DE

akYtec ALP
Agile Logic Programming

akYtec ALP v3.1

Programmiersoftware

Benutzerhandbuch

akYtec ALP v3.1_3-DE-139699-2.5
© Alle Rechte vorbehalten.
Technische Änderungen und Druckfehler vorbehalten.

Inhaltsverzeichnis

1. Einfuehrung 5
1.1. Systemanforderungen 5
1.2. Begriffe und Abkuerzungen 5
2. Installation 7
3. Benutzeroberflaeche 10
3.1. Hauptmenue 10
3.2. Symbolleisten 12
3.3. Bibliothek 14
3.4. Eigenschaftenfenster 16
3.5. Variablenfenster 16
3.6. Arbeitsbereich 17
3.7. Statusleiste 18
3.8. Display-Manager 20
4. Allgemeine Informationen 22
4.1. Programmausfuehrung 22
4.2. Projekterstellung 22
4.2.1. Projekt mit Schluesseldatei 23
4.3. Programmentwicklung 25
4.3.1. Textfeld 27
4.3.2. Variablenbaustein 28
4.3.3. Konstanten-Baustein 32
4.3.4. Verzoegerungslinie 33
4.3.5. Netzwerkvariablen-Baustein 34
4.3.6. ReadFromFB-/WriteToFB-Bausteine 35
4.3.7. Konvertierungsbausteine 36
4.3.8. Elemente anordnen 37
4.3.9. Ausfuehrungsablauf 37
4.4. Display-Programmierung 38
4.4.1. LCD mit monochromem Text 40
4.4.2. Grafisches Farb-LCD 45
4.5. Simulation 52
4.5.1. Visualisierung-Simulation 57
4.5.2. ST-Code-Simulation 58
4.6. Verbindung zum Geraet 61
4.7. Projekt auf Geraet uebertragen 63
4.8. Online-Debugging 64
4.9. Projektinformationen 67
4.10. Komponenten-Manager 69
4.11. Makroentwicklung 72
4.12. Verwendung der ST-Funktion 81
4.13. ST-Funktionsbausteine 86
5. Geraetekonfiguration 90
5.1. Display 90
5.2. Uhr 91
5.3. Datenaustausch 92

5.3.1. Schnittstellen 93
5.3.2. Betrieb ueber Modbus-Protokoll 99
5.4. Erweiterungsmodule 117
5.5. Eingaenge und Ausgaenge 118
5.6. Passwort 119
5.7. Verbindung zur akYtec Cloud 120
6. Variablen 122
6.1. Datentypen 129
6.2. Servicevariablen 131
6.3. Netzwerkvariablen 131
6.4. Variablen mit Parametern verknuepfen 132
6.5. Variablenblock kopieren und einfuegen 133
7. Bibliothek 135
7.1. Funktionen 135
7.1.1. Logische Operatoren 135
7.1.2. Mathematische Operatoren 138
7.1.3. Relationale Operatoren 141
7.1.4. Bitshift Operatoren 142
7.1.5. Bitoperatoren 143
7.2. Funktionsbausteine 146
7.2.1. Trigger 146
7.2.2. Timer 148
7.2.3. Generatoren 153
7.2.4. Zaehler 154
7.2.5. Analog 158
7.3. Projektmakros 165
7.4. ST-Funktionen 166
7.5. ST-Funktionsblock 170
7.6. Display-Elemente 174
7.6.1. Text-Box 175
7.6.2. I/O-Box (INT/REAL) 176
7.6.3. I/O-Box (BOOL) 178
7.6.4. Dynamische Box 180
7.6.5. ComboBox 182
7.7. Einfache grafische Elemente 184
7.7.1. Text 191
7.7.2. Indikator 193
7.7.3. Fortschrittanzeige 194
7.7.4. Dynamische Box 195
7.7.5. REAL/INT Ein-/Ausgabe 197
7.7.6. Zeitangabe/-ausgabe 201
7.7.7. IP-Adresse 202
7.7.8. Linie 204
7.7.9. Vieleck 205
7.7.10. Kreis 206
7.7.11. Taste 207
7.7.12. Schalter 209

7.7.13. Schaltergruppe ... 211
7.7.14. Bild ... 214
7.7.15. Menue ... 216
7.7.16. Diagramm ... 218
8. Geraet ... 223
8.1. Geraeteinformationen ... 223
8.2. Zykluszeit ... 224
8.3. Firmware-Update/-Reparatur ... 224
8.4. Kalibrierung ... 225
8.4.1. Eingangskalibrierung ... 226
8.4.2. Ausgangskalibrierung ... 227
8.5. Zielgeraet aendern ... 227
9. Plugins ... 229
9.1. Replikationsmaster ... 230
9.1.1. Programmuebertragung ueber Replikationsmaster unter Windows ... 235
9.1.2. Programmuebertragung ueber Replikationsmaster unter Linux ... 239
9.2. Exportieren des Geraets in die akYtec Cloud ... 240
10. Tastaturkuerzel ... 242
11. Programmbeispiele ... 244
11.1. Aufgabe 1: Lichtschalter mit Abschaltautomatik ... 244
11.2. Aufgabe 2: Mixersteuerung ... 246
11.3. Aufgabe 3: Direkte Verbindung eines PR103 zur akYtec Cloud ... 249
12. ST-Sprache ... 253
12.1. Syntax ... 253
12.1.1. Bitweiser Zugriff auf Variablen ... 253
12.1.2. ST-Funktionen in anderen ST-Programmelementen verwenden ... 253
12.1.3. Verwendung eines Funktionsbausteins in einem anderen Funktionsbaustein ... 254
12.1.4. Kommentare im ST-Editor ... 255
12.1.5. ST-Elemente zwischen Projekten kopieren ... 255
12.2. Dokumentation im ST-Editor ... 256
12.3. Datentypen ... 257
12.3.1. Reservierte Schluesselwoerter ... 258
12.3.2. Arrays ... 259
12.4. Sprachstrukturen ... 261
12.4.1. Operationen ... 262
12.4.2. Zuweisungsvorgang ... 264
12.4.3. IF-Anweisung ... 264
12.4.4. CASE-Anweisung ... 265
12.4.5. RETURN-Anweisung ... 266
12.4.6. FOR-Anweisung ... 267
12.4.7. WHILE-Anweisung ... 268
12.4.8. REPEAT UNTIL-Anweisung ... 269
12.5. Systemfunktionen ... 269
12.5.1. Datums- und Uhrzeitfunktionen ... 269
12.5.2. Mathematische Funktionen ... 271
12.5.3. Logarithmische Funktionen ... 273

12.6. Systemfunktionsbausteine 276
12.6.1. Trigger 276
12.6.2. Timer 280
12.6.3. Generatoren 287
12.6.4. Zaehler 288

1 Einfuehrung

ALP ist die Programmiersoftware für programmierbare Geräte der akYtec GmbH. Die Programmiersprache ist die grafische Sprache FBD (Function Block Diagram) und ST (Structure Language) nach IEC 61131-3. Die Software ermöglicht die Simulation und das Debuggen des erstellten Programms sowie dessen Hochladen in den nichtflüchtigen Speicher des Geräts.
Das erstellte Projekt enthält mindestens ein Schaltprogramm und eine Gerätekonfiguration.
Der erste Arbeitsbereich enthält das Hauptschaltprogramm. Makros können als Schaltprogramme in separaten Arbeitsbereichen erstellt werden.
Wenn das Zielgerät über ein Display verfügt, kann dieses mithilfe von Displayformen in separaten Arbeitsbereichen programmiert werden.
In ALP kann immer nur ein Projekt geöffnet werden.
Die akYtec ALP-Funktionalität unterscheidet sich für folgende Gerätegruppen:

– Geräte auf der ursprünglichen Hardwareplattform (PR100, PR102, PR200 und SMI200)
– Geräte der zweiten Generation (PR103, PR205 und neuer)

Die grundlegende akYtec ALP-Funktionalität ist für alle Geräte verfügbar, Funktionen und Schnittstellen für Geräte der zweiten Generation sind für ältere Geräte nicht verfügbar.

1.1 Systemanforderungen

Betriebssysteme:
– Windows 7 (SP1+)
– Windows 8.1
– Windows 10
– Windows 11

Systembibliotheken:
– Microsoft .NET Framework 4.8
– Microsoft .NET Desktop Runtime 6.0.8
– Microsoft Visual C++ 2015-2022

Empfohlene Hardwareanforderungen:
– 3,2 GHz Prozessor
– 4 GB RAM
– 700 MB freier Festplattenspeicher
– Freier USB-Anschluss
– Tastatur und Maus
– Bildschirmauflösung 1280×800

Eine Internetverbindung ist erforderlich für:
– Softwareaktualisierung
– Geräte-Firmware-Update
– Herunterladen von Makros im Komponenten-Manager

1.2 Begriffe und Abkuerzungen

– **ALP** – Programmiersoftware für programmierbare Geräte der akYtec GmbH
– **Arbeitsbereich** – Grafischer Bereich in der Benutzeroberfläche zum Erstellen, Ändern und Debuggen von Programmen
– **EEPROM** – Elektrisch löschbarer programmierbarer Nur-Lese-Speicher des Geräts
– **FBD (Function Block Diagram)** – Grafische Programmiersprache, die von IEC 61131-3 unterstützt wird
– **Funktion** – Strukturelle Programmeinheit mit einem Rückgabewert. Die Funktion speichert keine Informationen über ihren internen Zustand, d. h. wenn die Funktion mit denselben Eingabewerten aufgerufen wird, gibt sie denselben Ausgabewert zurück
– **Funktionsblock** – Strukturelle Programmeinheit mit internem Speicher und einem oder mehreren Ausgabewerten. Sie wird im Programm als Instanz verwendet, d.h. als Kopie mit eigenem Speicher
– **LoRa** – Funktechnologie für Datenübertragungen

- **LoRaWAN (Long Range Wide Area Network)** – Das Protokoll, das die Kommunikation über das LoRa-Netzwerk verwaltet
- **Makro** – Benutzerfunktionsblock
- **Programmzyklus** – Ausführungszeit des Schaltprogramms, die von seiner Komplexität abhängt
- **Projekt** – Mit ALP-Software für ein bestimmtes Gerät erstellte Benutzeranwendung, enthält das Schaltprogramm
- **Slot** – Ein Steckplatz im Gerät, der den Anschluss einer Schnittstelle oder eines Erweiterungsmoduls ermöglicht
- **ST (Structured Text)** – Eine Textprogrammiersprache, die von IEC 611131-3 unterstützt wird
- **Zielgerät** – Gerätetyp, für den das Projekt erstellt wird

2 Installation

1. Laden Sie die Datei *akYtecALP.exe* herunter und führen Sie sie auf Ihrem PC aus.
2. Wählen Sie die Installationssprache aus.

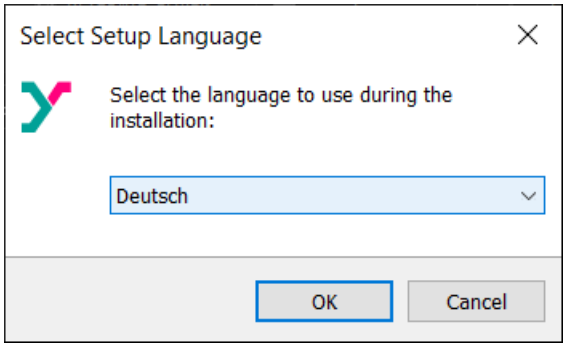


Abb. 2.1 Auswahl der Installationssprache

GEFAHR
Einschränkungen: ALP muss unter demselben Benutzerkonto installiert und ausgeführt werden.
Während der Installation erstellt ALP im AppData-Verzeichnis (zum Beispiel *C:\Users \user\AppData\Roaming\akYtec\akYtec ALP*) einen Ordner, der die für den ordnungsgemäßen Betrieb des Programms erforderlichen Dateien enthält.
Dieses Verzeichnis ist nur für den Benutzer zugänglich, unter dessen Konto die Installation durchgeführt wurde.
Dieses Verhalten ist eine Einschränkung des Windows-Betriebssystems.

3. Das Fenster des Installationsassistenten wird geöffnet. Prüfen Sie die Informationen und klicken Sie auf die Schaltfläche **Weiter**.

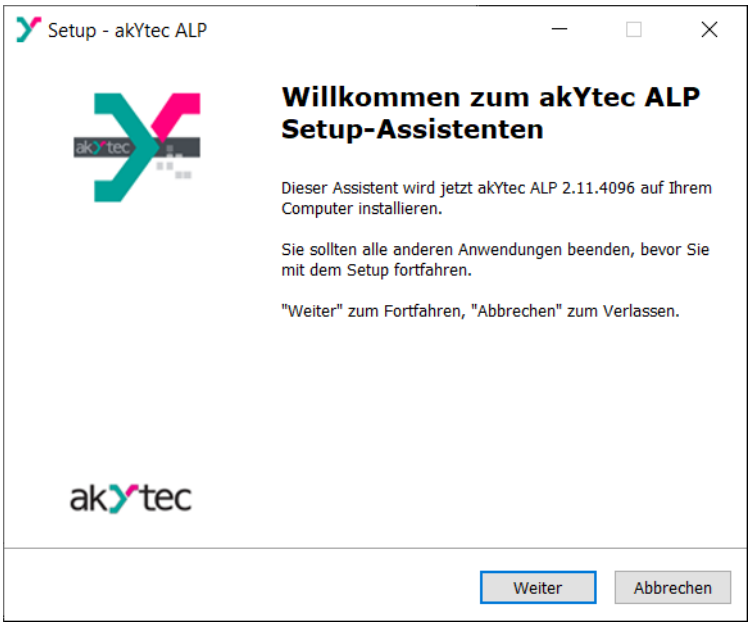


Abb. 2.2 Fenster des Installationsassistenten

4. Wählen Sie den Installationsordner für akYtec ALP und den Ordner für die Startmenü-Verknüpfung. Klicken Sie auf die Schaltfläche **Weiter**.
5. Aktivieren Sie bei Bedarf die entsprechenden Kontrollkästchen, um eine Desktop-Verknüpfung zu erstellen, eine Verknüpfung in der Schnellstartleiste zu erstellen, Dateien mit den Erweiterungen .owl und .owle mit akYtec ALP zu verknüpfen und die .NET6 SDK-Bibliothek zur

Erstellung eines Replikations-Assistenten für Linux zu installieren. Klicken Sie auf die Schaltfläche **Weiter**.

