und **Ausgang min.** begrenztes Signal an den Ausgang **Pwr** gelegt.

Hinweis: Wenn der maximale Verstärkung nicht ausreicht, um den Sollwert zu erreichen, kann das Auto-Tuning nicht abgeschlossen werden und wird fortgesetzt, bis es mit **Start AT = 0** gestoppt wird.

6.3 Projektmakros

Ein Makro ist ein Funktionsbaustein, der wie ein Hauptprogramm in einem eigenen Arbeitsbereich erstellt ist.

So erstellen Sie ein neues Makro:

- wählen Sie den Menüpunkt **Datei > Neues Makro** (Abschn. 6.3.3) oder
- wählen Sie Bausteine im Hauptarbeitsbereich aus und wählen Sie den Punkt **Neues Makro** im Arbeitsbereichs-Kontextmenü (Abschn. 6.3.4).

Um das Makro im Projekt zu verwenden, ziehen Sie es per Drag-and-Drop aus der Library Box in den Hauptarbeitsbereich.

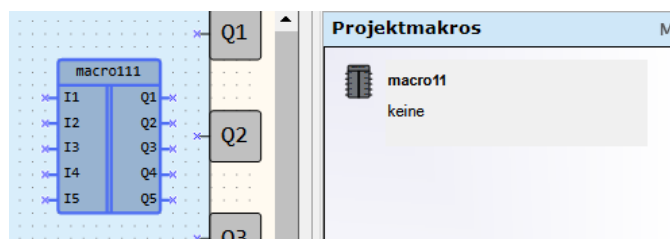


Abb. 6.79

Um das Projektmakro zur Bearbeitung in einem separaten Arbeitsbereich zu öffnen, wählen Sie es im Arbeitsbereich oder in der Bibliothek aus und wählen Sie den Punkt **Makro bearbeiten** im Makro-Kontextmenü.

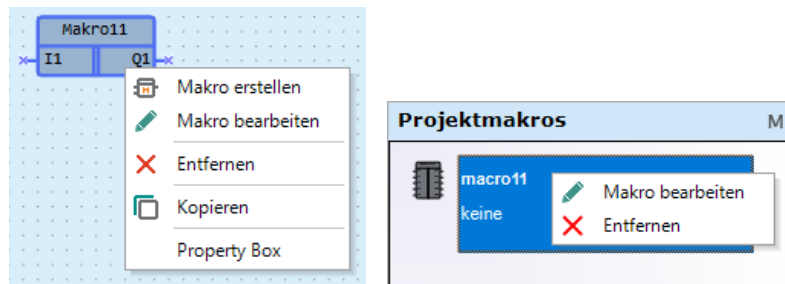


Abb. 6.80

Mit dem Menüpunkt **Datei > Makro speichern unter...** kann das Makro unter einem anderen Namen im Projekt gespeichert werden. Das gespeicherte Makro ist nur in diesem Projekt verfügbar.

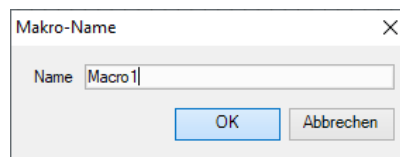


Abb. 6.81

Wenn das Makro in anderen Projekten verwendet werden soll, muss es als Datei gespeichert, d. h. exportiert und dann aus dieser Datei in ein anderes Projekt importiert werden (Abschn. 6.3.1).

Wenn ein im Projekt verwendetes Makro geändert wird, wird es im Programm rot angezeigt und der Benutzer wird aufgefordert, das Makro zu aktualisieren (Abschn. 6.3.5).

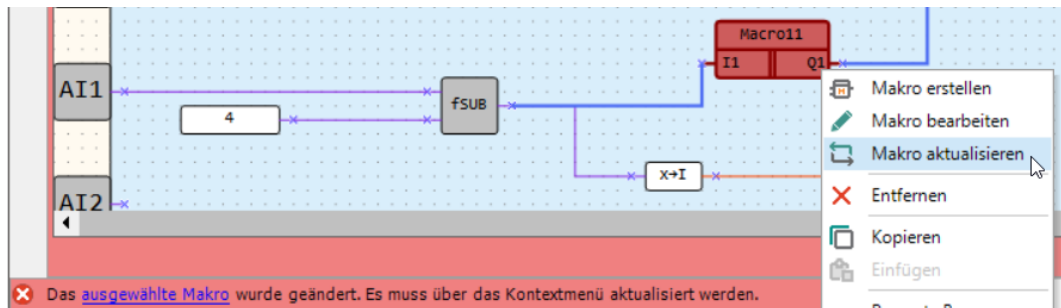


Abb. 6.82 Aktualisierungsaufforderung des Makros

6.3.1 Makro exportieren, importieren, herunterladen

Ein Makro kann in eine Datei exportiert werden, wenn der Makro-Arbeitsbereich aktiv ist. Öffnen Sie das Makro zur Bearbeitung, wählen Sie den Menüpunkt **Datei > Makro exportieren**, geben Sie den Dateinamen und den Pfad an und bestätigen Sie mit **OK**. Die Makrodatei hat die Erweiterung ***.tpl**.

Ein Makro kann aus der Datei in die Bibliothek importiert werden, wenn das Hauptprogramm aktiv ist. Wählen Sie den Menüpunkt **Datei > Makro importieren**, wählen Sie die Datei im geöffneten Dialog aus und bestätigen Sie mit **OK**. Das Makro wird der Bibliothek im Bereich **Makros** hinzugefügt.

Zum Herunterladen von Makros aus der **Online-Datenbank** ist eine Internetverbindung erforderlich. Wählen Sie den Menüpunkt **Datei > Komponentenmanager** (der Hauptarbeitsbereich muss aktiv sein), um den **Komponentenmanager** in einem separaten Fenster zu öffnen (Abschn. 2.9).

6.3.2 FB in Makro

Wird im Makro ein Funktionsbaustein verwendet, kann der Anwender festlegen, ob die FB-Parameter als Parameter des Makros im Hauptprogramm zur Verfügung stehen.

Wenn der Parameter **Verwenden in Makro** auf **Ja** gesetzt ist, werden die Parameter des FB zu Parametern des Makros, und dem Makro wird eine neue Option **Parameter des Makros** hinzugefügt. Mit dieser Option kann der Benutzer den Namen für jeden FB-Parameter im Makro angeben, um den im Hauptprogramm zu verwenden.

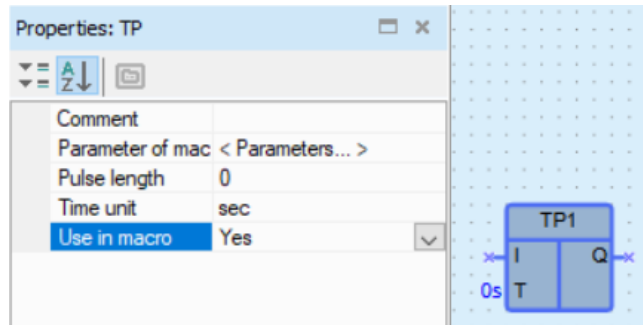


Abb. 6.83

6.3.3 Neues Makro über Hauptmenü

Wählen Sie im Hauptmenü den Punkt **Datei > Neues Makro**. Geben Sie im geöffneten Dialog die Anzahl der Ein- und Ausgänge an und bestätigen Sie mit **OK**. Das neue leere Makro wird in einem separaten Arbeitsbereich geöffnet.

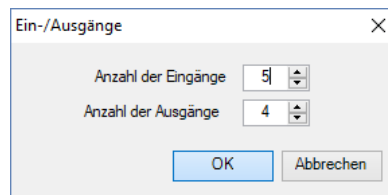


Abb. 6.84

Die Anzahl der Ein- und Ausgänge kann jederzeit über das Kontextmenü des Arbeitsbereichs geändert werden.

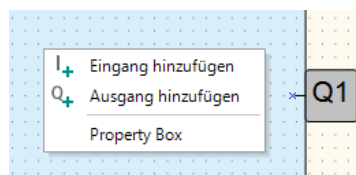


Abb. 6.85

Um einen Eingang oder einen Ausgang zu entfernen, wählen Sie **Löschen** in seinem Kontextmenü.

Der Datentyp für jeden Ein- und Ausgang kann in der Property Box ausgewählt werden.

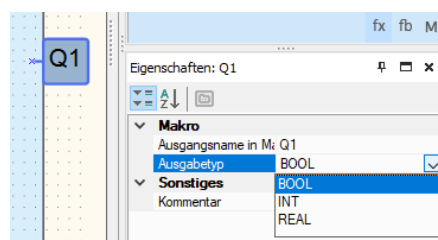


Abb. 6.86

Geben Sie einen Namen, eine Beschreibung und eine Gruppe für das Makro in der Property Box an.

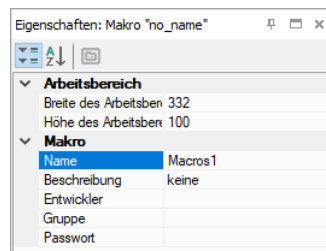


Abb. 6.87

Der Name wird in der Registerkarte des Arbeitsbereichs angezeigt. Es ist der Name des Makros im Projekt.

Der Text im Parameter **Beschreibung** wird in einem QuickInfo angezeigt, wenn sich der Mauszeiger über dem Makro im Arbeitsbereich befindet.

Der Name im Parameter **Gruppe** wird in der Bibliothek verwendet. Ist der Gruppenname leer, wird das Makro der Gruppe **Andere** zugeteilt.

Wenn Sie das Kennwort für das Makro festlegen, wird es jedes Mal abgefragt, wenn der Menüpunkt **Makro bearbeiten** ausgewählt wird.

6.3.4 Neues Makro über Kontextmenü

Sie können ein Makro erstellen, indem Sie im Arbeitsbereich ein Auswahlrechteck zeichnen und im Kontextmenü des Arbeitsbereichs den Punkt **Neues Makro** wählen. Alle ausgewählten Bausteine werden in einen neuen Makro-Baustein verschoben, der diese im Arbeitsbereich ersetzt. Alle externen Verbindungslinien werden beibehalten.

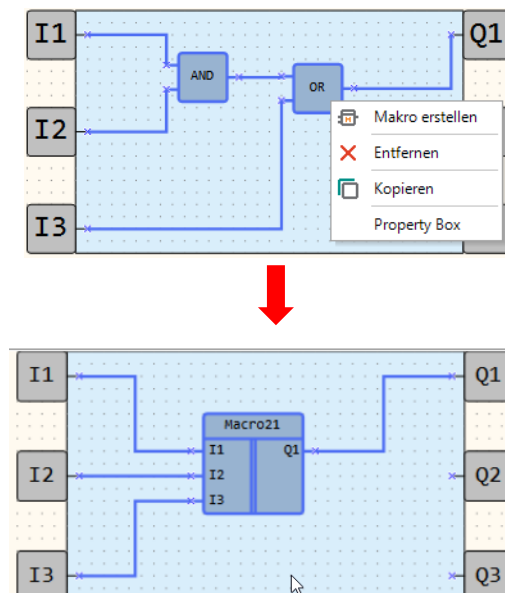


Abb. 6.88

Es gibt einige spezifische Aspekte beim Erstellen von Makros über das Kontextmenü:

- Die Anzahl der Ein- und Ausgänge des Makros entspricht der Anzahl der angeschlossenen Ein- und Ausgangsverbindungen im ausgewählten Bereich. Wenn die Bausteine ohne Verbindungen ausgewählt sind, wird das Makro mit einem Eingang und einem Ausgang erstellt.
- Wenn ein Standardvariablen-Baustein ausgewählt ist, wird die Variable unter demselben Namen in das Makro kopiert.

Hinweis: Trotz des gleichen Namens sind die Variablen im Makro und im Hauptprogramm unterschiedlich, es besteht kein Konflikt zwischen ihnen.

- Wenn alle Bausteine der Variablen ausgewählt sind und keine anderen Verknüpfungen im Programm vorhanden sind, wird die Variable in das Makro verschoben.
- Wenn die ausgewählte Variable außerhalb des ausgewählten Bereichs verwendet wird (Bausteine oder andere Verknüpfungen enthält), wird sie unter demselben Namen in das Makro kopiert und das Original verbleibt im Arbeitsbereich.
- Wenn nur ein Baustein der Eingangs- oder Ausgangsvariablen ausgewählt ist, wird die Variable unter demselben Namen in das Makro kopiert und das Original bleibt im Arbeitsbereich.
- Wenn das Makro über das Kontextmenü erstellt wird, werden die folgenden Bausteine nicht darin enthalten sein:
 - Geräteein- und ausgänge
 - Servicevariablen
 - Netzwerkvariablen
 - PID-Regler
- Wenn die oben genannten Bausteine ausgewählt sind, verbleiben sie im Hauptarbeitsbereich und werden mit den entsprechenden E/A-Punkten des Makros verbunden.
- Wenn Bausteine **WriteToFB** / **ReadFromFB** (Abschn. 7.7) mit dem ausgewählten FB verknüpft sind, werden sie auch in das Makro aufgenommen, auch wenn sie nicht ausgewählt sind. Wenn diese Lese-/Schreibblöcke ausgewählt sind, jedoch nicht der verknüpfte FB, werden sie nicht in das Makro aufgenommen.

6.3.5 Makro aktualisieren

Wenn das im Programm verwendete Makro geändert wurde (Name, Typ, Anzahl der E/A-Punkte oder der Parameter **Verwenden in Makro** eines Bausteins), wird es im Programm rot angezeigt und der Benutzer zum Aktualisieren des Makros aufgefordert. Ein Makro gilt als geändert, nachdem die im Makro-Editor vorgenommenen Änderungen gespeichert sind.

Verwenden Sie das Kontextmenü, um das Makro zu aktualisieren.

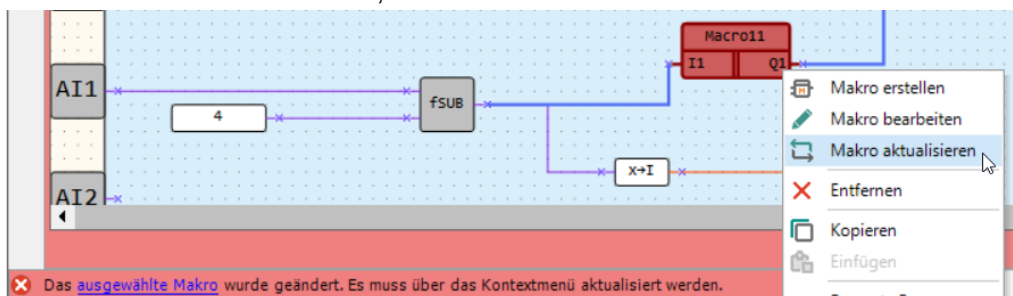


Abb. 6.89

Sobald das Makro aktualisiert ist, wird das nächste geänderte Makro zur Aktualisierung aufgefordert.

Aktualisierungs-Regeln:

- Wenn der Typ oder Name des Makro- E/A-Punktes mit der angeschlossenen Verbindung geändert wird, wird die Verbindung nach der Aktualisierung getrennt.
- Wenn dem Makro E/A-Punkte hinzugefügt werden, werden die vorhandenen E/A-Punkte nach dem Update nicht getrennt.
- E/A-Punkte des Makros werden nach Name und Typ identifiziert. Wenn Sie den Namen oder Typ eines E/A-Punkts mit einer externen Verbindung ändern und einen

neuen E/A-Punkt mit demselben Namen und Typ erstellen, wird die Verbindung nach der Makroaktualisierung automatisch mit dem neuen E/A-Punkt verknüpft.

6.3.6 Makro ersetzen

Wenn ein Makro durch ein anderes ersetzt werden soll, kann dies manuell erfolgen: löschen Sie das Makro, fügen Sie ein neues hinzu und stellen Sie die Verbindungslinien wieder her.

Es ist bequemer, für diesen Zweck das Makro-Kontextmenüpunkt **Ersetzen** zu benutzen. Dabei bleiben die Verbindungslinien zu den E/A-Punkten des Makros erhalten, wenn die Namen und Datentypen der alten und der neuen E/A-Punkte identisch sind.

Wenn der Name oder Datentyp eines E/A-Punkts nicht übereinstimmt, wird die Verbindungslinie unterbrochen und sollte manuell repariert werden.

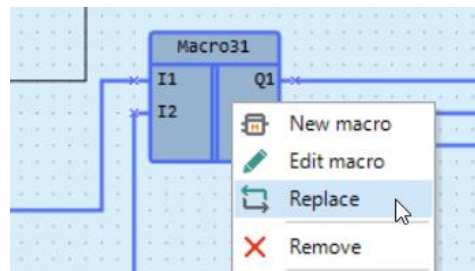



Abb. 6.90


6.3.7 Reihenfolge der E/A-Punkte ändern

Die E/A-Punkte des Makros werden an den Seiten des Makros in der Reihenfolge platziert, in der sie hinzugefügt wurden, von oben nach unten.

Diese Reihenfolge kann geändert werden. Dies kann nützlich sein, wenn Sie logisch verwandte E/A-Punkte in der Nähe platzieren möchten oder wenn Sie ein leeres Makro in das Programm einfügen und die Position seiner E/A-Punkte später nach der Entwicklung des Algorithmus bestimmen möchten.

Gehen Sie wie folgt vor:

- Öffnen Sie das Makro im Editor und ziehen Sie die E/A-Punkte in die gewünschte Reihenfolge.
- Klicken Sie auf das Symbolleistensymbol  **E/A-Reihenfolge synchronisieren**, um die Positionen der E/A-Punkte zu synchronisieren, und speichern Sie das Makro.
Hinweis: Die Synchronisation funktioniert nicht, wenn die Makro-E/A-Punkte nicht mit anderen Programmbausteinen verbunden sind.
- Gehen Sie zum Hauptprogramm. Das geänderte Makro wird rot hervorgehoben und es wird angeboten, es über das Kontextmenü zu aktualisieren (Abschn. 6.3.5).
- Nach der Aktualisierung wird die Reihenfolge der Makro-E/A-Punkte im Hauptprogramm dieselbe wie im Makro-Editor. Die Verbindungslinien des Makros bleiben erhalten.

Wenn der Synchronisationsschalter  nicht aktiviert ist, werden die Makro-E/A-Punkte im Hauptprogramm in der Standardreihenfolge angezeigt. Dies kann nützlich sein, wenn Sie dem Makro einen Eingang oder einen Ausgang hinzufügen müssen, aber die vorhandenen Verbindungslinien nicht verschränken möchten.

6.4 Displayelemente

Displayelemente sind Bibliothek-Bausteine, die die auf dem Gerätedisplay angezeigten Informationen steuern. Sie sind in der Library Box verfügbar, wenn ein Displayform-

Bibliothek

Arbeitsbereich aktiv ist, und können per Drag-and-Drop innerhalb der Displayform platziert werden. Folgende Elemente stehen zur Verfügung:

- Textfeld
- I/O-Box (INT/REAL)
- I/O-Box (BOOL)
- Dynamische Box
- Combobox

Verwenden Sie die Property Box, um ein Element anzupassen.

Die gemeinsamen Parameter für alle Elemente:

- **Koordinate X** – die Position des ersten (linken) Zeichenplatzhalters des Elements von der linken Formkante (von 0 bis 15).
- **Koordinate Y** – die Position des ersten (linken) Zeichenplatzhalters des Elements von der oberen Formkante, abhängig von der Anzahl der Zeilen im Formular.
- Es gibt zwei Möglichkeiten, die Koordinaten zu bestimmen: konstant (Standard) oder variabel. Um die von einer Variablen abhängige Koordinate zu verwenden, markieren Sie die Koordinate und öffnen die Liste rechts neben dem Eingabefeld.
 - **Konstant** – spezifizieren Sie die Koordinaten in der Property Box oder platzieren Sie das Element per Drag-and-Drop in der Form.



Abb. 6.91

- **Variabel** – klicken Sie auf **Auswählen**, um eine INT-Variable aus der Liste auszuwählen, und bestätigen Sie mit **OK**. Das Displayelement bewegt sich entsprechend dem Koordinatenwert, der durch die Variable gesteuert wird.
- **Länge** – die Anzahl der reservierten Zeichen. Das Displayelement belegt eine Display-Zeile in der Höhe, die Länge kann von 1 bis 16 Zeichen sein.

6.4.1 Textfeld

Ein Textfeld wird verwendet, um einfachen Text anzuzeigen.

Parameter:

Text – Text zum Anzeigen. Der Parameter **Länge** bestimmt die Anzahl der reservierten Zeichen an.

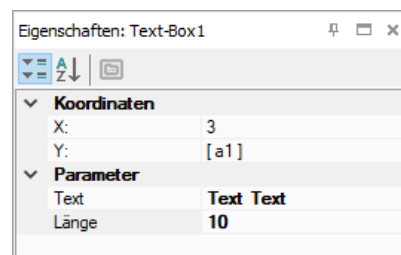


Abb. 6.92

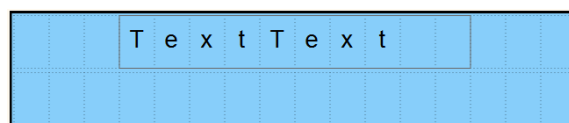


Abb. 6.93

6.4.2 I/O-Box (INT/REAL)

I/O-Box (INT/REAL) wird verwendet, um eine Variable vom Typ INT oder REAL anzuzeigen. Der Wert der Variablen kann mit den Geräte-Funktionstasten geändert werden.

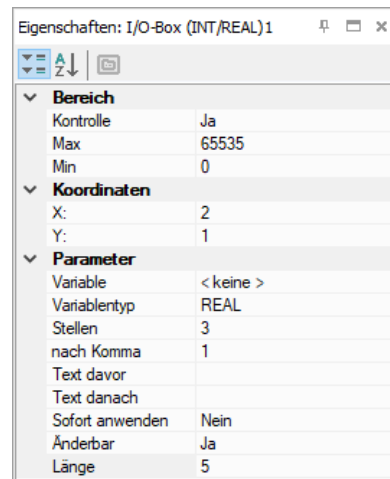



Abb. 6.94

Parameter:

Variable – die Verknüpfung mit einer Variablen. Verwenden Sie das Symbol  im Eingabefeld, um die Variable auszuwählen.

Datentyp – INT oder REAL. Wenn die Variable bereits ausgewählt wurde, wird ihr Datentyp übernommen.

Stellen – die Gesamtzahl der angezeigten Stellen

Dezimalstellen – die Anzahl der Stellen nach dem Komma: 0...6 Zeichen oder **Auto** für Auto-Genauigkeit *.

Text vor – der Text links von der angezeigten Variablen

Text nach – der Text rechts von der angezeigten Variablen

Editierbar – wenn **Ja**, kann der angezeigte Wert mit den Geräte-Funktionstasten geändert werden

***Hinweis:** Eine Ausgangsvariable sollte ausgewählt werden. Die Option hat keine Wirkung bei einer Eingangsvariablen.*

Länge – die Gesamtzahl der reservierten Zeichen, einschließlich des vorangestellten Textes und des nachfolgenden Textes

Bereich:

Die Parametergruppe dient zur Begrenzung des Eingabewertes. Wenn **Editierbar = Nein**, haben die Parameter dieser Gruppe keine Wirkung.

Begrenzen – wenn **Ja**, wird mit den Geräte-Funktionstasten eingegebene Wert durch die Benutzerparameter **Max** und **Min** begrenzt, ansonsten nur durch den verfügbaren Speicherbereich

Max – der Höchstwert für die Benutzereingabe

Min – der Mindestwert für die Benutzereingabe

Beispiel:

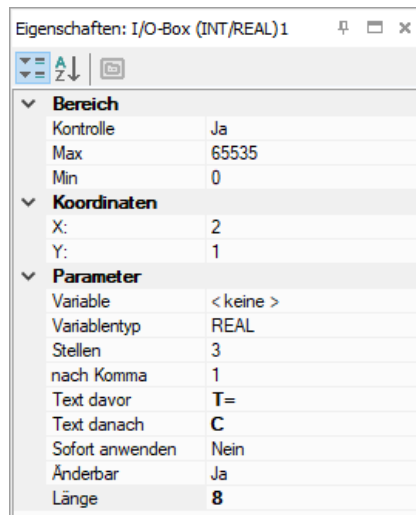


Abb. 6.95

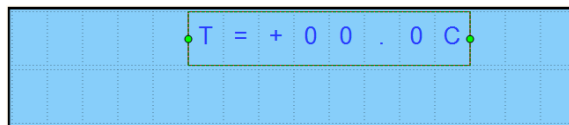


Abb. 6.96

* Auto-Genauigkeit

Mit dieser Option kann eine REAL-Variable für die festgelegte Anzahl reservierter Zeichen (Parameter **Digits**) am genauesten angezeigt werden. Um die Option zu verwenden, wählen Sie im Arbeitsbereich ein I/O-Box (INT/REAL) mit zugeordneter Variablen vom Typ REAL aus und setzen Sie den Parameter **Dezimalstellen** in Property Box auf **Auto**.