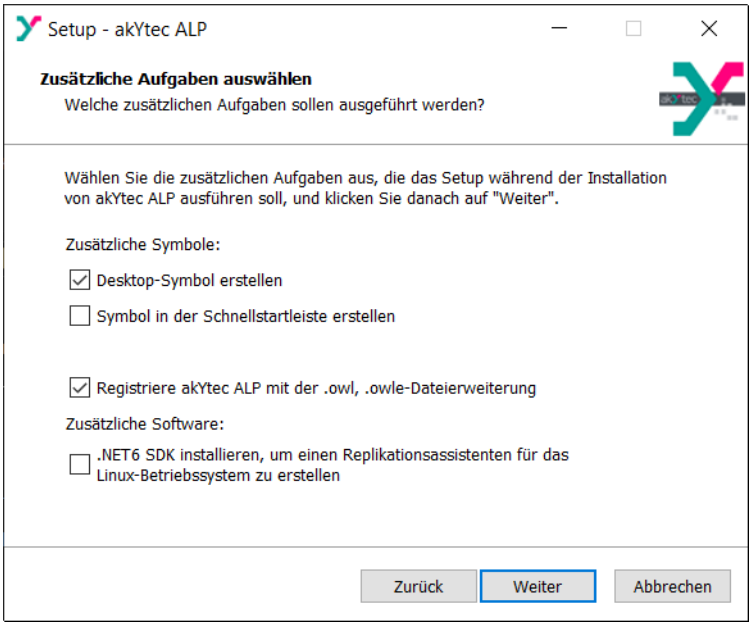


Abb. 2.3 Auswahl zusätzlicher Aufgaben

6. Prüfen Sie die Installationsinformationen und klicken Sie auf die Schaltfläche **Installieren**.

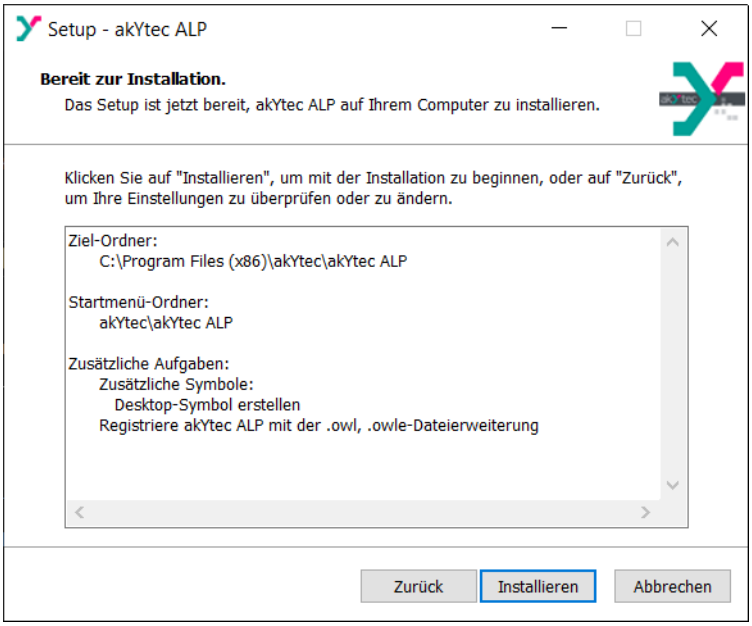


Abb. 2.4 Fenster zum Start der Installation

7. Es öffnet sich ein Fenster, das den Fortschritt der Installation anzeigt.

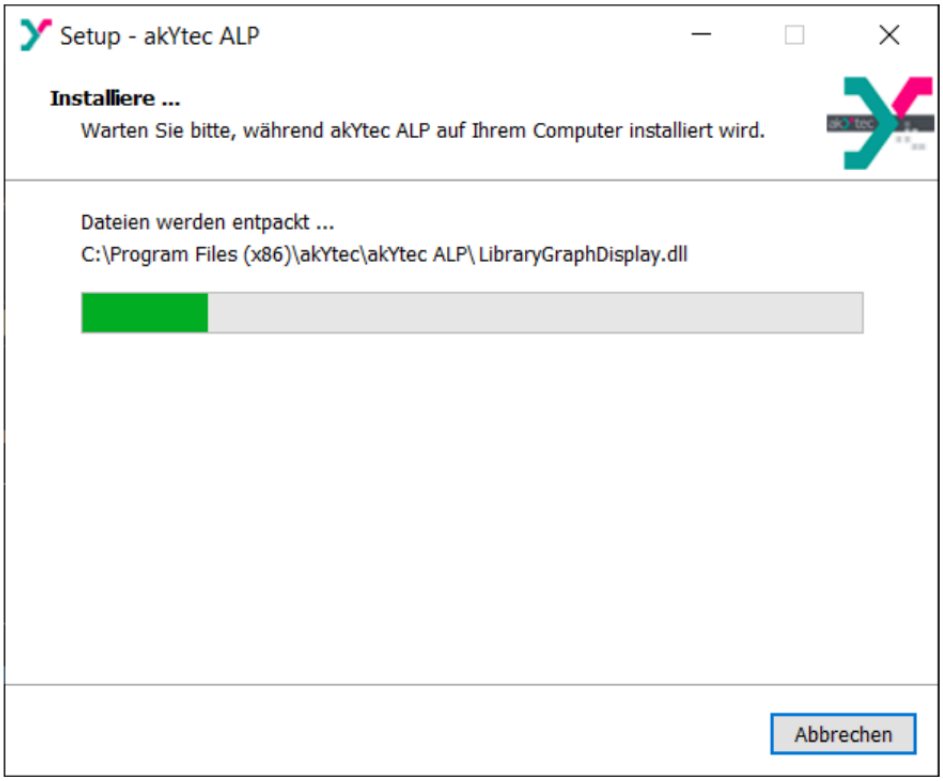


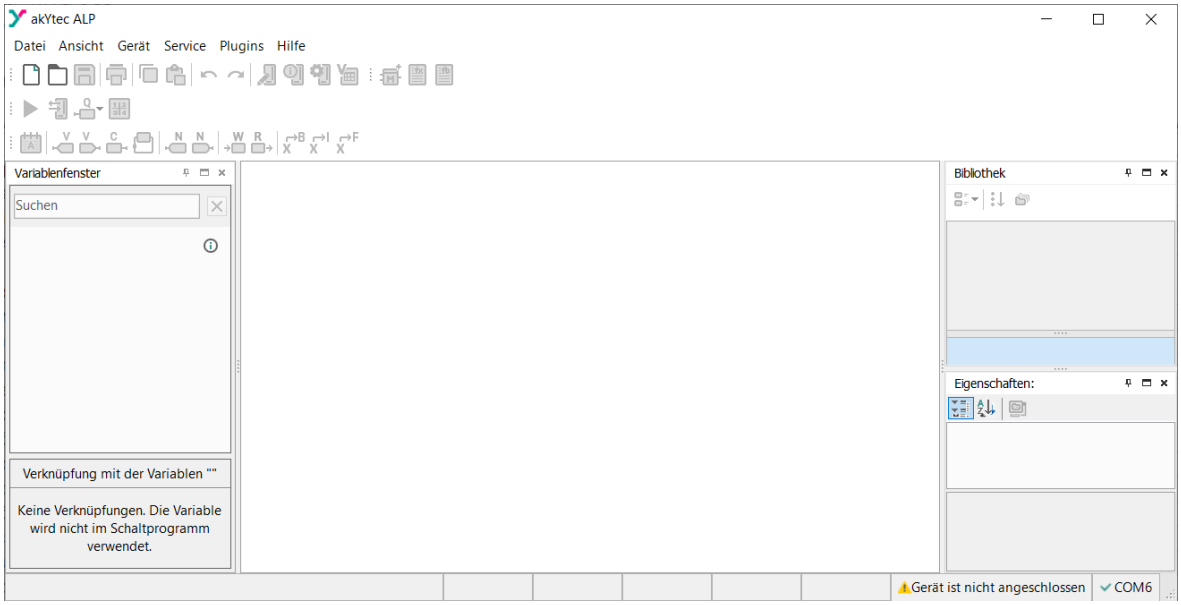
Abb. 2.5 Programminstallationsvorgang

8. Warten Sie, bis die Installation abgeschlossen ist, und aktivieren Sie bei Bedarf das Kontrollkästchen **akYtec ALP starten**.

Abb. 2.6 Abschluss der Installation

9. Klicken Sie auf die Schaltfläche **Fertigstellen**.

3 Benutzeroberflaeche



1. ***Titelleiste*** – Zeigt den Namen der Software und den Pfad zur geöffneten Projektdatei
2. ***Menüleiste*** – Besteht aus den folgenden Gruppen: Datei, Ansicht, Gerät, Service, Plugins und Hilfe
3. ***Symbolleisten*** – Standard, Service und Einfügen: Schneller Zugriff auf die wesentlichen Funktionen von ALP
4. ***Bibliothek*** – Das Panel zeigt alle Funktionen, Funktionsblöcke und Makros, die im Projekt verwendet werden können (per Drag & Drop)
5. ***Eigenschaftenfenster*** – Das Fenster, in dem die Eigenschaften des ausgewählten Projektelements angezeigt und geändert werden können
6. ***Arbeitsbereich*** – Der grafische Bereich, in dem Schaltprogramme, ST-Programme, Display-Elemente oder Displayformen erstellt und geändert werden können
7. ***Statusleiste*** – Zeigt Status- und Fehlermeldungen, Speichernutzung des Zielgeräts, Status des angeschlossenen Geräts und der Programmierschnittstelle
8. ***Display-Manager*** – Ein Tool zum Programmieren der Displayinformationen (nur für Geräte mit Display verfügbar)
9. ***Variablenfenster*** – Das Panel zeigt alle Projektvariablen mit ihren Parametern und Referenzen (per Drag-and-Drop)
10. ***Komponenten-Manager*** – Spezielles Tool in einem separaten Fenster für den Zugriff auf die Online-Datenbank und zum Hinzufügen der Elemente aus der Online-Datenbank zur Offline-Bibliothek oder zur Projektbibliothek. Internetverbindung erforderlich

3.1 Hauptmenue

Datei

Neues Projekt	Erstellt ein neues Projekt. Das aktuelle Projekt wird geschlossen.
Zielgerät ändern	Ändert das Zielgerät des Projekts.
Projekt öffnen	Öffnet ein zuvor gespeichertes Projekt.
Aktiven Arbeitsbereich speichern	Speichert den aktiven Arbeitsbereich.
Projekt speichern	Speichert das aktuelle Projekt.

Projekt speichern unter…	Erstellt eine Kopie des Projekts in einem anderen Ordner oder unter einem anderen Namen.
Schlüsseldatei erstellen…	Erstellt eine Schlüsseldatei, um das Projekt vor unbefugtem Zugriff zu schützen (in Entwicklung).
Projektinformationen	Zeigt die *Projektinformationen* an und ermöglicht deren Bearbeitung.
Neues Makro	Öffnet ein neues Makro in einem separaten *Arbeitsbereich*.
Importieren	Importiert ein Makro, eine ST-Funktion oder einen ST-Funktionsbaustein aus einer Datei in die Projektbibliothek.
Exportieren	Speichert das aktuelle Makro, die ST-Funktion oder den ST-Funktionsblock als Datei.
Komponenten-Manager	Öffnet den *Komponenten-Manager*.
Drucken	Öffnet den Dialog zum Einstellen der Druckoptionen und zum Drucken des aktiven Arbeitsbereichs.
Zuletzt geöffnet	Zeigt eine Liste der kürzlich geöffneten Projekte an.
Beenden	Schließt ALP.

Ansicht

Rückgängig	Macht die letzte Aktion rückgängig.
Wiederherstellen	Stellt die zuletzt rückgängig gemachte Aktion wieder her.
Statusanzeigen	hinzufügen/entfernen Fügt Indikatoren zur bzw. aus der *Statusleiste* hinzu oder entfernt sie.
Bibliothek	Blendet die *Bibliothek* ein oder aus.
Eigenschaftenfenster	Blendet das *Eigenschaftenfenster* ein oder aus.
Variablenfenster	Blendet das *Variablenfenster* ein oder aus.
Display-Manager	*Display-Manager* anzeigen/ausblenden
Layout zurücksetzen	Stellt das Standard-Panel-Layout wieder her.

Gerät

Programm auf Gerät übertragen	Lädt das aktuelle Projekt in den Gerätespeicher hoch.
Firmware-Update	Führt ein Firmware-Update des Geräts durch.
Geräteninformationen	Zeigt *Informationen* über die Software, das Zielgerät und das angeschlossene Gerät an.
Variablentabelle	Zeigt *editierbare Tabelle* der Projektvariablen mit ihren Parametern an.
Kalibrierung	Startet die analoge E/A-*Kalibrierung*.
Gerätenkonfiguration	Öffnet die *Gerätekonfiguration*.
Porteinstellungen	Zeigt die Einstellungen der *Programmierschnittstelle* an.

Service

Elemente anordnen	Ordnet Funktionsblöcke gleichen Typs im Arbeitsbereich automatisch von oben nach unten und von rechts nach links neu an.
Simulation	Startet oder stoppt die *Simulation*.
Sprache	Ermöglicht die Auswahl der Sprache der Benutzeroberfläche.
OFFLINE-Modus	Aktiviert den OFFLINE-Modus.

Hilfe

Automatische Update-Prüfung	Führt die Update-Prüfung beim Programmstart aus, wenn die Funktion aktiviert ist.
Auf Updates prüfen...	Sucht nach Softwareupdates.
Hilfe	Öffnet das Hilfefenster.
Versionsgeschichte	Zeigt eine laufende Liste der Änderungen an, die in allen Softwareversionen vorgenommen wurden.
Über die Software	Zeigt Informationen zur aktuellen Softwareversion an.

3.2 Symbolleisten

Standard

	Neues Projekt	Erstellt ein neues Projekt. Das aktuell geöffnete Projekt wird geschlossen.
	Projekt öffnen	Öffnet ein zuvor gespeichertes Projekt.
	Projekt speichern	Speichert das aktuelle Projekt.
	Drucken	Öffnet den Dialog zum Einstellen der Druckoptionen und zum Drucken des aktiven Arbeitsbereichs.
	Kopieren	Kopiert das ausgewählte Element.
	Einfügen	Fügt das kopierte Element ein.
	Rückgängig	Macht die letzte Aktion rückgängig.
	Wiederherstellen	Stellt die zuletzt rückgängig gemachte Aktion wieder her.
	Programm auf Gerät übertragen	Lädt das aktuelle Projekt in den Gerätespeicher hoch.
	Geräteninformationen	Zeigt Informationen über die Software, das Zielgerät und das angeschlossene Gerät an.
	Gerätenkonfiguration	Öffnet die Gerätekonfiguration.
	Variablentabelle	Zeigt die Tabelle der Projektvariablen an.

Service

	Simulation	Startet oder stoppt die Simulation.
	Online-Debugging	Startet oder stoppt das Online-Debugging.
	Ausführungsablauf	Ändert die Ausführungsreihenfolge der Ausgänge oder Verzögerungsleitungen in einem Schaltprogramm oder Makro.
	Elemente anordnen	Ordnet Funktionsblöcke gleichen Typs im Arbeitsbereich automatisch von oben nach unten und von rechts nach links neu an.

Einfügen

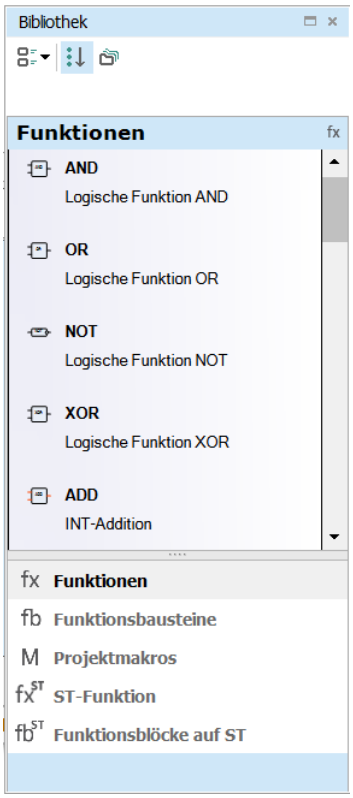
	Textfeld	Fügt ein Textfeld zum Kommentieren des Programms ein.
	Variablen-Ausgangsbaustein	Fügt einen Variablen-Ausgangsbaustein ein, dessen Wert im Programm geschrieben werden kann.
	Variablen-Eingangsbaustein	Fügt einen Variablen-Eingangsbaustein ein, dessen Wert im Programm gelesen werden kann.
	Konstanten-Baustein	Fügt einen Konstanten-Baustein mit festem Wert ein.
	Verzögerungslinien	Fügt Verzögerungslinien für eine Rückmeldung mit einer Verzögerung von einem Zyklus ein.
	Netzwerkvariablen-Ausgangsbaustein	Fügt einen Netzwerkvariablen-Ausgangsbaustein ein, dessen Wert über das Netzwerk geschrieben werden kann.
	Netzwerkvariablen-Eingangsbaustein	Fügt einen Netzwerkvariablen-Eingangsbaustein ein, dessen Wert über das Netzwerk gelesen werden kann.
	Baustein WriteToFB	Verbindet den Ausgangswert eines Blocks mit einem Parameter eines Funktionsblocks und liest diesen aus.
	Baustein ReadFromFB	Verbindet den Ausgangswert des Blocks mit dem ausgewählten Parameter des ausgewählten Funktionsblocks und dient zum Lesen des Parameters.
	Konvertierung in BOOL	Wandelt beliebige Werte in einen BOOL-Wert um.
	Konvertierung in INT	Wandelt beliebige Werte in einen INT-Wert um.
	Konvertierung in REAL	Wandelt beliebige Werte in einen REAL-Wert um.
	Neues Makro	Erstellt ein neues Benutzermakro.
	Neue ST-Funktion	Erstellt eine neue Benutzerfunktion in der ST-Sprache.
	Neuer ST-Funktionsblock	Erstellt einen neuen Benutzerfunktionsbaustein in der ST-Sprache.

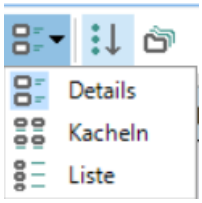
3.3 Bibliothek

Das Panel ***Bibliothek*** ist eine Übersicht der im Projekt verfügbaren Programmblöcke. Das Panel besteht aus Bibliotheken:

– *Funktionen*
– *Funktionsbausteine*
– *Projektmarkos*
– *ST-Funktionen*
– *Funktionsbausteine auf ST*

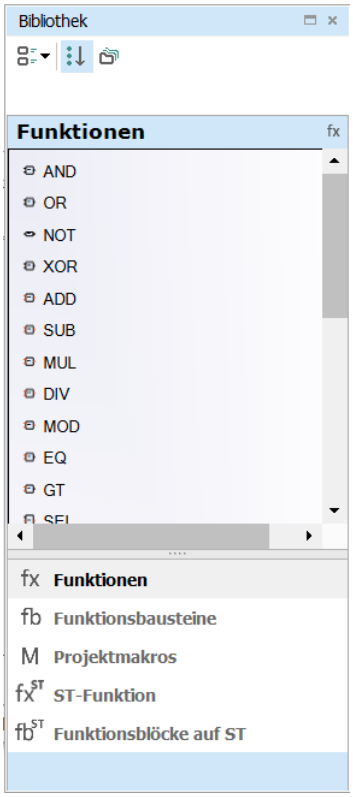
Wählen Sie ein Element in der unteren Symbolleiste des Bedienfelds aus, um den entsprechenden Inhalt anzuzeigen.
Die Standardposition des Panels ist die obere rechte Fensterecke (kann geändert werden).



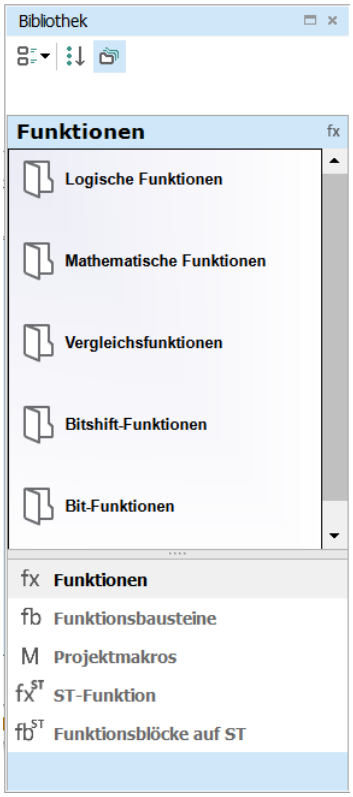


Die Panelansicht kann über die Symbole in der oberen Symbolleiste geändert werden.

– Klicken Sie auf das Symbol **Alle zeigen**, um alle Blöcke der ausgewählten Bibliothek anzuzeigen.



– Klicken Sie auf das Symbol **In Gruppen zeigen**, um die Blöcke der ausgewählten Bibliotheksgruppe anzuzeigen. Doppelklicken Sie auf den Ordner, um ihn zu öffnen.



Eine Beschreibung der Bibliotheksgruppen und einzelnen Blöcke finden Sie im Abschnitt *Bibliothek*.

3.4 Eigenschaftenfenster

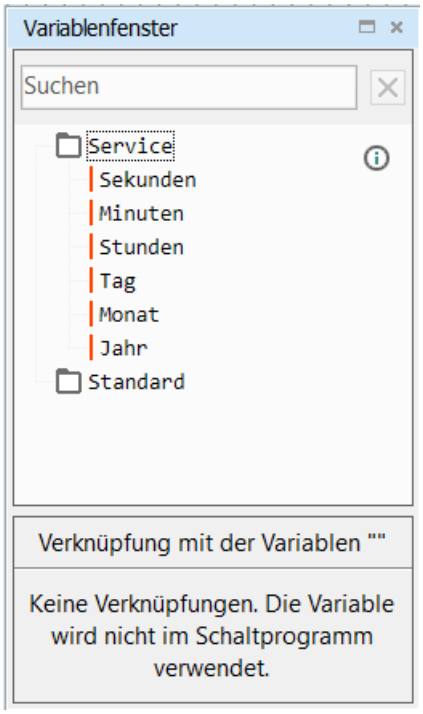
Das Panel **Eigenschaftenfenster** wird zum Anzeigen und Ändern der Parameter der Programmelemente verwendet. Wählen Sie das Element aus, um seine Eigenschaften anzuzeigen. Wenn kein Element ausgewählt ist, werden im Bedienfeld die Eigenschaften des Arbeitsbereichs angezeigt.
Die Standardposition des Panels ist die untere rechte Fensterecke (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 Parametereingabefeld aus, um seinen Wert zu bearbeiten.

3.5 Variablenfenster

Das Panel ***Variablenfenster*** zeigt die Projektvariablen aus der Variablentabelle.
Die Standardposition des Panels befindet sich in der oberen linken Fensterecke und kann geändert werden.
In einem Tooltip-Text können Sie sich die Informationen zur Variable ansehen.

Variablenblock im Arbeitsbereich

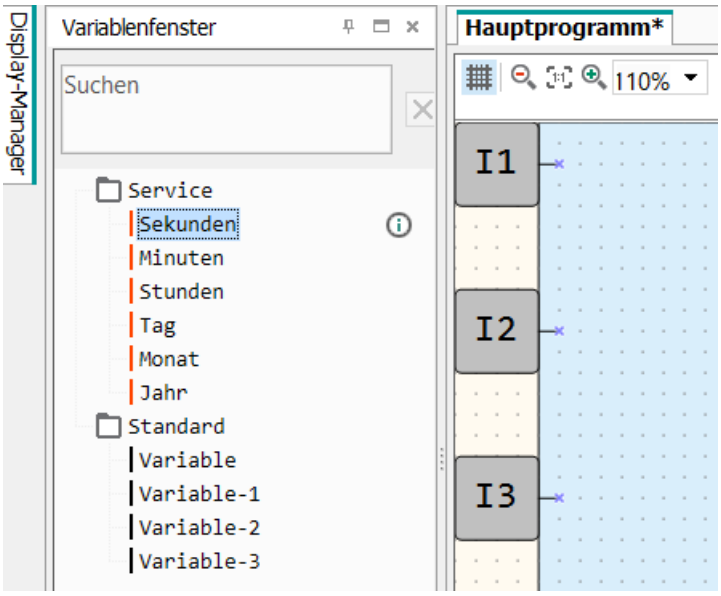
Ziehen Sie eine Variable per Drag & Drop, um sie als Eingangsbaustein in das Schaltprogramm einzufügen.



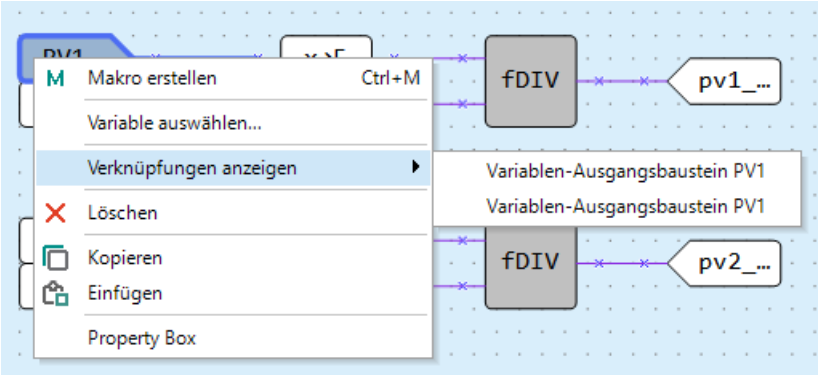
Um eine Variable als Ausgabeblock zu verwenden, halten Sie die Umschalttaste gedrückt, während Sie die Variable ziehen und ablegen.
Wird eine Variable per Drag & Drop auf einen Anschlusspin eines Blocks gezogen, wird der erstellte Variablenblock mit diesem Anschlusspin verbunden.

Verknüpfungen

Die Variablenreferenzen werden im unteren Panelbereich als Links angezeigt. Wenn Sie auf den Link klicken, wird der Block, auf den die Variable verweist, im Arbeitsbereich hervorgehoben.

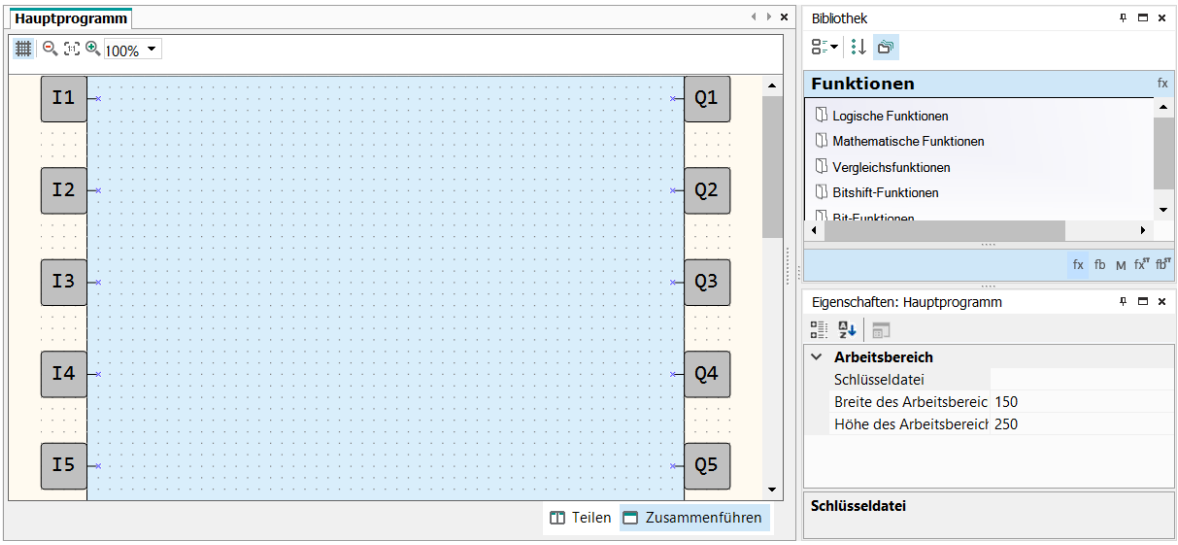


Wird eine Variable an mehreren Stellen im Projekt verwendet, können alle Verweise über den Eintrag ***Verknüpfungen anzeigen*** im Kontextmenü des Variablenblocks angezeigt werden. Klicken Sie auf den Link, um die Referenz anzuzeigen.



3.6 Arbeitsbereich

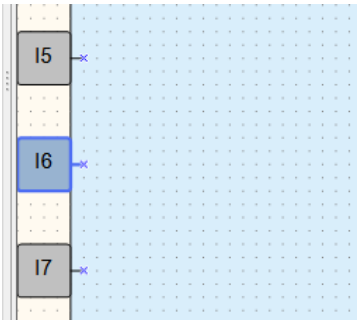
Bei geöffnetem Projekt wird in der Mitte des Fensters der Arbeitsbereich mit der Registerkarte ***Hauptprogramm*** angezeigt.
Wenn Ihr Gerät ST-Funktionen unterstützt, wird die Registerkarte **Funktionseditor** angezeigt.



Das Schaltprogramm wird aufgebaut, indem man Programmblöcke im Arbeitsbereich platziert und Linien zwischen ihnen verbindet. Die Größe des Arbeitsbereichs kann in der Eigenschaftsbox geändert werden. Die Eingänge (links) und Ausgänge (rechts) sind wie folgt beschriftet:

– **Ix** – Digitaleingänge
– **AIx** – Analogeingänge
– **Qx** – Relaisausgänge
– **AOx** – Analogausgänge
– **Fx** – LED-Anzeigen

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



Arbeitsbereich-Symbolleiste

	Raster anzeigen/ vebergen	Blendet vertikale und horizontale Lineale sowie ein Raster im Arbeitsbereich ein oder aus. Ist das Raster sichtbar, werden Blöcke und Verbindungslinien daran ausgerichtet.
	Zoom –	Verkleinert den Arbeitsbereich um 10 %.
	Originalgröße	Setzt den Arbeitsbereich auf die Originalgröße (100 %) zurück.
	Zoom +	Vergrößert den Arbeitsbereich um 10 %.