Beispiel:

Für die Darstellung der Variable VAR1 sind 4 Stellen mit automatischer Genauigkeit reserviert. Der Wert von 1,546745 wird abgerundet als 1,547 angezeigt. Wenn der Wert in 110,478696 geändert wird, wird er als 110,5 angezeigt.

6.4.3 I/O-Box (BOOL)

I/O-Box (BOOL) wird verwendet, um eine Variable vom Typ BOOL anzuzeigen. Der Wert der Variablen kann mit den Geräte-Funktionstasten geändert werden.

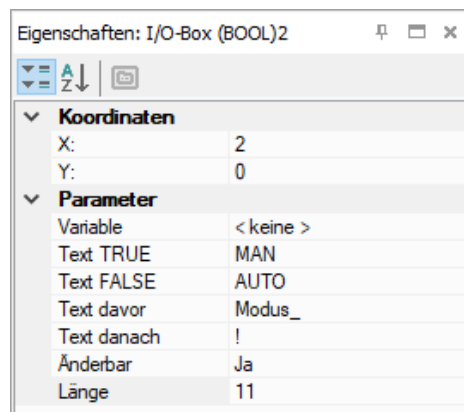



Abb. 6.97

Parameters:

Variable – die Verknüpfung mit einer Variablen. Verwenden Sie das Symbol  im Eingabefeld, um die Variable auszuwählen.

Text TRUE – der Text wird angezeigt, wenn die Variable **Wahr** ist

Text FALSE – der Text wird angezeigt, wenn die Variable **Falsch** ist

Text vor – der Text links von der angezeigten Variablen

Text nach – der Text rechts von der angezeigten Variablen

Editierbar – wenn **Ja**, kann der angezeigte Wert mit den Geräte-Funktionstasten geändert werden

Hinweis: Eine Ausgangsvariable sollte ausgewählt werden. Die Option hat keine Wirkung bei einer Eingangsvariablen.

Länge – die Gesamtzahl der reservierten Zeichen, einschließlich des vorangestellten Textes und des nachfolgenden Textes

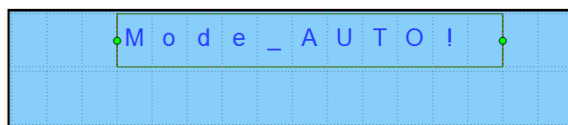


Abb. 6.98

6.4.4 Dynamische Box

Dynamische Box ist ein Ausgabefeld. Es wird verwendet um, abhängig von der Zeilen-**ID**, eine der Textzeilen aus einer Liste anzuzeigen. Die Zeilen-**ID** wird in einer verknüpften Variablen vom Typ INT gespeichert.

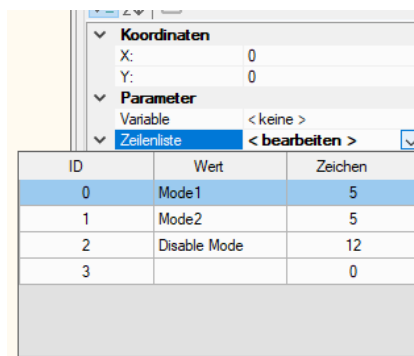



Abb. 6.99

Parameters:

Variable – die Verknüpfung mit einer Variablen. Verwenden Sie das Symbol  im Eingabefeld, um die Variable auszuwählen.

Zeilenliste – die Tabelle mit Textzeilen. Der **Text** in der Zeile wird angezeigt, wenn der Wert der verknüpften Variablen gleich der Zeilen-**ID** ist. Die Spalte **Zeichen** zeigt die Anzahl der Zeichen im Text an. Neben der Anzahl wird ein Ausrufezeichen angezeigt, wenn der Wert des Parameters **Länge** überschritten wird.

Länge – die Anzahl der reservierten Zeichen

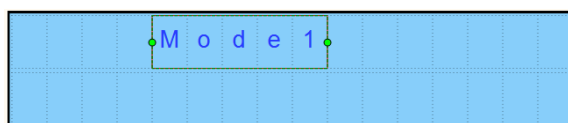


Abb. 6.100

6.4.5 ComboBox

ComboBox ist ein Ein-/Ausgabefeld. Es wird verwendet um, abhängig von der Zeilen-**ID**, eine der Textzeilen aus einer Liste anzuzeigen. Die Zeilen-**ID** wird in einer verknüpften Variablen vom Typ INT gespeichert. Die **ID** kann auch mit den Geräte-Funktionstasten ausgewählt werden.

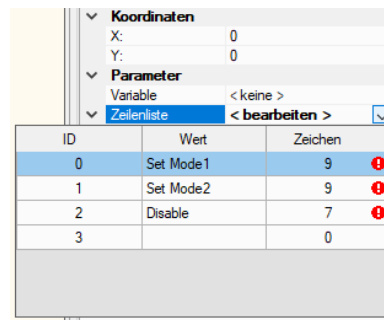



Abb. 6.101

Parameters:

Variable – die Verknüpfung mit einer Variablen. Verwenden Sie das Symbol  im Eingabefeld, um die Variable auszuwählen.

Zeilenliste – die Tabelle mit Textzeilen. Der **Text** der ausgewählten Zeile wird angezeigt und die Zeilen-**ID** wird in der verknüpften Ausgangsvariablen gespeichert. Die Spalte **Zeichen** zeigt die Anzahl der Zeichen im Text an. Neben der Zahl wird ein Ausrufezeichen angezeigt, wenn der Wert des Parameters **Länge** überschritten wird.

Länge – die Anzahl der reservierten Zeichen

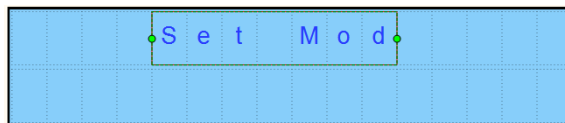


Abb. 6.102

7 Schaltprogrammentwicklung

Es wird empfohlen, die Erstellung eines Schaltprogramms mit der Planung zu beginnen. Der Plan sollte alle möglichen Zustände des Geräts während des Betriebs in Form eines Modus-Diagramms, einer Tabelle der E/A-Zustände, eines elektrischen oder Funktionsdiagramms usw. beschreiben.

Nachdem alle operativen Aufgaben beschrieben sind, kann das Programm unter Nutzung von Standard-Bausteinen aus der Symbolleiste **Einfügen** (Tabelle 2.6) und spezifischen Bausteinen aus der Projektbibliothek (Abschn. 6) entwickelt werden. Die in Library Box (Abschn. 2.4) dargestellte Projektbibliothek enthält die Funktionen (Abschn. 6.1) und Funktionsbausteine (Abschn. 6.2), die für das Zielgerät verfügbar sind, sowie die dem Projekt hinzugefügten Makros (Abschn. 6.3).

Weitere Informationen zur Verwendung der einzelnen Bausteine finden Sie in den Abschn. 7.1 - 7.8, weitere Praktiken der Programmentwicklung in den Abschn. 7.9 - 7.11.

Um Verbindungslinien zu zeichnen, verwenden Sie die linke Maustaste:

- Klicken Sie auf den Ausgangs-Pin des ersten Bausteins. Die Linie ist daran angehängt und folgt dem Mauszeiger.
- Um die Linienrichtung zu ändern, klicken Sie auf den Arbeitsbereich.
- Ziehen Sie die Linie bis zum Eingangs-Pin des zweiten Bausteins und klicken Sie darauf, um die Linie zu beenden.

Verbindungslinie kann nur zwischen Baustein-Anschlusspins gezogen werden, die demselben Datentyp zugeordnet sind. Um die Anschlusspins zu verbinden, die verschiedenen Datentypen zugeordnet sind, verwenden Sie Konvertierungsblöcke (Abschn. 7.8).

Klicken Sie auf den Baustein, um ihn auszuwählen. Ziehen Sie das Rechteck um mehrere Bausteine, um eine Gruppe auszuwählen.

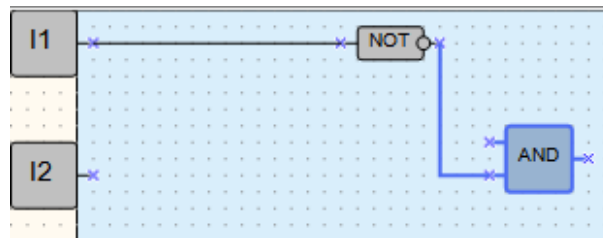


Abb. 7.1

Die Parameter der Programmbausteine können in der Property Box festgelegt werden (Abschn. 2.5).

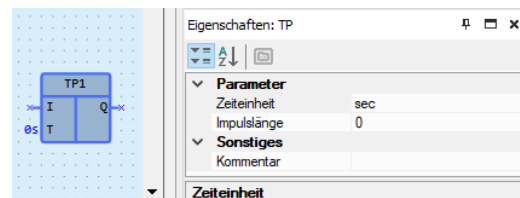


Abb. 7.2

Verwenden Sie das Baustein-Kontextmenü für alle mit dem Baustein verfügbaren Manipulationen.

7.1 Verwendung von Bibliotheks-Bausteinen

Um einen Bibliotheks-Baustein in das Schaltprogramm zu übernehmen, markieren Sie den gewünschten Baustein im Library Box und verschieben es per Drag-and-Drop auf den Arbeitsbereich.

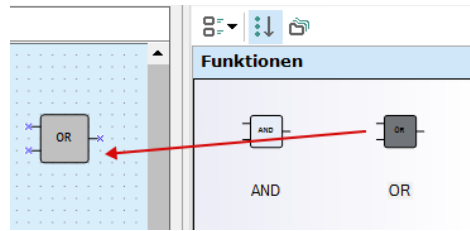


Abb. 7.3

7.2 Verwendung von Textfeld

Textfelder können verwendet werden, um dem Programm Erklärungen hinzuzufügen.

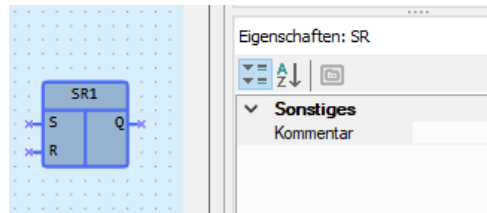


Abb. 7.4


Um dem Programm ein Textfeld hinzuzufügen, klicken Sie auf das Symbol  in der Symbolleiste **Einfügen** (Tabelle 2.6), dann auf einen Platz im Arbeitsbereich, um den Textfeld zu platzieren, und zeichnen Sie mit der linken Maustaste ein Rechteck.



Abb. 7.5

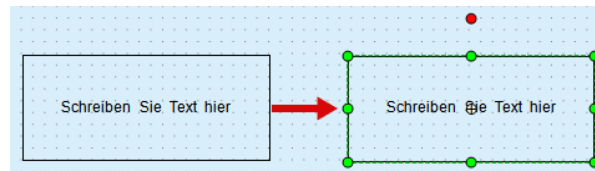


Abb. 7.6

Doppelklicken Sie auf das Textfeld, um den Text zu schreiben.

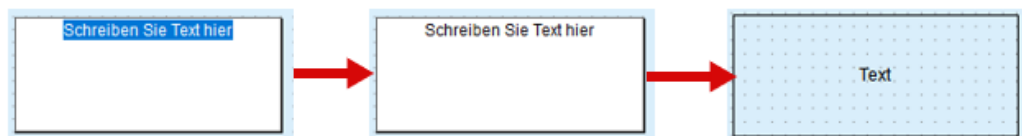


Abb. 7.7

Die Parameter des Textfeldes können in der Property Box geändert werden.

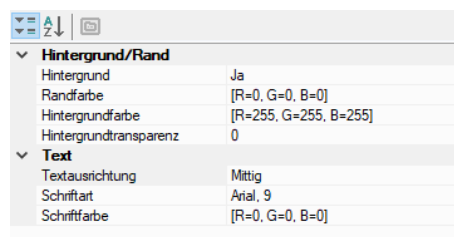


Abb. 7.8

7.3 Verwendung von Variablen

Um dem Programm eine Variable hinzuzufügen, klicken Sie auf das entsprechende Symbol in der Symbolleiste **Einfügen** (Tabelle 2.6) und klicken Sie dann auf einen Platz im Arbeitsbereich, um den Variablenbaustein zu platzieren.



Abb. 7.9

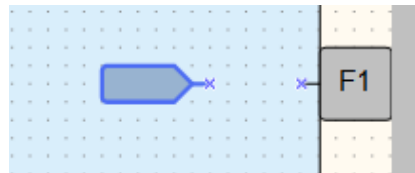


Abb. 7.10

Um einem Baustein eine Variable zuzuweisen, klicken Sie auf das Symbol in der Zeile **Variable** in der Property Box. Die Variablen-tabelle wird geöffnet, aber nur die entsprechenden Registerkarten in der Tabelle sind sichtbar. Die Auswahl der angezeigten Variablen richtet sich nach dem Typ des Bausteins.

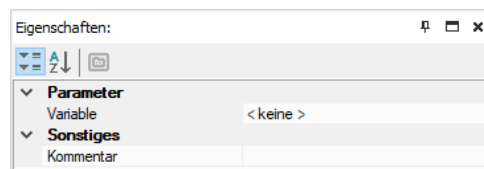


Abb. 7.11

Wählen Sie eine Variable aus oder erstellen Sie eine neue in der Variablen-tabelle (Abschn. 5) und bestätigen Sie mit **OK**.

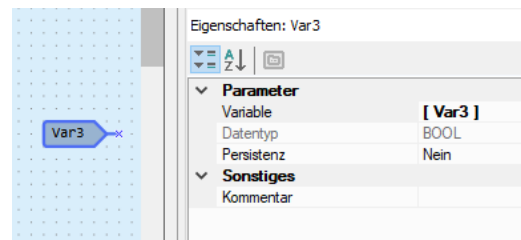


Abb. 7.12

Verwenden Sie Netzwerkvariablen, um Daten mit anderen Geräten auszutauschen, die mit dem Zielgerät über Netzwerk verbunden sind. Weitere Informationen zur Verwendung von Netzwerkvariablen finden Sie im Abschn. 7.6.

Wenn ein Variablenbaustein rot markiert ist, bedeutet dies, dass die Erstellung falsch oder nicht abgeschlossen ist.

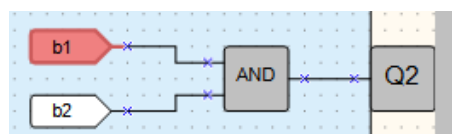


Abb. 7.13


Die Informationen über den Fehler werden in der Statusleiste angezeigt.

Es wird empfohlen, die Programmierung mit der Erstellung von Variablen in der Variablen-tabelle zu starten.

Schaltprogrammentwicklung

Wenn eine Variable in einem Projekt mehrmals verwendet wird, können alle Verknüpfungen mit dem Punkt **Referenzen anzeigen** im Variablenblock-Kontextmenü verfolgt werden. Die Funktion ist auch im Simulations- oder Online-Debugging-Modus verfügbar.

7.4 Verwendung von Konstanten

Um dem Programm einen konstanten Wert hinzuzufügen, klicken Sie auf das Symbol  in der Symbolleiste **Einfügen** (Tabelle 2.6) und klicken Sie dann auf einen Platz im Arbeitsbereich, um den Konstanten-Baustein zu platzieren.

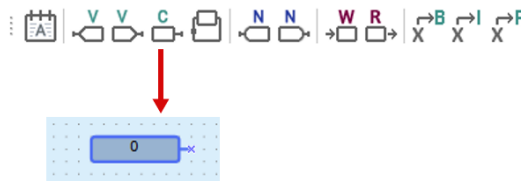



Abb. 7.14

Wählen Sie den Datentyp über das Symbol  in der Zeile **Datentyp** aus und geben Sie den Wert in der Zeile **Konstanter Wert** im Property Box ein.

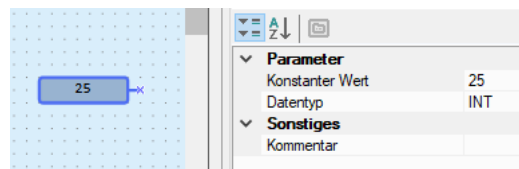


Abb. 7.15


Der Wert der Konstante kann sich während der Programmausführung nicht ändern. Er kann nach einem Doppelklick auf den konstanten Block im Property Box oder durch Auswahl vom Punkt **Wert ändern** im Block-Kontextmenü geändert werden.

Tabelle 7.1 Gültige Werte für Datentypen

Datentyp	Gültige Werte
BOOL	0 / 1
INT	0...4.294.967.295
REAL	-3,402823e+38...3,402823e+38

7.5 Verwendung von Verzögerungslinien

Über eine Verzögerungslinie wird der Wert von einem Bausteinausgang um einen Zyklus verzögert zu einem Bausteineingang übertragen. Ausgang und Eingang können zu unterschiedlichen Bausteinen gehören.

Klicken Sie auf das Symbol  in der Symbolleiste **Einfügen** (Tabelle 2.6) und zeichnen Sie eine Linie vom Ausgang bis zum Eingang einer Funktion oder eines Funktionsbausteins. Die Verzögerungslinie wird als rote gestrichelte Linie mit einem Pfeil angezeigt.

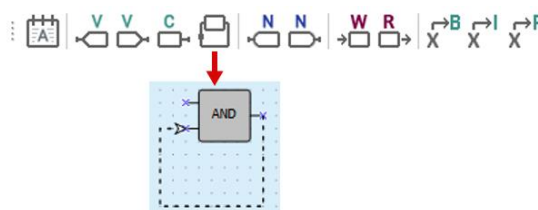


Abb. 7.16

Beispiel:

Schaltprogrammentwicklung

Ein konstanter Wert 1 wird an den Eingang I1 des Additionsbausteins ADD (INT) übertragen. Ein Wert aus dem im vorhergehenden Zyklus berechneten Bausteinausgang (Q) wird über die Verzögerungslinie an den Eingang I2 übertragen.

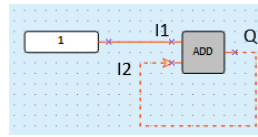


Abb. 7.17

Tabelle 7.2 Zyklus-Signalwerte

Zyklus	1	2	3	4	5	6	7	8	9	10
I2	0	0	1	1	2	2	3	3	4	4
Q	1	1	2	2	3	3	4	4	5	5

7.6 Netzwerk-Datenaustausch

Die Netzwerkeingangs- und -ausgangsvariablen sind spezielle Arten von Variablen für den Datenaustausch zwischen Geräten, die in einem gemeinsamen Netzwerk miteinander verbunden sind.

Die Variablen, die über das Netzwerk gelesen werden können, werden als Netzwerkeingangsvariablen (\square^N) bezeichnet.

Die Variablen, die über das Netzwerk geschrieben werden können, werden als Netzwerkeingangsvariablen (\square^N) bezeichnet.

Um dem Programm eine Netzwerkvariable hinzuzufügen, gehen Sie wie folgt vor:

- Klicken Sie auf das Symbol \square^N oder \square^N in der Symbolleiste **Einfügen** (Tabelle 3.6).
- Klicken Sie auf einen Platz im Arbeitsbereich, um die Variable einzufügen.
- Klicken Sie auf das Symbol \square^N in der Zeile **Variable** im Property Box, um eine Variable für den Baustein auszuwählen.

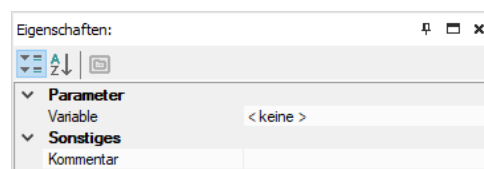


Abb. 7.18

- Wählen Sie eine Variable aus oder erstellen Sie eine neue in der geöffneten Variablen- und Wertetabelle und bestätigen Sie mit **OK**. Die gewählte Variable wird dem Variablenbaustein zugewiesen.

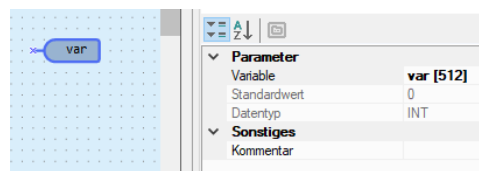


Abb. 7.19

- Verbinden Sie die Netzwerkvariable mit dem gewünschten Baustein im Arbeitsbereich.

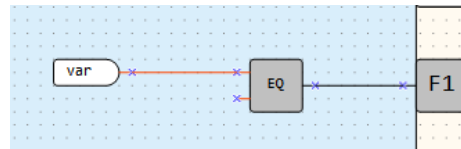


Abb. 7.20

Wenn der Netzwerkvariablen-Ausgangsbaustein für Abfrage eines Slaves verwendet wird, bekommt er einen zusätzlichen Parameter **Schreiben am Zyklusende**. Wenn der Parameter auf **Ja** gesetzt ist, wird der neue Wert der Variablen erst am Ende des Zyklus zugewiesen, wenn alle Eingangsvariablen bereits ausgelesen wurden.



Abb. 7.21

Hinweis: Eine Variable kann nicht mit einem Baustein gekoppelt werden, wenn in den Geräteeinstellungen keine Kommunikationsschnittstellen vorhanden sind.

7.7 Lesen / Schreiben in FB

Der Baustein **WriteToFB** wird verwendet, um einen Parameter eines FBs während des Prozesses zu ändern.

Beispiel:

Der Wert des Parameters **Einschaltdauer** des FBs **BLINK1** sollte je nach Wert am Eingang **I5** entweder 2 oder 10 betragen.

Um dem Programm den Baustein **WriteToFB** hinzuzufügen, klicken Sie auf das Symbol in der Symbolleiste **Einfügen** (Tabelle 2.6) und klicken dann auf einen Platz im Arbeitsbereich, um den Baustein zu platzieren.

Gehen Sie zur Property Box, wählen Sie den FB **BLINK1** in der Zeile **Funktionsbaustein** und den Parameter des FB in der Zeile **Parameter in FB** (Abb. 7.22) aus.

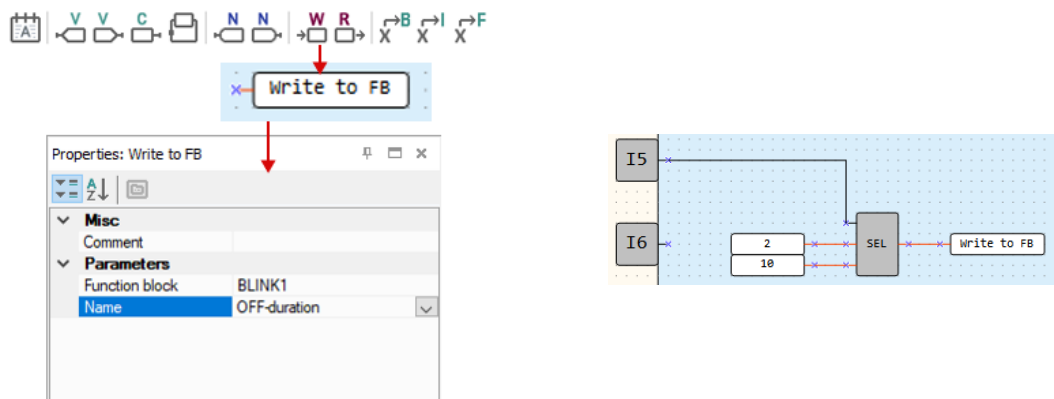


Abb. 7.22

Der Baustein **ReadFromFB** wird verwendet, um den aktuellen Wert eines FB-Parameters zu lesen und ihn im Programm zu verwenden. Die Verwendung ist die gleiche wie beim Baustein **WriteToFB**.