Den gewünschten Maßstab können Sie über das Dropdown-Menü rechts neben den oben beschriebenen Schaltflächen einstellen.

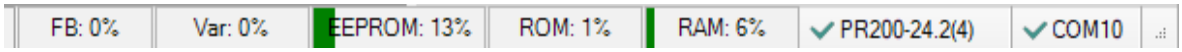
Die Symbole Zusammenführen und Teilen befinden sich in einer Symbolleiste unterhalb des Arbeitsbereichs. Verwenden Sie das Symbol Zusammenführen, um dasselbe Schaltprogramm in zwei Arbeitsbereichen anzuzeigen. Dies kann nützlich sein, wenn das Programm zu groß ist und Sie zwei verschiedene Teile des Programms gleichzeitig anzeigen möchten. Verwenden Sie das Symbol Teilen, um einen Arbeitsbereich anzuzeigen.

3.7 Statusleiste

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

Ansicht

Im rechten Feld der Statusleiste befinden sich verschiedene Statusanzeigen, die Informationen über die Speichernutzung des Zielgeräts (Ressourcenanzeigen), den Status des angeschlossenen Geräts und die Programmierschnittstelle anzeigen. Welche Anzeigen in der Statusleiste verfügbar sind, hängt vom Typ des Zielgeräts ab.



Ressourcenindikatoren zeigen die verbrauchte Ressource in Prozent der insgesamt verfügbaren Menge an. Bewegen Sie den Mauszeiger über den Indikator, um die absolute Menge der Ressource anzuzeigen.

Indikatoren

Wenn das Gerät verbunden ist, zeigt die Statusleiste folgende Informationen an:

- **FB**: Zeigt die Anzahl der verwendeten und verfügbaren Funktionsbausteine an.
- **Var**: Zeigt die Anzahl der verwendeten und verfügbaren Variablen an.
- **Stack**: Zeigt den Prozentsatz des verwendeten und verfügbaren Stack-Speichers an. Der Stack wird für Zwischenberechnungen im Programm verwendet.
- **Sys EEPROM**: Zeigt den Prozentsatz des verwendeten und verfügbaren systemeigenen nichtflüchtigen Speichers an. Der Indikator ist voll, wenn das Projekt Netzwerkvariablen verwendet, Variablen mit Visualisierungs- oder Geräteparametern verknüpft sind oder Erweiterungsmodule verwendet werden.
- **EEPROM**: Zeigt den Prozentsatz des verwendeten und verfügbaren nichtflüchtigen Speichers an. Der Indikator ist voll, wenn das Projekt Standard-Nichtflüchtigvariablen verwendet.
- **ROM**: Zeigt den verwendeten und verfügbaren ROM-Speicher an.
- **Sys RAM**: Zeigt den verwendeten und verfügbaren systemeigenen RAM-Speicher an. Dieser Indikator erscheint, wenn der RAM zu mehr als 80 % belegt ist.

HINWEIS
Die ALP-Software berechnet automatisch die verfügbaren Ressourcen des Geräts und zeigt eine Warnung an, sobald ein kritischer Wert erreicht wird.

- **RAM**: Zeigt den verwendeten und verfügbaren RAM-Speicher an.
- **System Visualisierungs-ROM**: Zeigt den verfügbaren Visualisierungs-ROM des Geräts in Prozent der Gesamtkapazität an (nur für Geräte mit grafischem Farb-LCD).
- **Benutzer Visualisierungs-ROM**: Zeigt den verfügbaren Visualisierungs-ROM für Bitmap-Bilder an (nur für Geräte mit grafischem Farb-LCD).
- **Visualisierungs-RAM**: Zeigt den verfügbaren Visualisierungs-RAM des Geräts in Prozent der Gesamtkapazität an. Die Berechnung erfolgt für den am stärksten belasteten Bildschirm (nur für Geräte mit grafischem Farb-LCD).
- **Gerät**: Zeigt das Gerätemodell an, sofern eine Verbindung besteht.

HINWEIS
*Klicken Sie auf den Gerät-Indikator, um in den **OFFLINE**-Modus zu wechseln. In diesem Modus wird die Verbindung zum Gerät unterbrochen. Ein erneuter Klick deaktiviert den **OFFLINE**-Modus.*

- **COMx**: Zeigt die gewählte COM-Port-Nummer (Programmierschnittstelle) an.

HINWEIS
*Klicken Sie auf den Indikator, um das Fenster **Porteinstellungen** zu öffnen.*

ACHTUNG
Wenn einer der Indikatoren überläuft, ist das Schreiben eines Programms auf das Gerät deaktiviert.

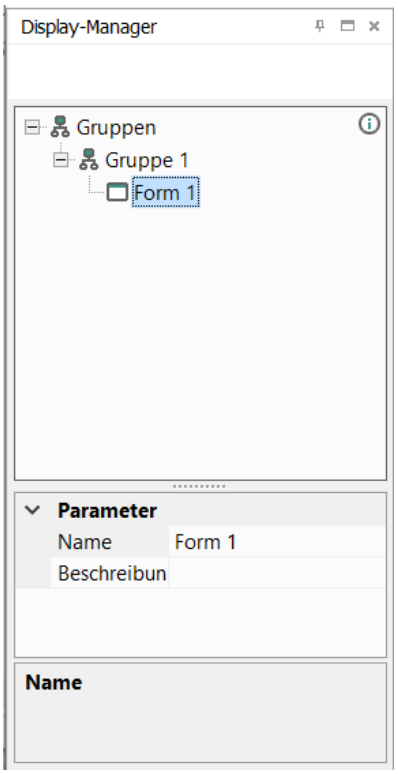
Sie können die Anzahl der Indikatoren in der Statusleiste im Menü ***Ansicht*** anpassen.

OFFLINE-Modus

Im ***OFFLINE-Modus*** wird die Verbindung zwischen ALP und dem Gerät unterbrochen.
Der OFFLINE-Modus kann über den Menüpunkt ***Service > OFFLINE-Modus*** oder durch einen Klick auf die Statusanzeige ***Gerät*** aktiviert/deaktiviert werden. Mit dem nächsten Klick wird der OFFLINE-Modus deaktiviert.
Weitere Einzelheiten finden Sie im Abschnitt *Projekt auf Gerät hochladen* .

3.8 Display-Manager

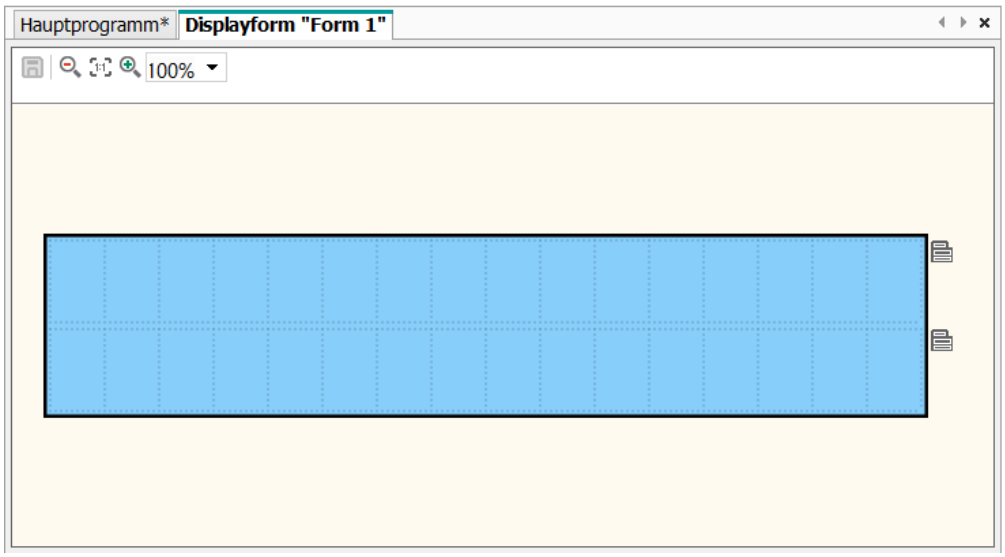
Wenn das Zielgerät über ein Display verfügt, können die angezeigten Informationen mithilfe einer oder mehrerer Anzeigeformen programmiert werden. Weitere Einzelheiten zur Displayprogrammierung finden Sie im Abschnitt *Displayprogrammierung*.
Die Programmierung erfolgt mit dem Programmiertool ***Display-Manager***. Der Reiter Display Manager befindet sich in der oberen linken Ecke des Fensters. Klicken Sie auf den Reiter, um das Panel zu öffnen. Das Panel enthält eine Symbolleiste, eine hierarchische Struktur (Baum) von Displayformen und die Parameter des ausgewählten Objekts.

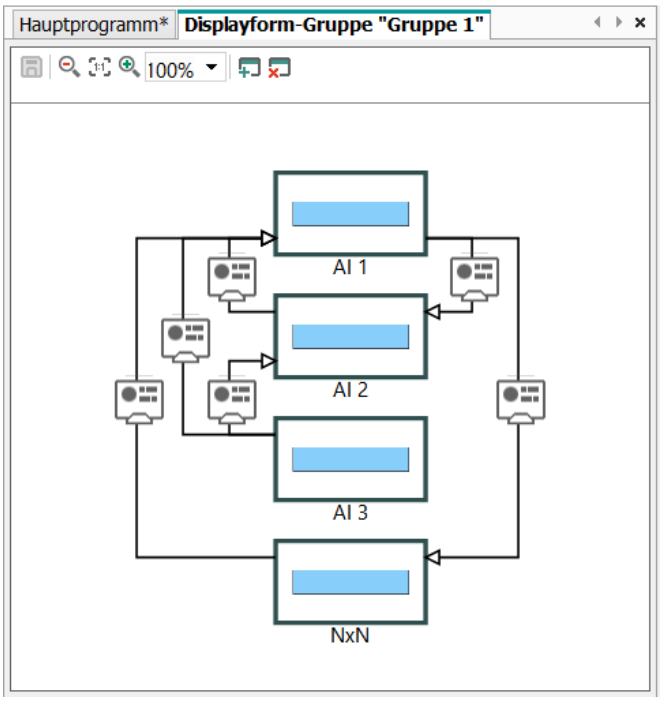


Im unteren Teil des Panels werden die Parameter der gewählten Anzeigeform angezeigt.
Um das ausgewählte Formular zu programmieren, öffnen Sie es in einem separaten Arbeitsbereich ***Displayform***, über das Kontextmenü oder doppelklicken Sie auf die Form in der Gruppe.

	Displayform hinzufügen
	Displayform löschen
	Displayform bearbeiten

Auf der Arbeitsfläche wird die gewählte Displayform mit den Icons rechts daneben angezeigt, mit denen sich die Anzahl der angezeigten Zeilen verändern lässt. Die zuerst angezeigten Zeilen sind fett umrahmt.
Wenn mehr als eine Displayform erstellt wird, sollten Sie „Sprünge“ zwischen ihnen angeben, damit der Bediener zwischen den Formen wechseln kann, um die gewünschten Informationen anzuzeigen. Dies kann in einem separaten Arbeitsbereich ***Displayform-Gruppe*** erfolgen, der die grafische Struktur von Anzeigeformularen mit „Sprüngen“ und deren Bedingungen darstellt. Um ihn zu öffnen, verwenden Sie den Befehl ***Gruppe bearbeiten*** im Gruppenkontextmenü.





Oben im Anzeigeeditor und in der Bildschirmgruppe befinden sich Schaltflächen:

	Arbeitsbereich speichern
	Um 10 % verkleinern
	Originalgröße
	Um 10 % vergrößern

Der Maßstab kann über das Dropdown-Menü rechts neben den Schaltflächen geändert werden.

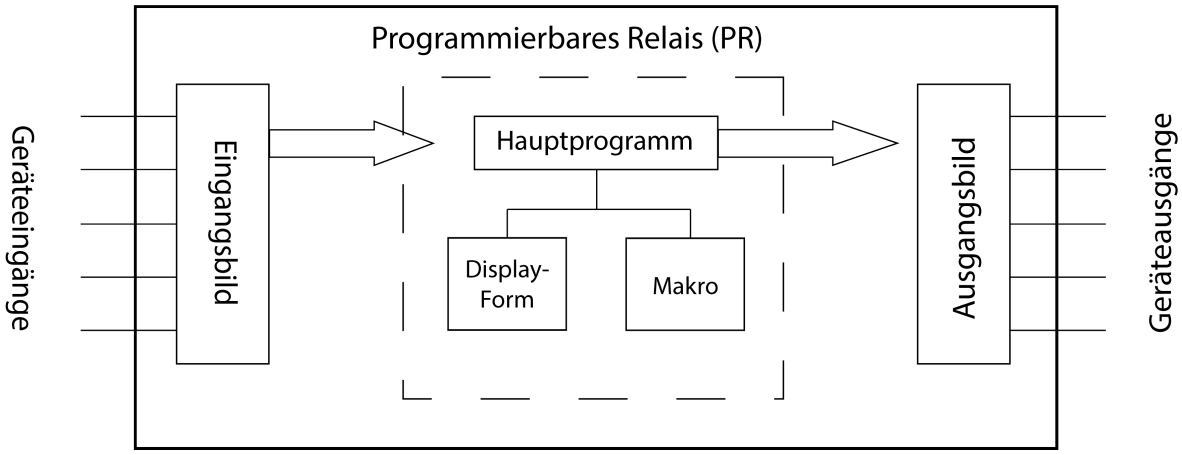
4 Allgemeine Informationen

In diesem Abschnitt werden die grundlegenden Konzepte des Geräts und das Prinzip der Erstellung eines Programms zum Laden in das Gerät beschrieben:

- *Programmausführung*
- *Projekterstellung*
- *Programmerstellung*
- *Display-Programmierung*
- *Simulation*
- *Verbindung zum Gerät*
- *Projekt auf Gerät hochladen*
- *Online-Debugging*
- *Projektinformationen*
- *Komponenten-Manager*
- *Makroentwicklung*
- *ST-Funktion*
- *ST-Funktionsblock*

4.1 Programmausfuehrung

Das ausgewählte Zielgerät bestimmt die Anzahl der verfügbaren Ein- und Ausgänge und die Verfügbarkeit einer Echtzeituhr.
Der allgemeine Aufbau des programmierbaren Relais:



Das programmierbare Relais ist eine Art SPS mit einem zyklisch ausgeführten Programm:

- Schritt 1 – Der Status der physischen Eingänge wird in den Eingangsspeicherzellen (Input Image Table) gespeichert.
- Schritt 2 – Die eingegebenen Speicherzellen werden ausgelesen und das Programm wird vom ersten bis zum letzten Befehl ausgeführt.
- Schritt 3 – Die Ergebnisse werden in den Ausgabespeicherzellen (Output Image Table) gespeichert und auf die Ausgaben angewendet.