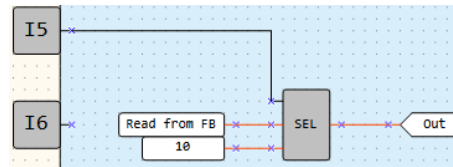


Abb. 7.23

7.8 Konvertierungsblöcke



Fig. 7.24

Konvertierungsblöcke sind Universalblöcke, mit denen ein Eingangswert eines beliebigen Typs in einen Wert eines bestimmten Typs umgewandelt wird. In der Symbolleiste **Einfügen** stehen drei Blöcke zur Verfügung (Tabelle 2.6):

Tabelle 7.3 Konvertierungsblöcke

Konvertierung in BOOL	Konvertierung von INT oder REAL in BOOL (Wenn der Eingangswert > 0, der Ausgang = 1 / Wahr)
Konvertierung in INT	Konvertierung von BOOL oder REAL in INT (REAL wird auf INT abgerundet)
Konvertierung in REAL	Konvertierung von BOOL oder INT in REAL

Um den Konvertierungs-Baustein zum Programm hinzuzufügen, klicken Sie auf eines der drei Symbole in der Symbolleiste **Einfügen** (Tabelle 2.6) und anschließend auf die gewünschte Stelle im Arbeitsbereich.

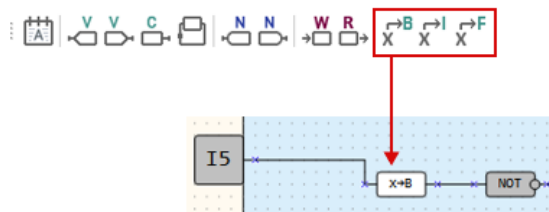



Abb. 7.25

7.9 Elemente anordnen

Die Sequenznummern der Funktionsbausteine können durch Anklicken der Schaltfläche  **Elemente anordnen** in der Symbolleiste **Service** (Tabelle 2.6) automatisch neu zugewiesen werden. Die Bausteine desselben Typs werden nacheinander von oben nach unten und von links nach rechts nummeriert.

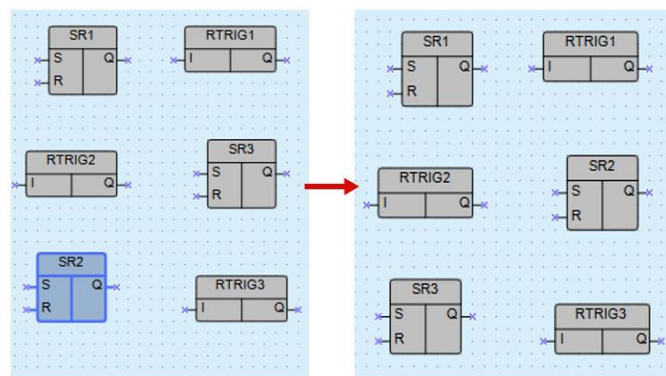


Abb. 7.26

7.10 Ausführungsablauf

Die Berechnung der Werte für Ausgänge und Verzögerungslinien erfolgt in einer bestimmten Reihenfolge. Um diese Reihenfolge zu sehen, klicken Sie auf den Pfeil neben dem Symbol in der Symbolleiste **Service** und wählen Sie die **Verzögerungslinien** oder **Ausgänge** aus (Tabelle 2.6).

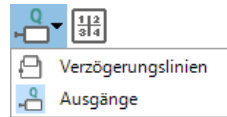


Abb. 7.27

Um die Reihenfolge zu ändern, doppelklicken Sie auf einen Ausgang oder eine Verzögerungslinie und geben die gewünschte Nummer ein.

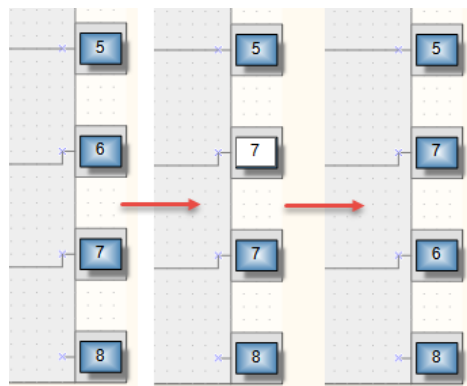


Abb. 7.28

Klicken Sie auf das Symbol , um den Bearbeitungsmodus zu deaktivieren.

7.11 Simulation

Verwenden Sie die Simulation, um die Korrektheit des erstellten Programms zu bestätigen. Nur die Offline-Simulation ist zurzeit möglich. Die Simulation ermöglicht es, die Werte aller Signale innerhalb des Schaltprogramms zu analysieren. Ändern Sie die Werte an den digitalen und analogen Eingängen sowie von Variablen und Konstanten und überprüfen Sie die Werte an den Ausgängen.

7.11.1 Arbeitsweise

Um die Simulationsmodus zu starten / beenden, klicken Sie auf das Symbol **Simulation** in der Symbolleiste **Service** (Tabelle 2.6). Die neue Symbolleiste **Simulation** wird mit den folgenden Steuerelementen angezeigt:



Abb. 7.29

Tabelle 7.4 Simulations-Symbolleiste

	Start	Permanente Simulation starten
	Einzelzyklus	Schritt-für-Schritt-Simulation. Beim Anklicken des Symbols wird ein Programmzyklus ausgeführt.
	Pause	Simulation anhalten. Klicken Sie erneut auf das Symbol, um die Simulation fortzusetzen.

Schaltprogrammentwicklung

	Stop	Simulation beenden
Refresh time (ms) 100	Aktualisierung	Simulation-Aktualisierungszeit (Zyklusstartperiode) in Millisekunden
Cycle time 100	Zykluszeit	Zykluszeit (Abschn. 3.2)
ms	Zeiteinheit	Auswahl der Maßeinheiten für die Zykluszeit: Millisekunden, Sekunden, Minuten, Stunden
	Sichtfenster	Sichtfenster (Abschn. 7.11.2) öffnen / schließen, um die Variablenwerte bei jedem Programmschritt zu verfolgen

Die zusätzliche Symbolleiste **Kalender** wird im Simulationsmodus angezeigt, wenn im Projekt Funktionsbausteine vom Typ CLOCK oder CLOCKW vorhanden sind (nur für Geräte mit Echtzeituhr).

Datum/Uhrzeit 10:37:56 19 Jul 2019

Abb. 7.30 Kalender-Symbolleiste

Der Parameter **Aktualisierung (ms)** gibt die Zyklusstartperiode für die Simulation an. Die Parameter **Zykluszeit** gibt die Programm-Zyklusdauer an.

Hinweis: Die Parameter **Zykluszeit** im Gerät (Abschn. 3.2) und im ALP-Simulationsmodus sind trotz des gleichen Namens unterschiedlich.

Sie können die Werte der Eingänge, Ausgänge und Variablen festlegen, indem Sie auf den Baustein im Arbeitsbereich klicken und im geöffneten Dialogfeld einen neuen Wert eingeben. Es kann auch der Wert einer Netzwerkvariablen festgelegt werden.

Die digitalen Ein- und Ausgänge ändern nach jedem Klick ihren Wert und entsprechend ihre Farbe.

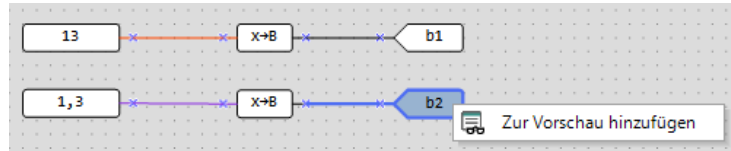


Abb. 7.31

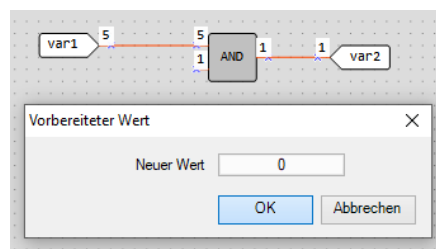


Abb. 7.32

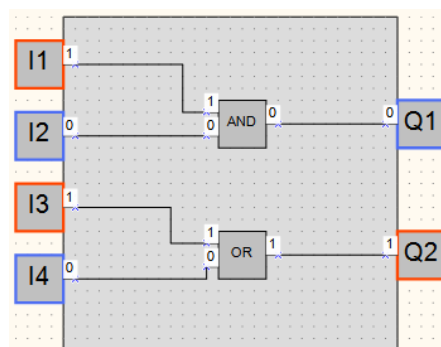


Abb. 7.33


Schaltprogrammentwicklung

Wenn Sie das Programm ändern wollen, stoppen Sie die Simulation.

Hinweise:

- Makros sind von der Simulation ausgeschlossen. Die Simulation für Makros sollte separat im Arbeitsbereich des Makros durchgeführt werden.
- Simulation kann nicht ausgeführt werden für:
 - Bausteine ohne Verbindung mit einem Geräteausgang oder einer Netzwerkvariablen
 - Nicht korrekt gekoppelte Variablen
 - Retain-Variable

7.11.2 Beobachtungsfenster

Klicken Sie auf das Symbol  der Simulations-Symbolleiste, um die Eingangs-, Ausgangs- oder Variablenwerte bei jedem Programmschritt zu verfolgen.

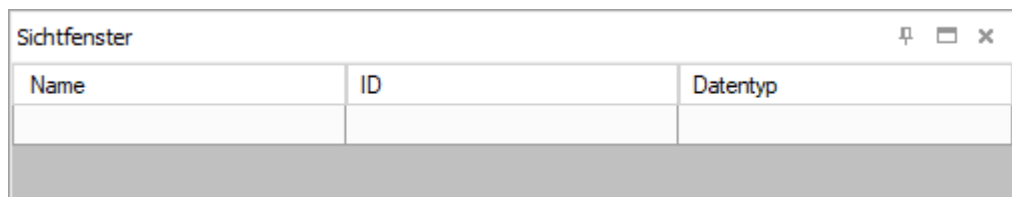



Abb. 7.34 Beobachtungsfenster

Um dem Beobachtungsfenster eine Variable, ein Eingang oder ein Ausgang hinzuzufügen, klicken Sie in das leere Feld in der Spalte **Name** und dann auf das links angezeigte Symbol . Das Kontextmenü dieser Bausteine kann ebenfalls verwendet werden.

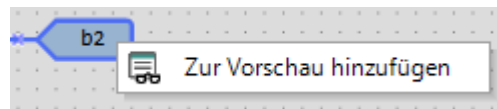


Abb. 7.35 Variablen-Kontextmenü in der Simulation

Die Werte der Variablen, Ein- und Ausgänge können während der Simulation in der Spalte **Wert** eingestellt werden.

7.12 Online-Debugging

Um das Online-Debugging zu starten, klicken Sie auf das Symbol  in der Symbolleiste.

In diesem Modus werden die aktuellen Werte aller Programmvariablen einschließlich Funktionen, Funktionsblöcken, Makros, Ein- und Ausgängen aus dem angeschlossenen Gerät ausgelesen und im Arbeitsbereich angezeigt. Auf diese Weise können Sie die Logik des Geräteprogramms überprüfen.




Abb. 7.36

Das Online-Debugging ist nur möglich, wenn:

- Das Gerät ist an den PC angeschlossen
- Das Programm im Gerät und das Programm geöffnet in ALP ist das gleiche
- Die Version der Gerätefirmware ist mit der aktuellen Version von ALP kompatibel

Schaltprogrammentwicklung

Das Online-Debugging ist nur für das Hauptprogramm verfügbar, nicht für Makros.

Während des Online-Debugging können im Projekt keine Änderungen vorgenommen werden. Um Korrekturen vorzunehmen, beenden Sie das Online-Debugging, indem Sie erneut auf das Symbol  klicken.

Hinweis: Wenn die Kommunikation mit dem Gerät unterbrochen wird, wird das Online-Debugging nach 10 Sekunden beendet und das Gerät in den Betriebsmodus geschaltet. Wird die Verbindung innerhalb von 10 Sekunden wiederhergestellt, wird das Online-Debugging fortgesetzt, die eingegebenen Werte werden jedoch zurückgesetzt.

Manuelle Werteeingabe

Sie können den Eingangswert einer Variablen während des Online-Debugging ändern, indem Sie auf den angezeigten Wert klicken (Abb. 7.34). Der neue Wert sollte in das Feld **Neuer Wert** im geöffneten Fenster **Vorbereiteter Wert** eingegeben werden. Es gibt zwei Möglichkeiten, um den Wert zu ändern: einmalige oder dauerhafte Änderung.

Durch die einmalige Änderung kann der variable Eingangswert für einen Programmzyklus geändert werden. In den folgenden Zyklen wird das Signal eines der Geräteeingänge oder der Ausgang einer anderen an diesen Eingang angeschlossenen Programmkomponente verwendet. Diese Option ist nützlich für die Simulation von Einzelimpulsen.