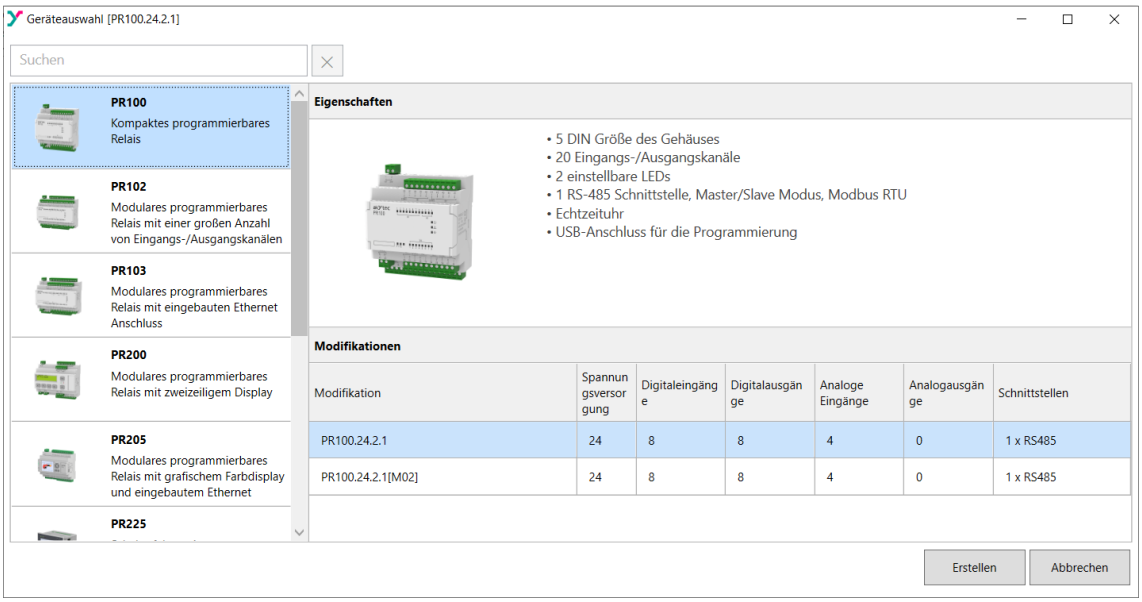
Wenn der letzte Schritt abgeschlossen ist, wird das Programm erneut vom ersten Schritt an ausgeführt.

4.2 Projekterstellung

Projekterstellung

So erstellen Sie ein neues Projekt:

1. Klicken Sie in der Taskleiste auf das Symbol oder wählen Sie im Hauptmenü ***Datei*** →***Neues Projekt…***.
2. Wählen Sie im Dialogfenster ***Geräteauswahl*** das Zielgerät aus und bestätigen Sie mit ***OK***.



Der Auftritt des neuen Projekts:

– *Arbeitsbereich* – Leeres Schaltprogramm
– *Statusleiste* – Informationen zu verfügbaren Ressourcen
– *Bibliothek* – Verfügbare Programmbausteine
– *Eigenschaften* – Arbeitsbereichseigenschaften

HINWEIS
Wenn ein Gerät mit dem PC verbunden ist, schlägt ALP im Auswahlfenster das Modell des verbundenen Gerätes vor.

Wenn das ausgewählte Gerät über ein Display verfügt, wird links vom Arbeitsbereich die Registerkarte *Geräte-Manager* angezeigt. Mit diesem Tool können Sie die angezeigten Informationen *programmieren*.
Sie können das aktuelle Projekt speichern oder ein gespeichertes Projekt öffnen, indem Sie die entsprechenden Schaltflächen in der Symbolleiste oder im Hauptmenü ***Datei*** verwenden.

Schaltungsprogrammentwicklung

Nun können Sie im Arbeitsbereich das *Hauptschaltprogramm* mit den gängigen Programmbausteinen aus der Symbolleiste ***Einfügen*** und den spezifischen Programmbausteinen aus der *Bibliothek* erstellen. Zeichnen Sie Verbindungslinien zwischen Eingängen, Ausgängen und Bausteinen, um logische Verknüpfungen im Programm herzustellen.

Simulation

Das Programm kann offline simuliert werden. Starten Sie den *Simulationsmodus* über den Menüpunkt ***Service*** →***Simulation*** oder das Symbol ▶ in der Symbolleiste, ändern Sie den Zustand der Eingänge und beobachten Sie den Zustand der Ausgänge, um die Korrektheit des Programms zu überprüfen.

Online-Debugging

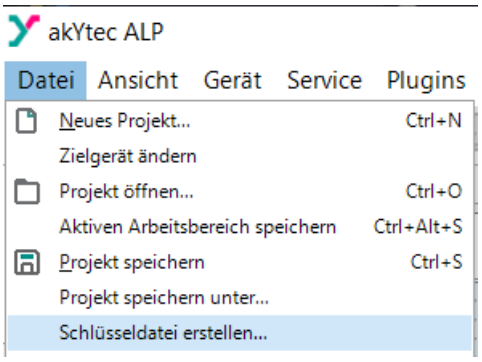
Wenn das Gerät angeschlossen ist und das Programm im Gerät und in ALP identisch ist, können Sie mithilfe des *Online-Debuggings* die Korrektheit des Programms im Gerät überprüfen.

4.2.1 Projekt mit Schluesseldatei

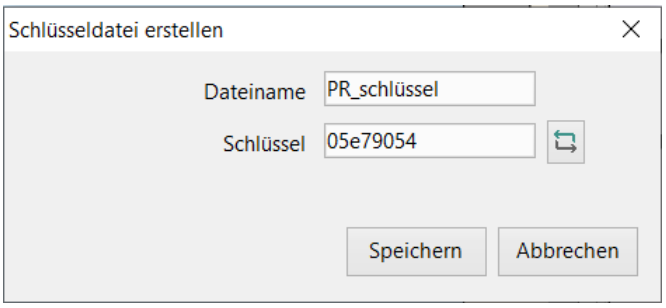
Erstellen einer Datei mit einem Schlüssel

So erstellen Sie eine Datei mit einem Schlüssel:

1. Wählen Sie im Hauptmenü **Datei** → **Schlüsseldatei erstellen…**



2. Geben Sie im sich öffnenden Dialogfeld den Dateinamen ein und generieren Sie den Schlüssel. Sie können den Schlüssel bearbeiten, indem Sie auf die Schaltfläche klicken oder ihn manuell bearbeiten.

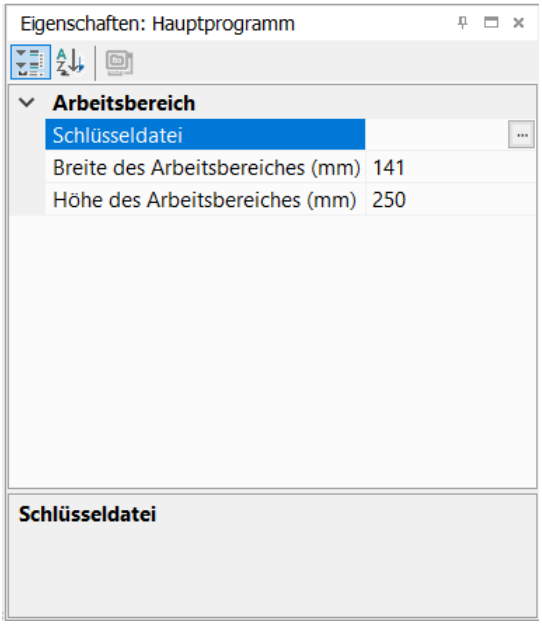


3. Datei speichern. Standardmäßig speichert ALP die Schlüsseldatei im Ordner *C:\Benutzer\Benutzername\Dokumente\ALP\Schlüssel*.

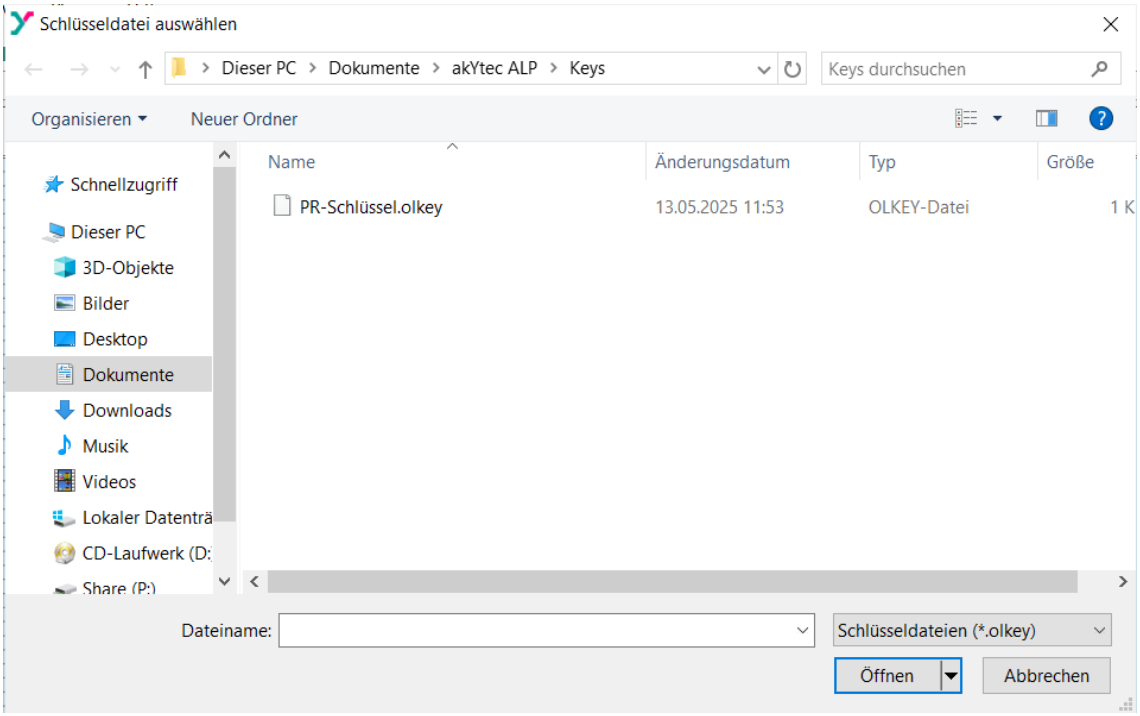
Schlüsseldatei binden

So binden Sie eine Schlüsseldatei an ein Projekt:

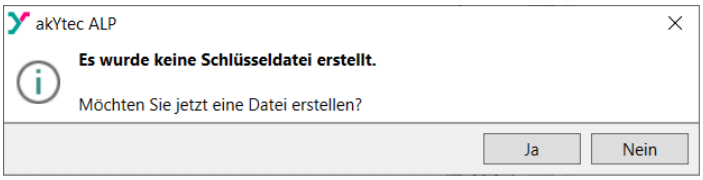
1. Klicken Sie in den Dateieigenschaften (**Eigenschaften: Hauptprogramm**) auf das Symbol "**…**".



2. Wählen Sie im neuen Fenster die Schlüsseldatei aus.



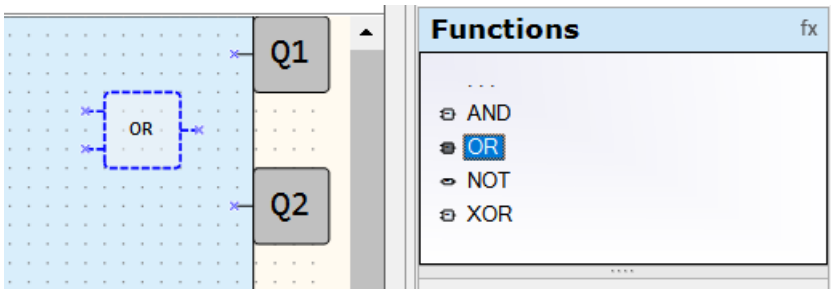
HINWEIS
Wenn im Ordner *C:\Benutzer\Benutzername\Dokumente\akYtec ALP\Keys* kein Schlüssel vorhanden ist, bietet ALP an, einen neuen Schlüssel zu erstellen.



3. Klicken Sie auf die Schaltfläche **Öffnen**.

4.3 Programmentwicklung

Es wird empfohlen, mit der Erstellung eines Schaltprogramms mit der Planung zu beginnen. Der Plan sollte alle möglichen Zustände des Geräts während des Betriebs in Form eines Modusdiagramms, einer Tabelle der E/A-Zustände, eines elektrischen oder Funktionsdiagramms usw. beschreiben.
Nachdem alle Betriebsaufgaben beschrieben wurden, kann das Programm mithilfe der Standardblöcke aus der Symbolleiste ***Einfügen*** und der spezifischen Blöcke aus der Projektbibliothek entwickelt werden. Die in der *Bibliothek* dargestellte Projektbibliothek enthält die für das Zielgerät verfügbaren Funktionen und Funktionsblöcke sowie die dem Projekt hinzugefügten Makros.
Um einen Bibliotheksblock im Schaltprogramm zu platzieren, wählen Sie den gewünschten Block in der Bibliothek aus und ziehen Sie ihn per Drag & Drop auf die Arbeitsfläche.



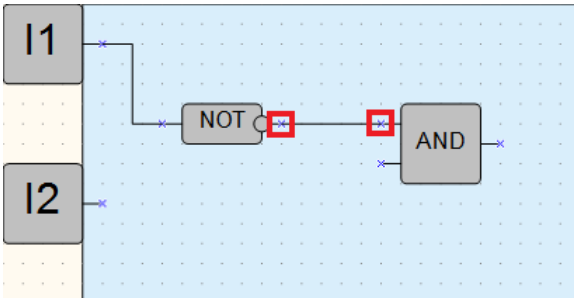
Um Verbindungslinien zwischen den Ein- und Ausgängen der Geräte- und Programmblöcke zu zeichnen, verwenden Sie die linke Maustaste.

– Klicken Sie auf den Eingangspin des Gerätes. Die Linie wird daran angehängt und folgt dem Mauszeiger.
– Um die Linienrichtung zu ändern, klicken Sie auf einen Punkt im Arbeitsbereich.
– Ziehen Sie die Linie bis zum Eingangspin eines Blocks und klicken Sie darauf, um die Linie fertigzustellen.

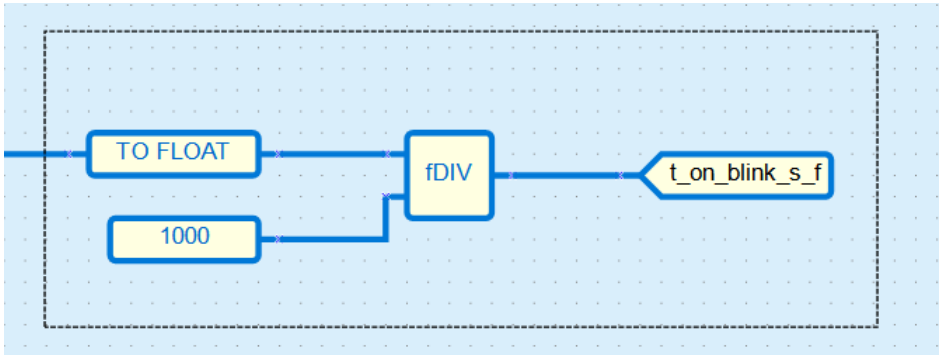
Die Verbindungslinie kann nur zwischen Anschlusspins gezeichnet werden, denen der gleiche Datentyp zugeordnet ist.

HINWEIS
Wenn die Eingabe- und Ausgabedatentypen der Komponenten nicht übereinstimmen, kann die Verbindungslinie nicht erstellt werden. Weitere Informationen finden Sie im Abschnitt Variablentypen.

Klicken Sie auf den Block, um ihn auszuwählen. Um eine Gruppe auszuwählen, ziehen Sie das Rechteck um mehrere Blöcke.

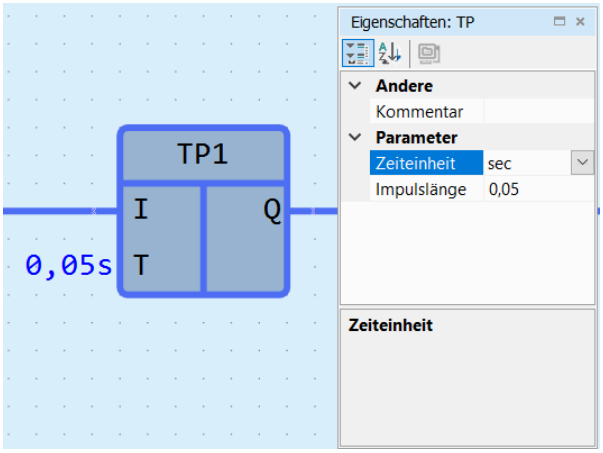


Um die den unterschiedlichen Datentypen zugeordneten Anschlusspins zu verbinden, verwenden Sie Konvertierungsblöcke.

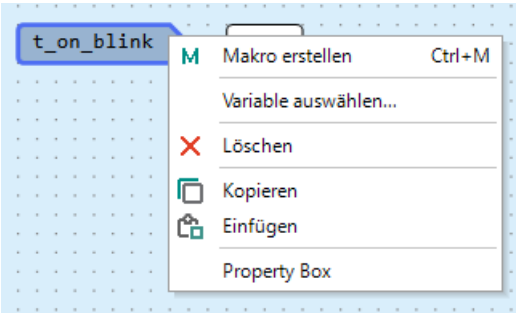


Komponenteneinstellungen

Die Parameter jedes Elements können im *Eigenschaftenfenster* festgelegt werden.



Verwenden Sie das Blockkontextmenü für alle für das Element verfügbaren Manipulationen.

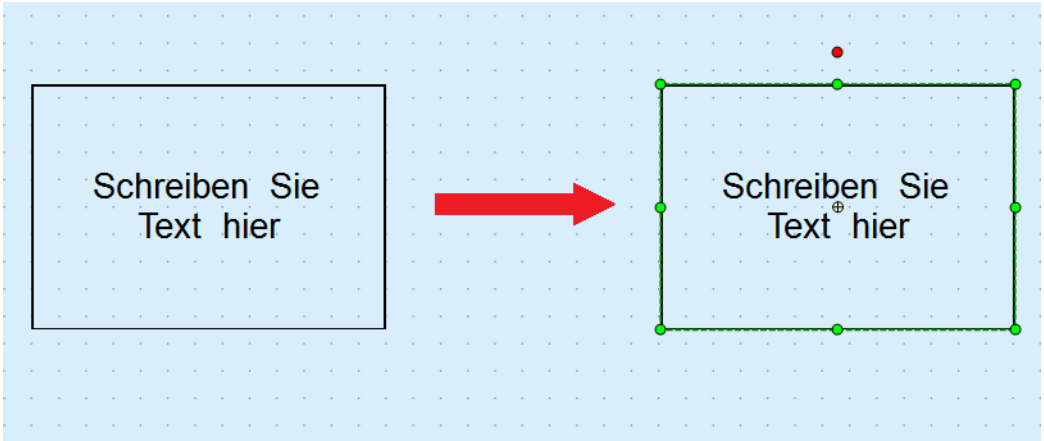


Zur Entwicklung des Programms werden folgende Bausteine und Funktionen verwendet, die im Einfügepanel aufgerufen werden:

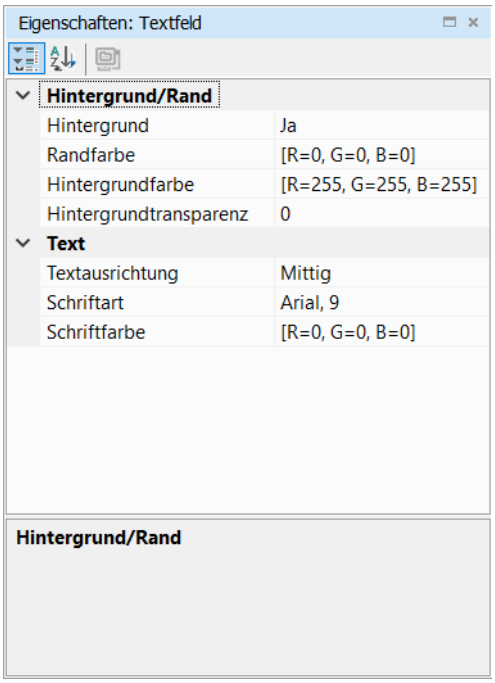
Textfeld	Fügt einen Textkommentar zum Schaltprogramm hinzu.
Variablenbaustein	Fügt einen Baustein zum Schreiben oder Lesen von Programmwerten hinzu.
Konstanten-Baustein	Erstellt eine Verzögerung um einen Zyklus bei der Übertragung eines Wertes vom Ausgang einer Komponente zu ihrem Eingang.
Verzögerungslinie	Eine Verzögerung für einen Zyklus der Übertragung eines Wertes vom Ausgang der Komponente zu ihrem Eingang erstellen
Netzwerkvariablen-Baustein	Fügt Bausteine für den Datenaustausch zwischen Geräten hinzu, die an ein gemeinsames Netzwerk angeschlossen sind.
ReadFromFB-/WriteToFB-Bausteine	Liest Werte einzelner Parameter eines Funktionsbausteins in eine Variable ein oder schreibt sie aus einer Variable zurück.
Konvertierung	Konvertiert Werte unterschiedlicher Typen für die Übertragung.
Elemente anordnen	Weist Sequenznummern von FB-Schemata neu zu.
Ausführungsablauf	Ändert die Reihenfolge, in der Programmausgabewerte berechnet werden.

4.3.1 Textfeld

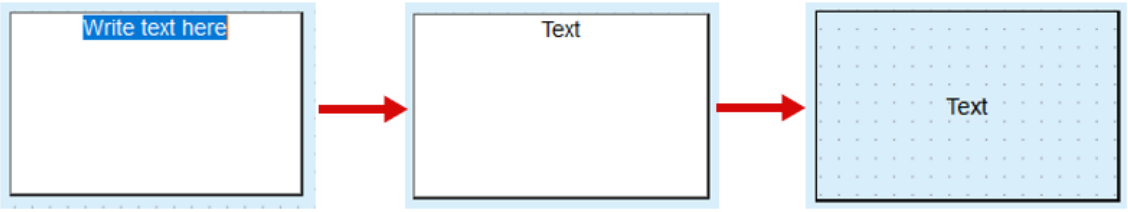
Die Textfelder dienen zur Erläuterung des Programms.
Um dem Programm ein Textfeld hinzuzufügen, klicken Sie in der Symbolleiste ***Einfügen*** auf das Element, klicken Sie dann auf den Punkt im Arbeitsbereich, um die obere/linke Ecke des Textfelds zu platzieren, und zeichnen Sie ein Rechteck, um seine Größe festzulegen.



Die Parameter des Textfelds können im ***Eigenschaftenfenster*** geändert werden.



Um die Hintergrundfarbe des Textblocks sichtbar zu machen, wird empfohlen, die ***Hintergrundtransparenz*** auf über 20 % einzustellen.
Doppelklicken Sie auf das Textfeld, um den Text zu schreiben.



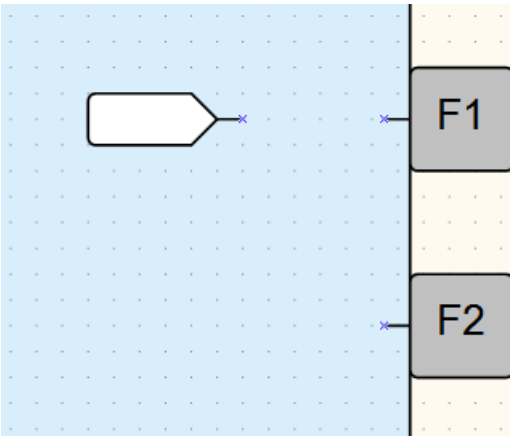
4.3.2 Variablenbaustein

Der Variablenbaustein ermöglicht die Verwendung einer Variablen in einem Schaltprogramm.
Um dem Programm einen Variablenbaustein hinzuzufügen, klicken Sie im Einfügefeld auf das entsprechende Symbol:

– **Variablen-Eingangsbaustein** – Um einen Wert an das Programm zu übergeben

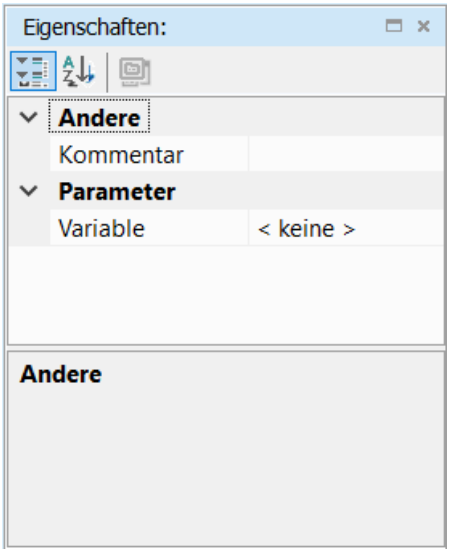
– **Variablen-Ausgangsbaustein** – Um Werte aus dem Programm hineinzuschreiben

Klicken Sie auf den Punkt im Arbeitsbereich, um den variablen Block zu platzieren.

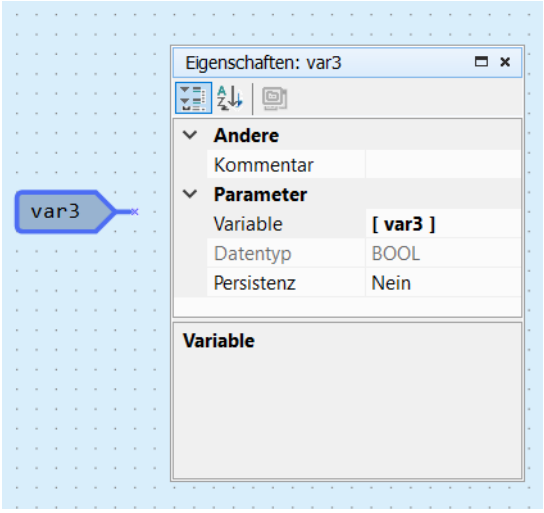


Ein Variablenbaustein kann dem Programm auch aus der *Variablentabelle* hinzugefügt werden.
So weisen Sie einem Variablenbaustein eine Variable zu:

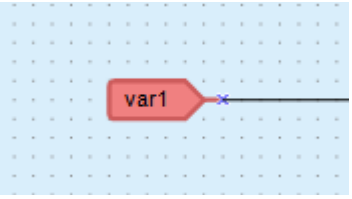
1. Wählen Sie einen Variablenbaustein aus.
2. Doppelklicken Sie auf den Baustein oder klicken Sie auf das Symbol "**...**" in der Zeile ***Variable*** im Propety Box, um eine Variable für den Baustein auszuwählen oder in der geöffneten *Variablentabelle* eine neue zu erstellen.



3. Bestätigen Sie die Auswahl mit ***OK***. Die Variable wird dem Baustein zugewiesen.

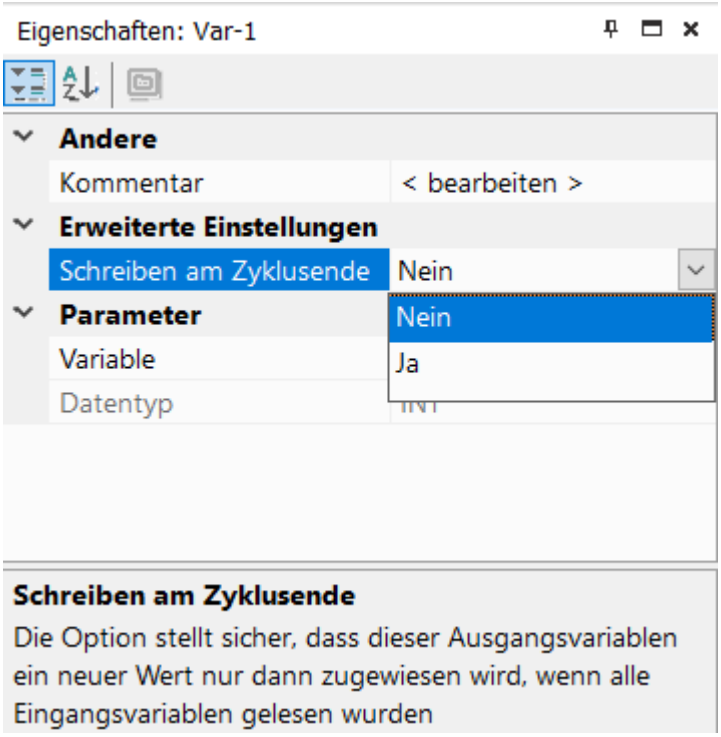


Zur Auswahl stehen nur die entsprechenden Reiter der Tabelle. Die Verfügbarkeit wird durch den Bausteintyp eingeschränkt.
Verbinden Sie den Variablenbaustein mit dem gewünschten Element im Arbeitsbereich.
Wenn der Variablenbaustein rot hervorgehoben ist, bedeutet dies, dass die Erstellung fehlerhaft oder nicht abgeschlossen ist. Die Informationen zum Fehler werden in der Statusleiste angezeigt.



Für den Baustein einer Ausgangsvariablen, die an einen Geräteparameter oder an ein Visualisierungselement gebunden ist, steht die Einstellung **Schreiben am Ende des Zyklus** zur

Verfügung, die dazu dient, den Wert nach allen Lesevorgängen im *Arbeitszyklus* in die Variable zu schreiben.