Bei Verwendung der Option **Permanente Änderung** wird der eingegebene Wert auf den Eingang angewendet, bis die Option deaktiviert oder das Online-Debugging gestoppt wird. Die Option ist nützlich für die Simulation von zeitkonstanten Signalen.

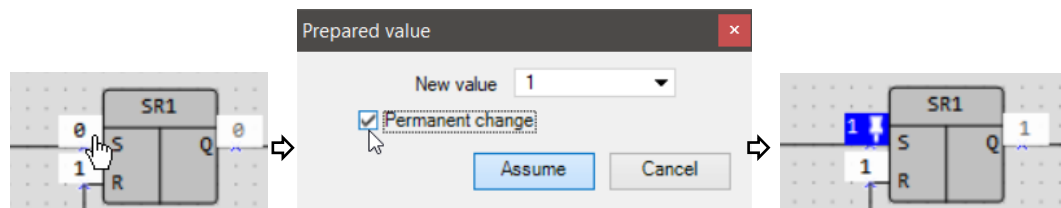


Abb. 7.37

Hinweis: Der Datenaustausch im Online-Debugging-Modus ist begrenzt und hängt vom Gerätemodell ab. Ein Zeichen dafür, dass die Einschränkung ausgelöst wird, sind leere Wertzellen im Diagramm während des Online-Debugging. In diesem Fall sollten Sie den Arbeitsbereich vergrößern (siehe 2.3), damit weniger Werte in den sichtbaren Bereich fallen. Die eingegebenen Werte, die nicht in den sichtbaren Bereich fallen, bleiben gespeichert, belegen jedoch nicht den für den Datenaustausch reservierten Speicherbereich.

8 Display-Programmierung

Um die anzuzeigenden Informationen zu bestimmen, verwenden Sie die Registerkarte **Display-Manager** in der oberen linken Ecke des Fensters. Der Display-Manager (Abschn. 2.6) ist nur für Zielgeräte mit Display verfügbar.

Ein Display kann mit einer oder mehreren Displayformen und Sprüngen von einer zu anderer programmiert werden, so dass die angezeigten Informationen durch Programmereignisse (Änderung der Variablen) oder durch Operator (Tasteneignis) geändert werden können.

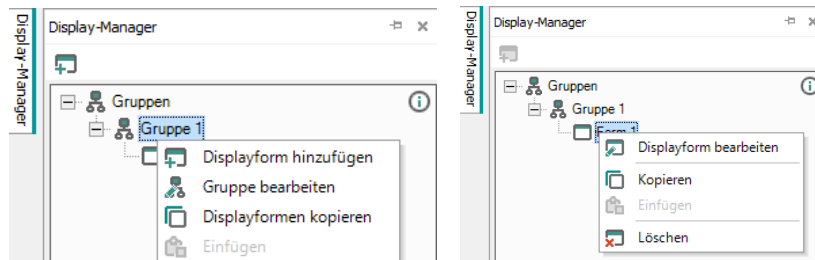


Abb. 8.1

Um die ausgewählte Displayform im Display-Editor (Abschn. 8.1) zu öffnen, verwenden Sie den Punkt **Displayform bearbeiten** im Form-Kontextmenü oder doppelklicken Sie auf die Form im Strukturbaum (Abb. 8.1).

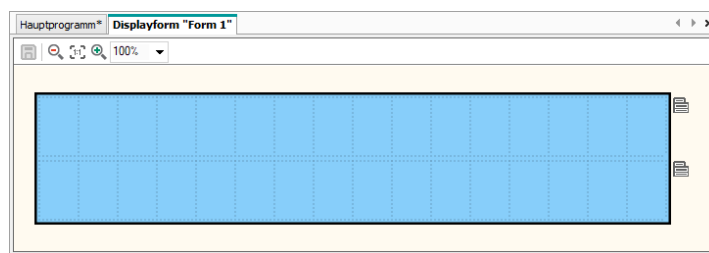


Abb. 8.2 Display-Editor

Um eine grafische Struktur von Displayformen im Struktur-Editor (Abschn. 8.2) zu öffnen, verwenden Sie den Punkt **Gruppe bearbeiten** im Gruppenkontextmenü (Abb. 8.1)

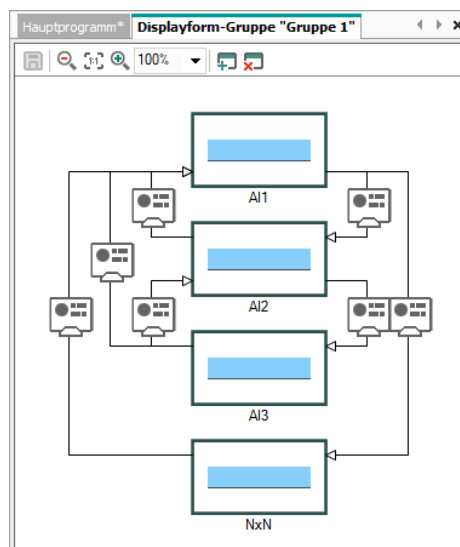



Abb. 8.3 Struktur-Editor

8.1 Display-Editor

Eine Form kann aus mehreren Zeilen bestehen, mindestens zwei. Der Operator kann mit den Geräte-Funktionstasten zwischen ihnen wechseln. Im Arbeitsbereich wird eine Displayform mit den Symbolen  am rechten Rand angezeigt, mit denen die Anzahl der angezeigten Zeilen geändert werden kann. Die zuerst angezeigten Zeilen werden umrandet.

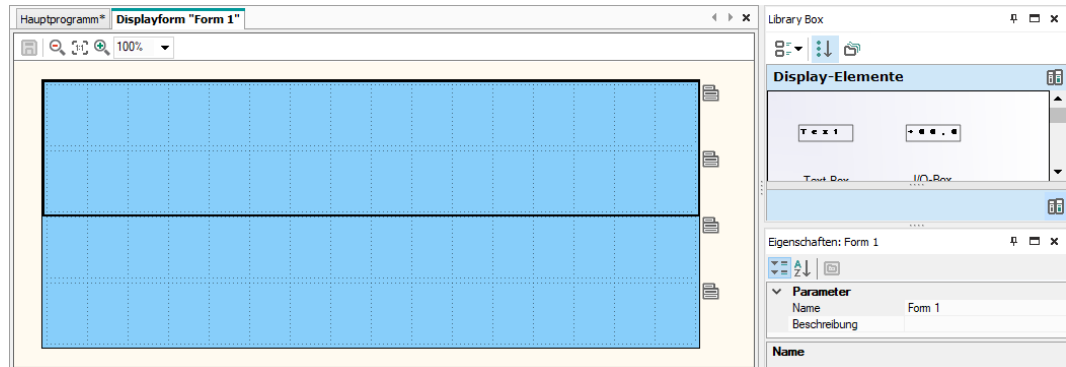




Abb. 8.4

Platzieren Sie die Displayelemente aus der Bibliothek per Drag-and-Drop auf die Form. Weitere Informationen zu Displayelemente finden Sie in Abschn. 6.4.


8.2 Grafische Struktur

Um eine grafische Struktur von Displayformen im Struktur-Editor (Abschn. 2.6) zu öffnen, verwenden Sie den Punkt **Gruppe bearbeiten** im Gruppenkontextmenü (Abb. 8.1).

Standardmäßig besteht die grafische Struktur aus einer Form. Um einer Gruppe eine Displayform hinzuzufügen:

- Im Display Manager verwenden Sie das Gruppenkontextmenü oder das Symbol  (aktiv, wenn eine Gruppe ausgewählt ist) (Abb. 8.1).
- Im Struktur-Editor verwenden Sie das Symbol  (Abb. 8.2).

Um eine Form aus der Gruppe zu entfernen:

- Im Display Manager verwenden Sie den Punkt **Löschen** im Form-Kontextmenü (Abb. 8.1)
- Im Struktur-Editor verwenden Sie das Symbol  oder das Form-Kontextmenü.

Um die Position einer Displayform innerhalb einer Gruppe im Display-Manager zu ändern, halten Sie die Umschalttaste gedrückt, während Sie diese Form ziehen und ablegen.

Wenn die Display-Struktur aus mehreren Formen besteht, sollten Sprünge definiert werden, um die Navigation zwischen Formen zu ermöglichen (Abschn. 8.3, Abb. 8.2).

Informationen zum Kopieren / Einfügen von Displayformen finden Sie in Abschn. 8.4.

8.3 Displayform-Eigenschaften / Sprünge

Wählen Sie eine Form in der Struktur aus, um seine Eigenschaften zu sehen:

- Name
- Beschreibung
- Sprungparameter

Jede neue Displayform hat einen Standardnamen, der hier geändert werden kann. Die Beschreibung ist optional.

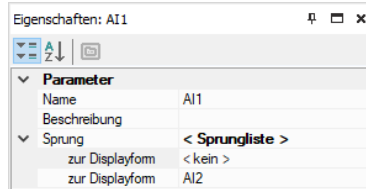



Abb. 8.5

Um die Navigation durch die Formen mit verschiedenen Ereignissen zu programmieren, klicken Sie auf das Symbol  in der Zeile **zur Displayform**.

Wählen Sie im geöffneten Dialog **Sprung** im Abschnitt **zur Displayform** die Form aus, zu der das Display wechseln soll, wenn das Ereignis eintritt, die im Abschnitt **Sprungbedingung** angegeben ist.

Wählen Sie das Ereignis im Bereich **Sprungbedingung** als Geräteereignis oder als Änderung eines Variablenwertes.

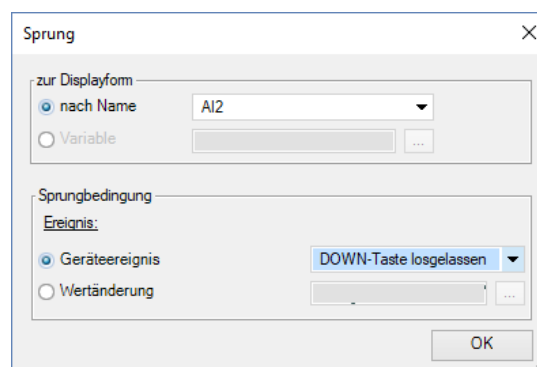


Abb. 8.6

Ein Tastenereignis kann als **Geräteereignis** ausgewählt werden.

Für die **Wertänderung** als Ereignis kann eine BOOL-Variable ausgewählt werden.

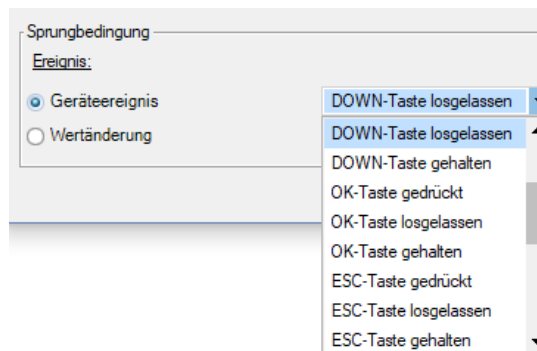


Abb. 8.7

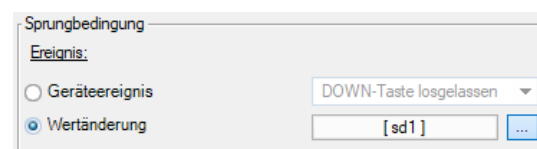


Abb. 8.8

Bestätigen Sie mit **OK**. Der angelegte Sprung wird in der Struktur angezeigt.

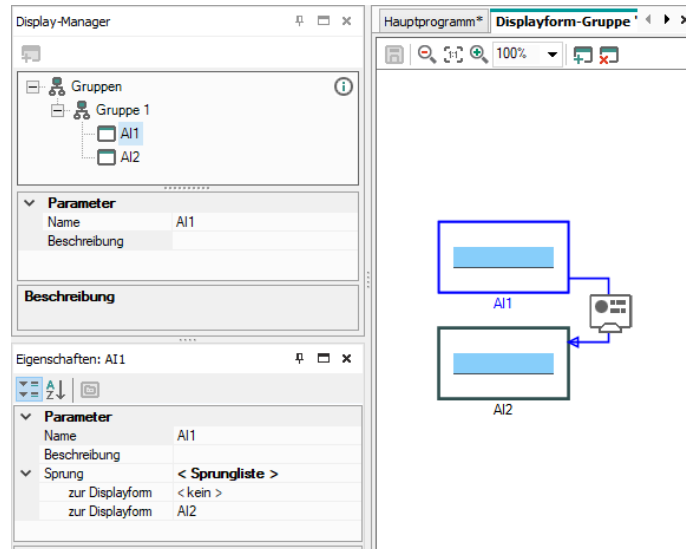


Abb. 8.9

Der Sprung zwischen zwei Formen kann durch mehrere Ereignisse erfolgen und die grafische Struktur kann eine sehr hohe Komplexität erreichen.