Der Standardwert in der Einstellung **Schreiben am Zyklusenende** wird abhängig von der Eigenschaft **Nichtflüchtigkeit** der an den Ausgangsbaustein gebundenen Variable festgelegt:

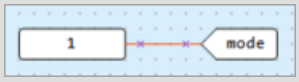
– Wenn das Kontrollkästchen **Persistenz** nicht markiert ist (das Kontrollkästchen ist deaktiviert), dann wird **Schreiben am Zyklusenende** auf **Nein** gesetzt.
– Wenn das Kontrollkästchen **Persistenz** aktiviert ist (das Kontrollkästchen ist eingeschaltet), dann wird **Schreiben am Zyklusenende** auf **Ja** gesetzt.

Der Standardwert für die Einstellung **Schreiben am Zyklusenende** wird wie folgt geändert:

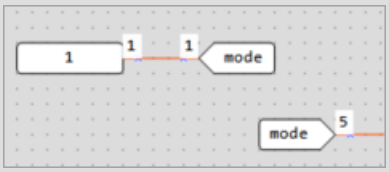
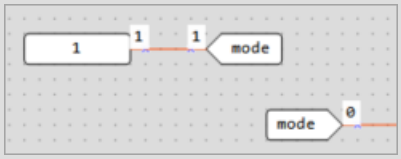
– immer dann, wenn sich die Bindung einer Variablen an einen Ausgabeblock ändert.
– immer dann, wenn sich die **Persistenz**-Eigenschaft der an den Ausgabeblock gebundenen Variablen ändert.

Beispiele für die Verwendung der Parametereinstellung **Schreiben am Zyklusenende** mit empfohlenen Werten:

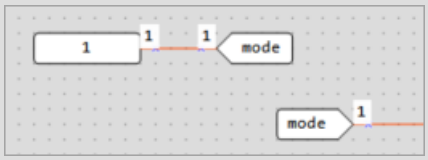
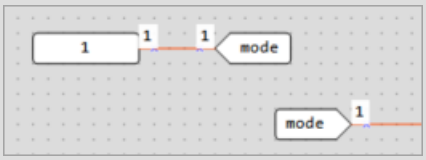
Beispiel 1
Es wird eine Variable namens „mode“ erstellt und an das Visualisierungselement gebunden. Die Variable wird mit dem Wert des Arbeitsbereichs initialisiert.



Es ist notwendig, im ersten *Arbeitszyklus* eine Aktion mit dieser Variable durchzuführen.
Wenn **Schreiben am Zyklusenende** auf **Ja**:

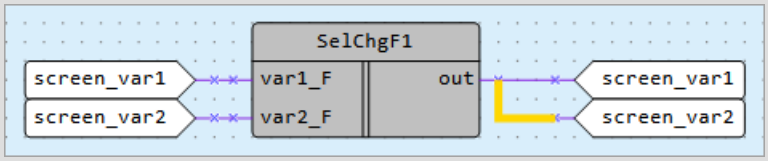
Persistenz Variable	Nichtpersistenz Variable
Die Modusvariable im Eingabevariablenblock wird mit einem Standardwert (z. B. 5) initialisiert: 	Die Modusvariable im Eingabevariablenblock wird auf Null initialisiert: 

Es wird empfohlen, **Schreiben am Zyklusenende** auf **Nein** einzustellen:

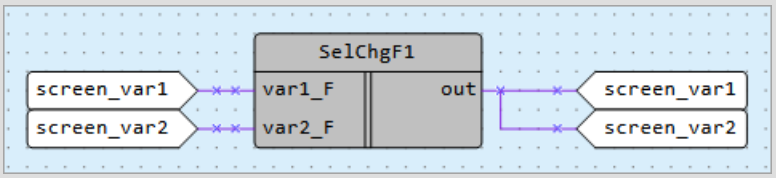
Persistenz Variable	Nichtpersistenz Variable
Die Modusvariable im Eingabevariablenblock wird mit dem Wert aus dem Arbeitsbereich initialisiert: 	Die Modusvariable im Eingabevariablenblock wird mit dem Wert aus dem Arbeitsbereich initialisiert: 

Es wird empfohlen, **Schreiben am Zyklusenende** auf **Nein**.

Beispiel 2
Es wurden zwei Standardvariablen, screen_var1 und screen_var2, erstellt, die mit verschiedenen Visualisierungselementen verknüpft sind, welche vom Gerätebildschirm aus modifiziert werden können.
Die Werte dieser Variablen müssen synchronisiert werden. Hierfür können Sie die Makros SelChg/SelChgF aus dem Koponenten-Manager verwenden.
Das Makro SelChgF dient dazu, eine int/float-Variable mithilfe zweier Variablen zu ändern. Die Ausgabe des Makros protokolliert die letzte Änderung einer der beiden Variablen.
Wenn **Schreiben am Zyklusenende** auf **Nein** eingestellt ist (Standardeinstellung für persistenz Variablen):



Eine der Variablen (in diesem Beispiel screen_var2) weist eine zyklische Beziehung auf (gelb hervorgehoben), weshalb es nicht möglich ist, den Wert der Variablen vom Bildschirm aus zu ändern, da die Variable screen_var2 den Wert der Variable screen_var1 annimmt.
Wenn **Schreiben am Zyklusenende** auf **Ja** gesetzt ist (Standardeinstellung für nichtpersistenz Variablen), wird die zyklische Beziehung für die Variable screen_var2 aufgehoben, und beide Variablen können vom Bildschirm aus bearbeitet werden:

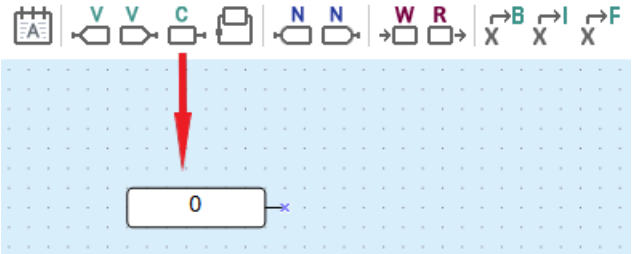


Es wird empfohlen, **Schreiben am Zyklusenende** auf **Ja**.
Die in den Beispielen enthaltenen Informationen sind lediglich Empfehlungen. Der Benutzer kann den Wert frei festlegen, der das gewünschte Verhalten in einem bestimmten Teil der Schaltung gewährleistet.

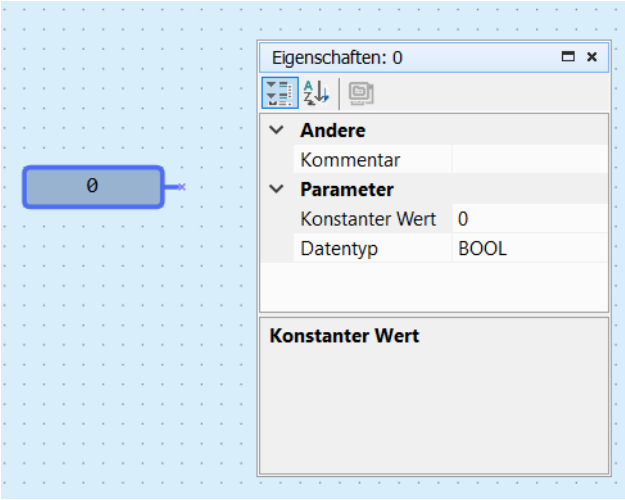
Es wird empfohlen, Schreiben am Zyklusende auf Ja einzustellen.

4.3.3 Konstanten-Baustein

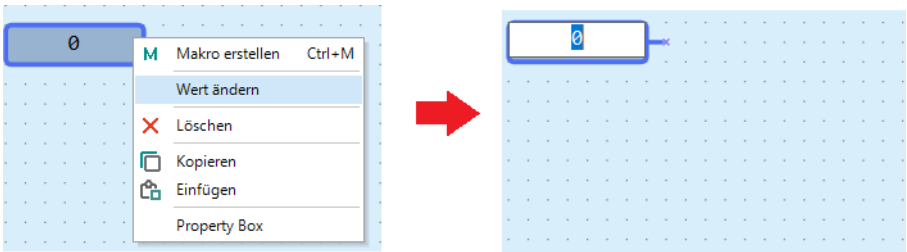
Um dem Programm einen konstanten Wert hinzuzufügen, klicken Sie in der Symbolleiste ***Einfügen*** auf das Symbol C und dann auf den Punkt im Arbeitsbereich, um den Konstantenblock zu platzieren.



Wählen Sie den Datentyp über das Symbol "**…**" in der Zeile ***Datentyp*** aus und tragen Sie den Wert in der Zeile ***Konstanter Wert*** im ***Eigenschaftenfeld*** ein.



Der Wert der Konstante unterliegt während der gesamten Programmausführung keiner Änderung. Er kann durch Doppelklicken auf den Konstantenblock im ***Eigenschaftenfeld*** oder durch Auswahl von ***Wert ändern*** im Bausteinkontextmenü geändert werden.



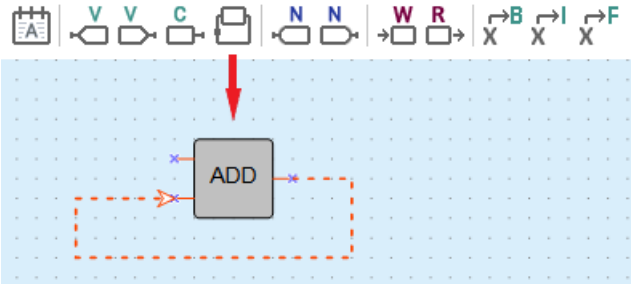
Gültige Werte der Datentypen:

BOOL	0 / 1
INT	0 … 4.294.967.295
REAL	–3,402823e+38 … 3,402823e+38

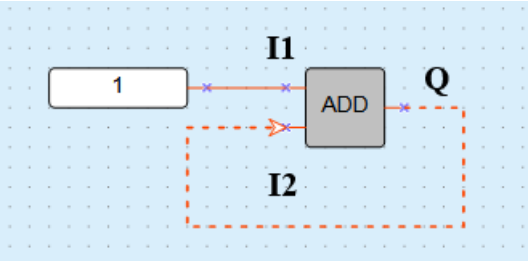
4.3.4 Verzoegerungslinie

Die Verzögerungslinie dient dazu, den Wert vom Bausteinausgang um einen Zyklus verzögert an den Bausteineingang zu übertragen. Ausgang und Eingang können zu unterschiedlichen Bausteinen gehören.

Klicken Sie in der Symbolleiste ***Einfügen*** auf das Symbol und ziehen Sie eine Linie vom Ausgang zum Eingang eines Funktionsbausteins. Die Verzögerungslinie wird im Hauptprogramm als gestrichelte Linie mit einem Pfeil dargestellt, dessen Farbe dem Datentyp der Variablen entspricht.



Beispiel:
Ein konstanter Wert 1 wird an den Eingang I1 des Additionsblocks ADD (Integer) übertragen. Ein im vorherigen Zyklus berechneter Wert vom Blockausgang (Q) wird über eine Verzögerungsleitung an den Eingang I1 übertragen.



Zyklussignalwerte:

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

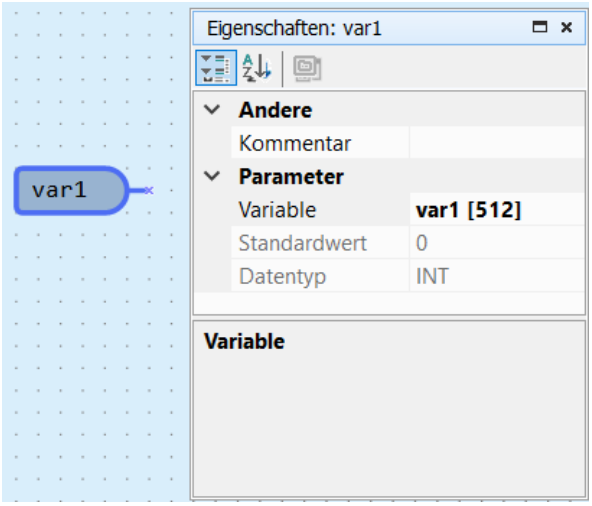
4.3.5 Netzwerkvariablen-Baustein

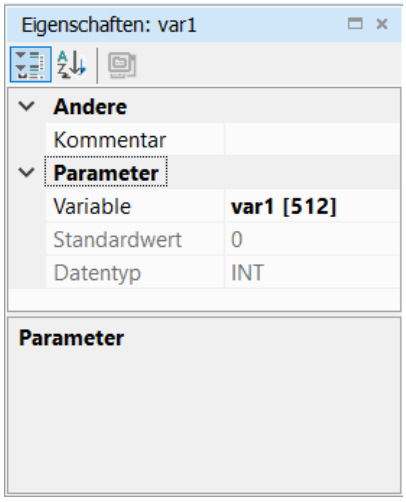
Die Netzwervariablen-Ein- und Ausgansbausteine sind spezielle Variablenbausteine für den Datenaustausch zwischen Geräten, die an ein gemeinsames Netzwerk angeschlossen sind.

– – Netzwerk-Ausgangsvariablen sind die Variablen, die über das Netzwerk gelesen werden können.
– – Netzwerk-Eingangsvariablen sind die Variablen, die über das Netzwerk geschrieben werden können.

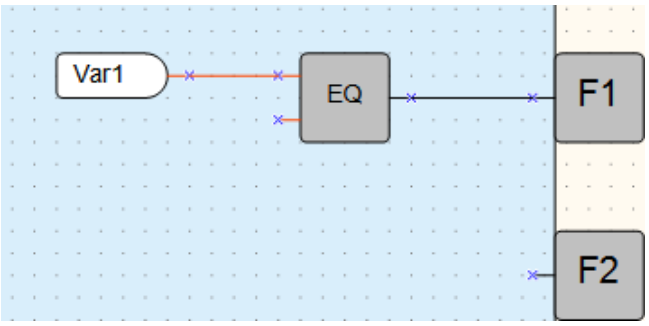
Hinweis: Dem Baustein kann keine Variable zugewiesen werden, wenn in der Gerätekonfiguration keine Kommunikationsschnittstellen vorhanden sind.
So fügen Sie dem Programm eine Netzwerkvariable hinzu:

1. Klicken Sie in der Symbolleiste ***Einfügen*** auf das Symbol oder .
2. Klicken Sie auf den Punkt im Arbeitsbereich, um den Variablenbaustein zu platzieren.
3. Doppelklicken Sie auf den Baustein oder klicken Sie auf das Symbol "**...**" in der Zeile ***Variable*** im Eigenschaftenfeld, um eine Variable für den Baustein auszuwählen oder in der geöffneten *Variablentabelle* eine neue zu erstellen.
4. Bestätigen Sie die Auswahl mit ***OK***. Die Variable wird dem Baustein zugewiesen.