8.4 Displayform kopieren / einfügen

Eine Displayform oder eine Gruppe von Formen kann kopiert und in dasselbe oder ein anderes Projekt eingefügt werden.

Um eine Displayform oder eine Gruppe von Formen zu kopieren, verwenden Sie den Punkt **Kopieren** im Formkontextmenü und **Gruppe kopieren** im Gruppenkontextmenü, oder einfach die Tastenkombination **Strg + C**. Mit den **Strg**- und **Umschalt**-Tasten können auch mehrere Formen ausgewählt werden.

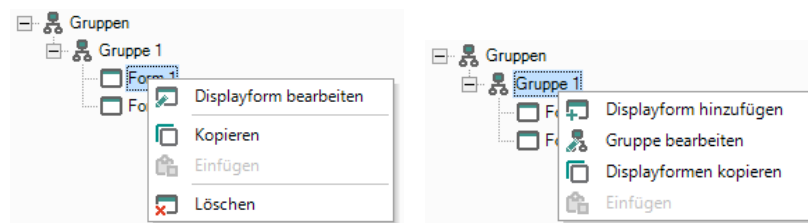


Abb. 8.10

Verwenden Sie den Punkt **Einfügen** im Form- oder Gruppenkontextmenü oder die Tastenkombination **Strg + V**, um das Form in die Gruppe einzufügen. Die mit der Form verknüpften Variablen werden gemäß der in Abschnitt 5.6 beschriebenen Regeln mitkopiert. Sprünge zwischen den ausgewählten Formen werden mitkopiert. Wenn nur eine der durch den Sprung verbundenen Formen ausgewählt ist, wird der Sprung beim Einfügen gelöscht.

9 Tastaturkürzel

Tastaturkürzel	Beschreibung	Verwendung
Strg + N	Neues Projekt	Menü / Datei
Strg + O	Projekt öffnen	Menü / Datei
Strg + Alt + S	Projekt speichern unter...	Menü / Datei
Strg + S	Projekt speichern	Menü / Datei
Strg + P	Drucken...	Menü / Datei
Strg + Z	Rückgängig	Menü / Ansicht
Strg + Y	Wiederherstellen	Menü / Ansicht
Strg + F7	Programm auf Gerät übertragen	Menü / Gerät
F1	Hilfe	Menü / Hilfe
Strg + C	Kopieren	Arbeit mit Elementen
Strg + V	Einfügen	Arbeit mit Elementen
Entf	Löschen	Arbeit mit Elementen
Strg + →	Elementbreite erhöhen	Elementgröße ändern
Strg + ←	Elementbreite verringern	Elementgröße ändern
Strg + ↑	Elementhöhe erhöhen	Elementgröße ändern
Strg + ↓	Elementhöhe verringern	Elementgröße ändern
Strg + Mausrad hoch	Zoom+	Arbeitsplatzgröße ändern
Strg + Mausrad runter	Zoom-	Arbeitsplatzgröße ändern

10 Programmbeispiele

Zwei Beispiele mit einfachen Aufgaben erklären die Erstellung eines Schaltprogramms in der ALP-Programmiersoftware.

10.1 Aufgabe 1: Lichtschalter mit automatischer Abschaltung

Die Aufgabe besteht darin, das Licht für eine bestimmte Zeit einzuschalten, z. B. für einen Hauseingang.

Aufgabenstellung:

1. Der Lichtsensor F1 und der Lichtschalter SB1 "TIME" sind vor der Eingangstür installiert.
2. Wenn der Taster SB1 kurz gedrückt wird und das Umgebungslicht nicht ausreicht, soll das Licht für 1 Minute eingeschaltet werden – diese Zeit sollte ausreichen, um das Schlüsselloch zu finden und die Tür zu öffnen.
3. Wenn der Taster SB1 für 2 Sekunden gedrückt wird, soll das Licht für 3 Minuten unabhängig von der Umgebungshelligkeit eingeschaltet werden – dieser Modus kann zum Beispiel für die Reinigung des Eingangs nützlich sein.
4. Zusätzlich ist die Möglichkeit vorzusehen, das Licht durch Befehle von externen Geräten oder mit dem Schalter SA1 "CONST" unabhängig von der Umgebungshelligkeit zu steuern. Dieser Modus kann während des Empfangs von Gästen oder für eine weitere Automatisierung der Wohnung als Teil eines "Smart-House"-Programms nützlich sein.
5. Es ist die Möglichkeit vorzusehen, das Licht nur zu einem bestimmten Zeitpunkt einzuschalten.

Geräteauswahl:

Das Steuergerät muss mindestens zwei digitale Eingänge, einen digitalen Ausgang und eine integrierte Echtzeituhr enthalten, um diese Aufgabe umzusetzen. Diese Merkmale besitzen Geräte der Serie PR110 mit den Buchstaben "RTC" in der Bezeichnung.

Die Task-Implementierung mit dem Gerät PR110-24.8D.4R-RTC:

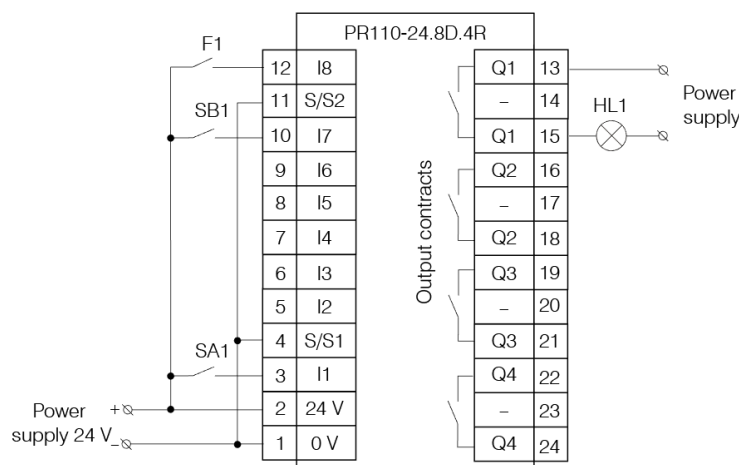


Abb. 10.1

Schaltprogramm

Das Schaltprogramm kann implementiert werden wie in Abb. 13.2 gezeigt.

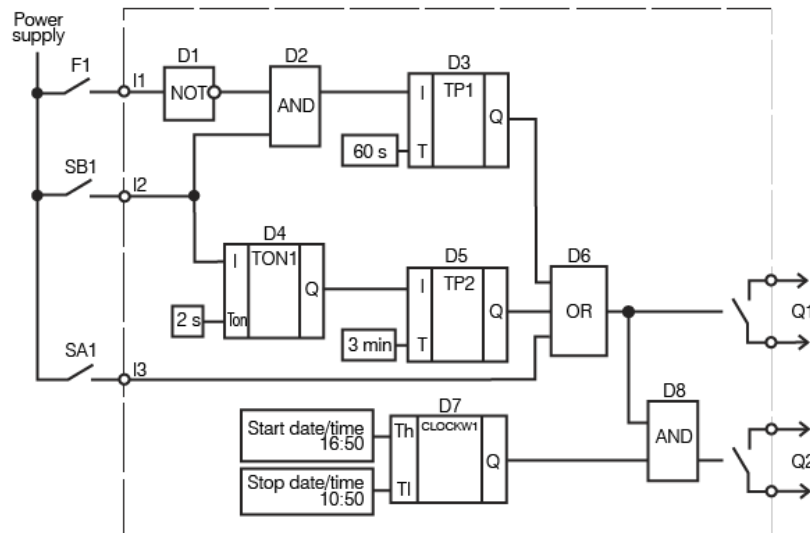


Abb. 10.2

Eingang I1 – verbunden mit dem Lichtsensor F1

Eingang I2 – verbunden mit dem Taster SB1

Eingang I3 – verbunden mit dem Schalter SA1

Ausgang Q1 – Ausgang zur Implementierung der Aufgabenpunkte 1-4

Ausgang Q2 – Ausgang zur Implementierung des Aufgabenpunktes 5

Programmbeschreibung:

1. Wenn der Taster SB1 kurz gedrückt wird (< 2 s), wird das logische UND (D2) aktiviert. Wenn die Umgebungshelligkeit unzureichend ist, ist der erste Eingang von D2 ebenfalls **Wahr** und der Timer TP "Pulse" (D3) gibt einen Impuls von 1 Minute Dauer. Dieser Impuls aktiviert den Ausgang Q1 über das logische ODER (D6) und das Licht wird für 1 Minute eingeschaltet.
2. Wenn der Taster SB1 für > 2 s gedrückt wird, aktiviert die Einschaltverzögerung TON (D4) den Timer TP "Impuls" (D5), ein Impuls mit der Dauer von 3 Minuten aktiviert den Ausgang Q1 über das logische ODER (D6) und das Licht wird für 3 Minuten eingeschaltet.
3. Wenn die Umgebungshelligkeit ausreicht, ist der Kontakt des Sensors F1 geschlossen, das logische UND (D2) gesperrt und der Timer TP "Pulse" (D3) gesperrt.
4. Wenn der Schalter SA1 "CONST" geschlossen ist, wird der Ausgang Q1 über das logische ODER (D6) aktiviert und das Licht ständig eingeschaltet.
5. Wenn Sie das Licht nur an bestimmten Wochentagen zu bestimmten Zeiten nutzen möchten, können Sie den Ausgang Q2 verwenden. Mit dem Wochenschaltuhr CLOCKW1 (D7) können Sie die Start- und Stoppzeit und die Wochentage für die Beleuchtung einstellen.

Das in ALP erstellte Schaltprogramm ist in Abb. 10.3 dargestellt.

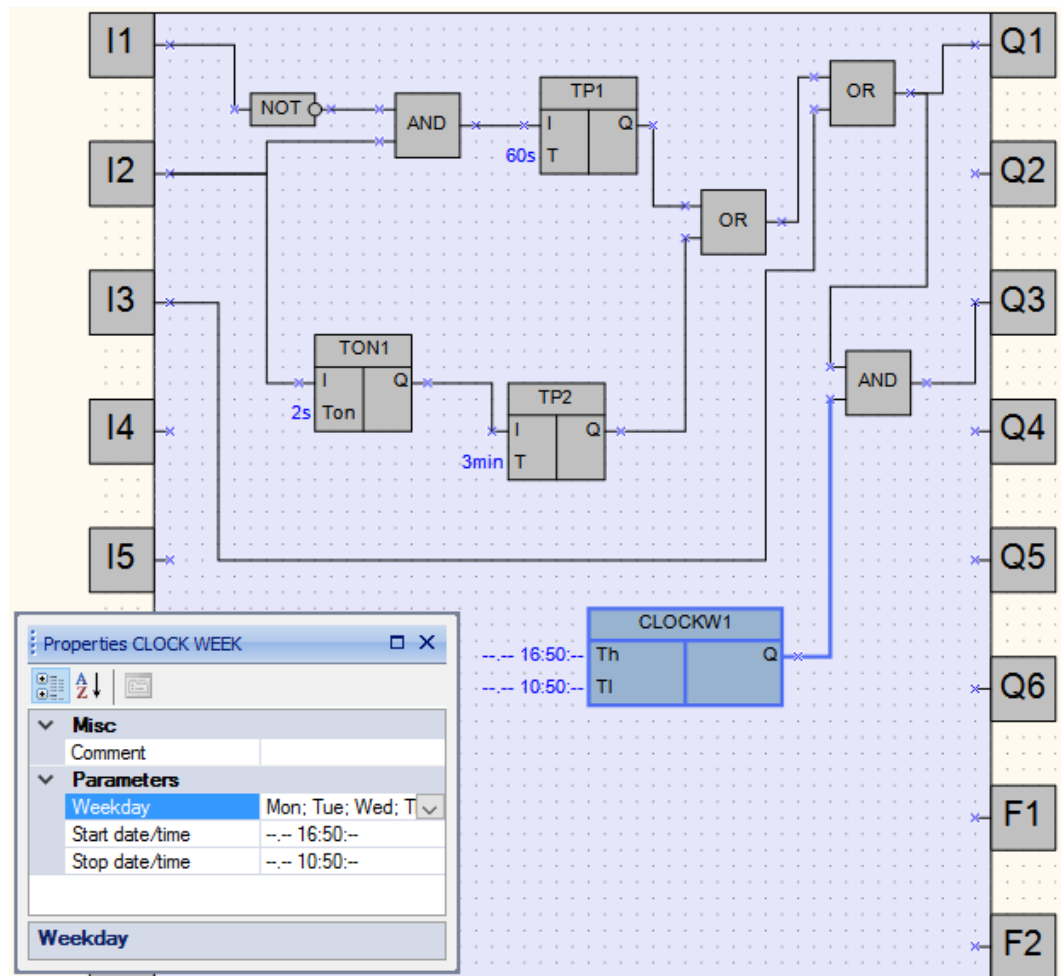


Abb. 10.3

10.2 Aufgabe 2: Steuerung eines Mischers

Die Aufgabe besteht darin, einen Industriemischer mit einfachen Steuerungsfunktionen zu implementieren.

Aufgabenstellung:

1. Es sind ein automatischer und ein manueller Betriebsmodus erforderlich. Der Schalter SA1 "MODE" wird installiert, um zwischen den Modi umzuschalten.
2. Im Automatikbetrieb kann der Betriebszyklus mit der Taste SB1 "START" gestartet werden und stoppt automatisch am Ende des Zyklus, oder kann manuell mit der Taste SB2 "STOP" gestoppt werden. Die Zyklusdauer beträgt 5 Minuten. Während des Zyklus ist der Motor des Mischers abwechselnd für 15 Sekunden eingeschaltet und für 10 Sekunden ausgeschaltet. Alle Einstellungen können im Programm geändert werden.
3. Im manuellen Modus kann der Motor mit der Taste SB1 "START" gestartet und mit der Taste SB2 "STOP" gestoppt werden.
4. Wenn der Motor überlastet ist (Überlastschalter F1), soll er automatisch abgeschaltet werden, ein unterbrochenes akustisches Warnsignal (HA1) mit einem 3-Sekunden-Intervall soll erzeugt und eine Betriebsstörung durch die Signalleuchte HL1 "Überlast" angezeigt werden.
5. Das akustische Signal kann mit der Taste SB3 "RESET" ausgeschaltet werden.
6. Die Taste SB4 "CONTROL" dient zur Funktionsprüfung der Lampe HL1 und des akustischen Signals HA1.

Programmbeispiele

Geräteauswahl:

Das Steuergerät muss mindestens 6 digitale Eingänge und 3 digitale Ausgänge haben, um diese Aufgabe zu realisieren. Diese Funktionen können von Geräten der Serie PR110 bereitgestellt werden.

Das Schaltbild in Abb. 10.4 stellt die Implementierung der Aufgabe auf dem Gerät PR110-24.8D.4R dar.

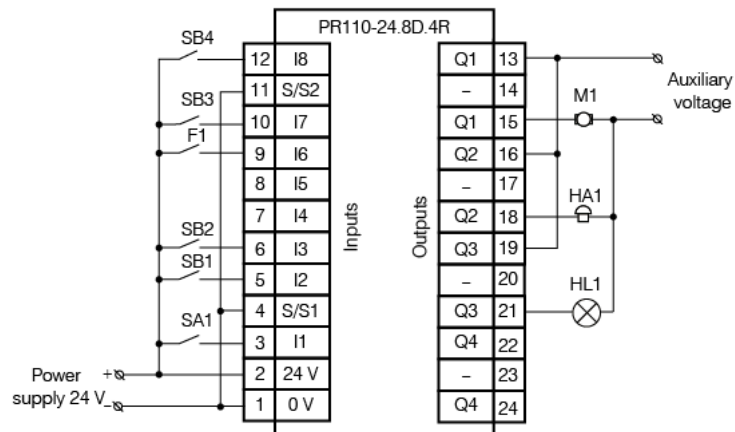


Abb. 10.4

Schaltprogramm

Das Schaltprogramm kann implementiert werden wie in Abb. 10.5 gezeigt.

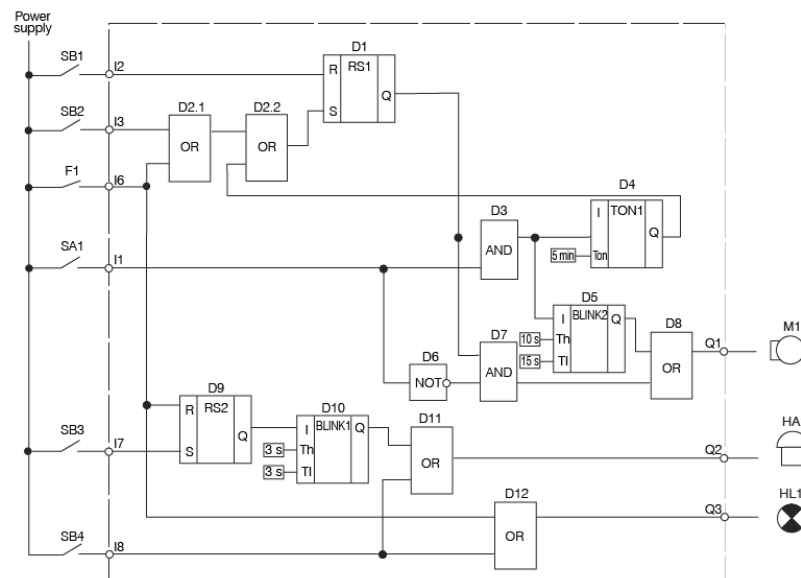


Abb. 10.5

Eingang I1 – verbunden mit dem Schalter SA1 "MODE"

Eingang I2 – verbunden mit der Taste SB1 "START"

Eingang I3 – verbunden mit der Taste SB2 "STOP"

Eingang I6 – verbunden mit dem Überlastschalter F1

Eingang I7 – verbunden mit der Taste SB3 "RESET"

Eingang I8 – verbunden mit der Taste SB4 "TEST"

Ausgang Q1 – verbunden mit dem Motor

Ausgang Q2 – verbunden mit dem akustischen Signal HA1

Ausgang Q3 – verbunden mit der Signalleuchte HL1

Programmbeispiele

Programmbeschreibung:

1. Eingang I2 (SB1 "START")

Wenn die Taste SB1 gedrückt wird, wird der RS-Trigger D1 auf **Wahr** gesetzt, solange kein Signal zum Zurücksetzen am Eingang R vorhanden ist. Der nachfolgende Signalweg hängt vom Zustand des Schalters SA1 "MODE" ab:

- Wenn SA1 offen ist (manueller Betrieb), sind das logische UND (D7) und das logische ODER (D8) freigegeben und der Motor M1 (Ausgang Q1) ist eingeschaltet.
- Wenn SA1 geschlossen ist (Automatikbetrieb), wird das logische UND (D7) deaktiviert und das Startsignal kann nur den Impulsgenerator BLINK (D5) zum Starten des Betriebszyklus (15 s an/10 s aus) und den Einschaltverzögerung TON (D4) zum Stoppen (in 5 Min) aktivieren.

2. Eingang I3 (SB2 "STOP")

Wird die Taste SB2 gedrückt oder ist der Schalter F1 aktiviert, wird der RS-Trigger D1 über den Eingang R zurückgesetzt und der Ausgang Q1 wird deaktiviert.

3. Eingang I1 (SA1 "MODE")

- Ist der Schalter SA1 offen (manueller Betrieb), wird das logische UND D3 deaktiviert und D7 freigegeben, der Timer D4 und der Impulsgenerator D5 werden deaktiviert und der Motor M1 kann nur mit SB1 gestartet und mit SB2 gestoppt werden.
- Ist der Schalter SA1 geschlossen (Automatikbetrieb), wird das logische UND D3 freigegeben und D7 deaktiviert, so dass der Motor M1 nur vom Impulsgenerator D5 (Zyklus: 15 Sek. an/10 Sek. aus) gestartet und vom Timer D4 nach 5 Minuten gestoppt werden kann.

4. Eingang I6 (Überlastschalter F1)

Wenn der Motor überlastet ist, wird der F1-Kontakt geschlossen, der RS-Trigger D1 wird zurückgesetzt und der Motor wird gestoppt.

Gleichzeitig wird die Signallampe HL1 über das logische ODER (D12) eingeschaltet und das akustische Signal HA1 wird über den RS-Trigger D9 aktiviert. Der Impulsgenerator D10 liefert ein unterbrochenes akustisches Signal mit dem Zyklus 3 Sek. an/3 Sek. aus.

5. I7-Eingang (SB3 "RESET")

Mit der Taste RESET wird das akustische Signal HA1 zurückgesetzt. Wenn die Taste SB3 gedrückt wird, wird der RS-Trigger D9 zurückgesetzt und der Impulsgenerator D10 für das akustische Signal HA1 wird gestoppt.

6. Eingabe I8 (SB4 "TEST")

Mit der Taste TEST werden das akustische Signal HA1 und die Signalleuchte HL1 geprüft. Wenn die Taste SB4 gedrückt wird, werden die logischen ODER D11 und D12 freigegeben, die Ausgänge Q2 und Q3 aktiviert und das akustische Signal und die Lampe eingeschaltet.

Das Schaltprogramm ist in Abb. 10.6 abgebildet.

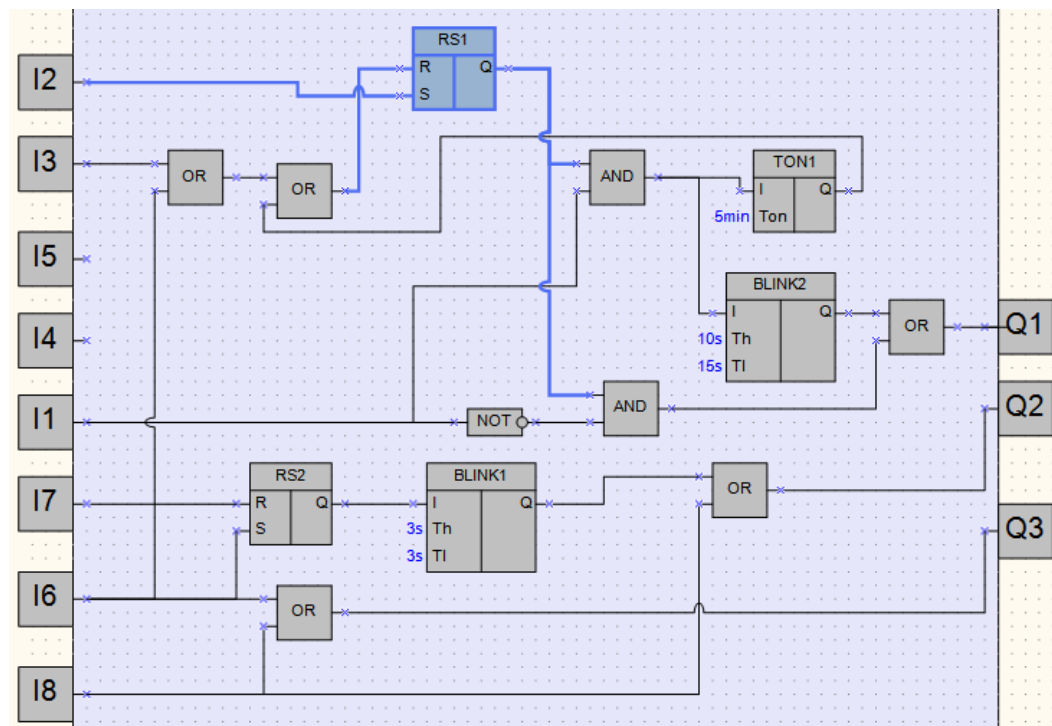


Abb. 10.6

Hinweise:

1. Die verbleibenden zwei unbenutzten Eingänge und ein Ausgang können für die Implementierung zusätzlicher Funktionen verwendet werden. Um beispielsweise zwischen verschiedenen Zeiteinstellungen für den automatischen Motorbetrieb umzuschalten oder auf andere Betriebsparameter des Mischers umzuschalten.
2. Der technologische Arbeitszyklus kann durch die Implementierung eines Schrittzählers (CT) zum Abschalten des RS-Triggers D1 vollständig automatisiert werden.