Zur Auswahl stehen nur die entsprechenden Reiter der Tabelle. Die Verfügbarkeit wird durch den Bausteintyp eingeschränkt.
Verbinden Sie den Netzwerkvariablenbaustein mit dem gewünschten Element im Arbeitsbereich.



Wenn der Variablenbaustein rot hervorgehoben ist, bedeutet dies, dass die Erstellung fehlerhaft oder nicht abgeschlossen ist.
Die Informationen zum Fehler werden in der Statusleiste angezeigt.
Es wird empfohlen, mit der Programmierung mit dem Anlegen von Variablen in der Variablentabelle zu beginnen.
Wird die Variable mehrfach in einem Projekt verwendet, können alle Verweise mit dem Eintrag ***Verknüpfungen anzeigen*** im Kontextmenü des Variablenbausteins verfolgt werden.

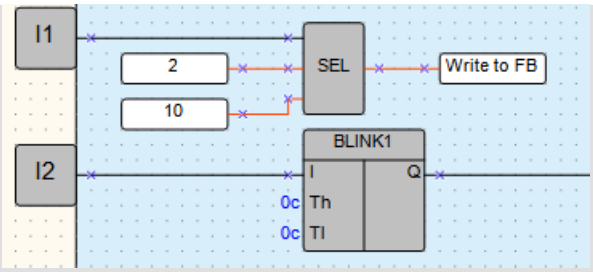
4.3.6 ReadFromFB-/WriteToFB-Bausteine

Die ReadFromFB/WriteToFB-Bausteine werden verwendet, um den Parameterwert eines Funktionsbausteins während der Programmausführung zu lesen oder zu schreiben.
Die folgenden Bausteine können dem Programm hinzugefügt werden:

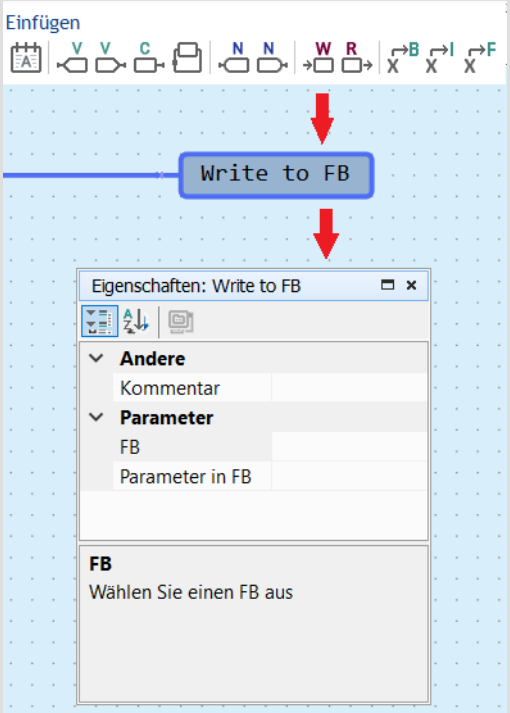
– W — um den Wert in einen FB zu schreiben
– R — um den Wert aus einem FB zu lesen

Um während des Vorgangs einen FB-Parameter zu verändern, dient der Baustein ***WriteToFB*** (W).

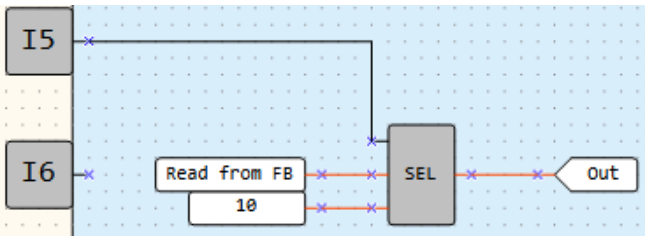
> ***Beispiel:***
> Der Wert des Parameters ***Einschaltdauer*** des FB ***BLINK1*** sollte je nach Wert am Eingang ***I1*** 2 oder 10 betragen.



Gehen Sie zum ***Eigenschaftenfeld***, wählen Sie den FB ***BLINK1*** in der Zeile ***Funktionsbaustein*** und den Parameter des FB in der Zeile ***Parameter in FB***.

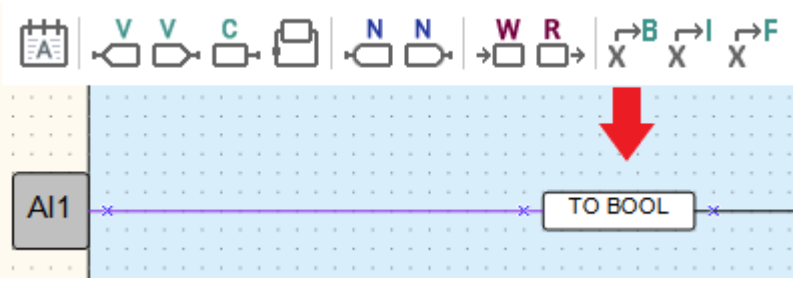


Der Baustein ***ReadFromFB*** (R) dient dazu, den aktuellen Wert eines FB-Parameters auszulesen und im Programm zu verwenden. Die Verwendung ist die gleiche wie beim Baustein **WriteToFB**.



4.3.7 Konvertierungsbausteine

Eine Kommunikationsverbindung zwischen Programmkomponenten kann nur für Ein- und Ausgaben desselben Typs hergestellt werden: BOOL, INT, REAL oder Array (mit gleicher Anzahl und gleichem Typ von Elementen). Um eine Kommunikationsverbindung zwischen Ein- und Ausgaben unterschiedlicher Typen herzustellen, verwenden Sie Konvertierungsblöcke.
Die Verwendung von Konvertierungsblöcken mit Variablen vom Typ Array ist nicht möglich. Die Konvertierung einzelner Array-Elemente ist jedoch zulässig.
Um den Konvertierungsblock dem Programm hinzuzufügen, klicken Sie in der Symbolleiste ***Einfügen*** auf das entsprechende Symbol und anschließend auf die gewünschte Stelle im Arbeitsbereich.

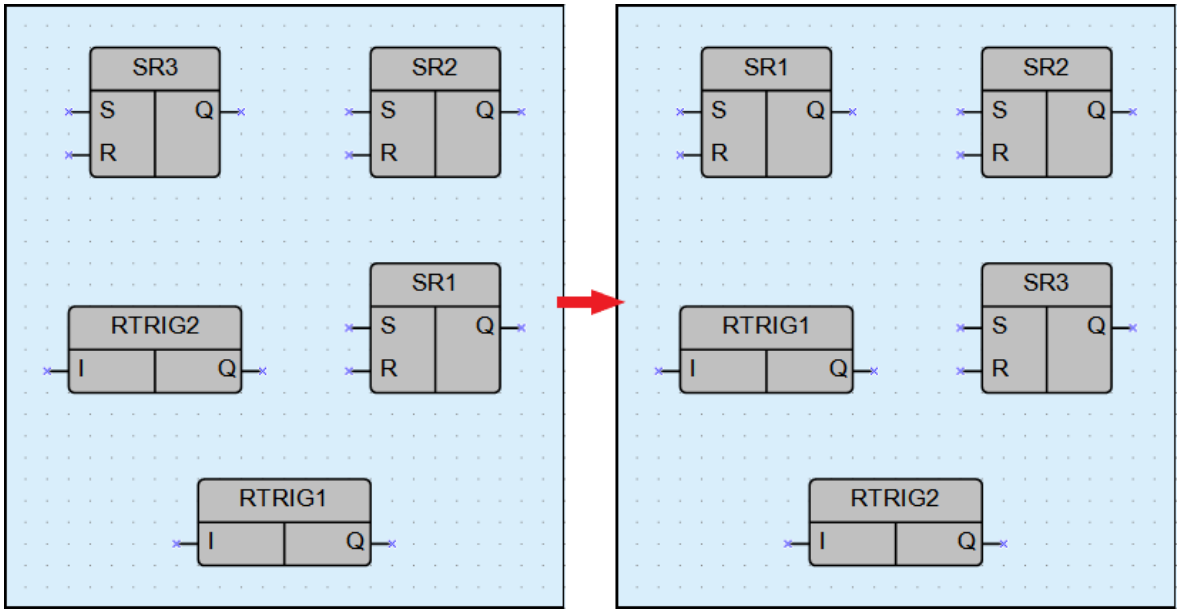


Konvertierungsbausteine:

B	**Konvertierung in BOOL**	Konvertiert INT- oder REAL-Werte in BOOL. Ist der Eingangswert größer als 0, wird der Ausgangswert auf 1 (TRUE) gesetzt.
I	**Konvertierung in INT**	Konvertiert BOOL- oder REAL-Werte in INT. REAL-Werte werden auf INT abgerundet, negative Werte werden in 0 umgewandelt.
F	**Konvertierung in REAL**	Konvertiert BOOL- oder INT-Werte in REAL.

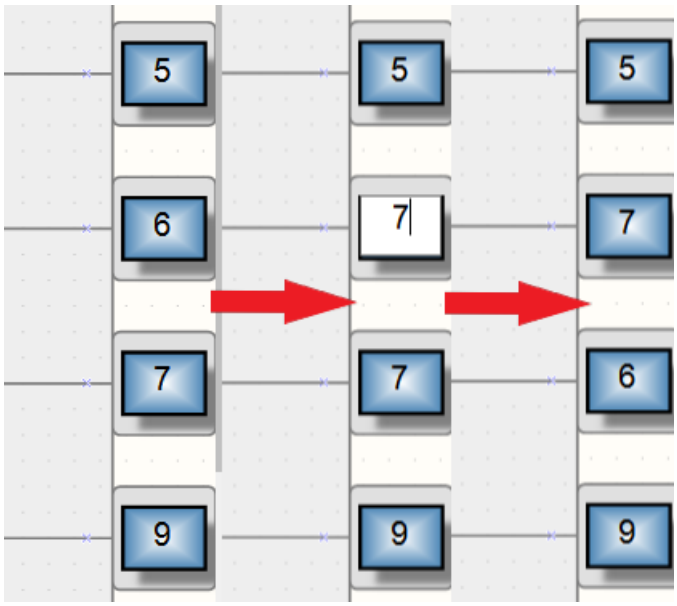
4.3.8 Elemente anordnen

Die Reihenfolgenummern der Funktionsblöcke können durch Klicken auf die Schaltfläche ***Elemente anordnen*** in der Symbolleiste ***Service*** automatisch neu vergeben werden. Die Bausteine des gleichen Typs werden dabei von oben nach unten und von links nach rechts durchnummeriert.



4.3.9 Ausfuehrungsablauf

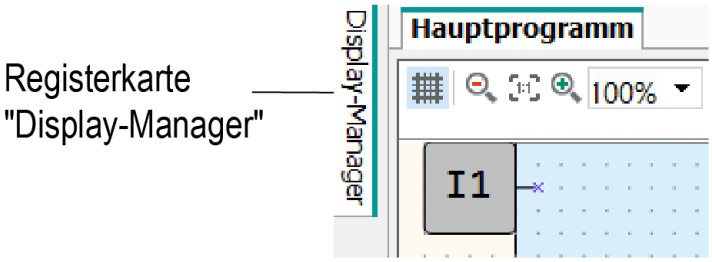
Die Berechnung der Werte für Ausgänge und Verzögerungslinien erfolgt in einer bestimmten Reihenfolge. Um diese Reihenfolge anzuzeigen, klicken Sie auf den Pfeil neben dem Symbol in der Symbolleiste ***Service*** und wählen Sie ***Verzögerungslinien*** oder ***Ausgänge***.
ALP wechselt in den Modus zum Festlegen der Ausführungsreihenfolge. Die Sequenznummern der Ausführungsreihenfolge werden neben den Ausgängen und Rückmeldungen angezeigt.
Um die Reihenfolge zu ändern, doppelklicken Sie auf einen Ausgang oder eine Verzögerungsleitung und geben Sie die gewünschte Nummer ein.



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

4.4 Display-Programmierung

Um die angezeigten Informationen zu bestimmen, verwenden Sie die Registerkarte *Display-Manager* in der oberen linken Ecke des Fensters. *Display-Manager* ist nur für Zielgeräte mit einem Display verfügbar.



Der Display-Manager unterscheidet sich bei Geräten mit grafischen Farb-LCDs und solchen mit monochromen Text-LCDs wie folgt:

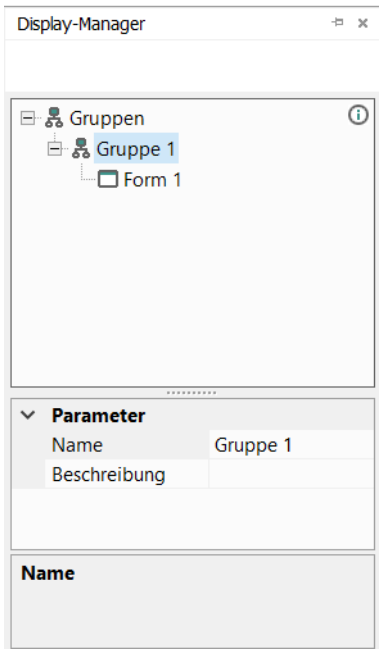


Abb. 4.1 Bildschirmmanager für Geräte mit monochromen Text-LCDs

Standardmäßig wird Displayform 1 der Gruppe 1 hinzugefügt:

– Bearbeitungsmöglichkeiten für Gruppe 1 sind verfügbar.
– Gruppe 1 kann nicht gelöscht werden.

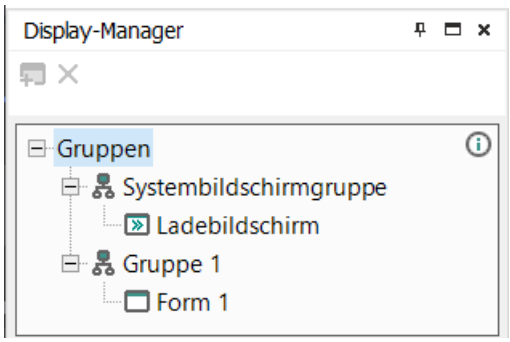


Abb. 4.2 Bildschirmmanager für Geräte mit grafischem Farb-LCD

Standardmäßig wird der Startbildschirm der Gruppe „Systembildschirme“ hinzugefügt:

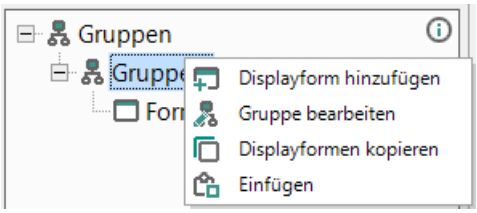
– Die Gruppe „Systembildschirme“ und der Startbildschirm können nicht bearbeitet oder gelöscht werden.

Standardmäßig wird Displayform 1 der Gruppe 1 hinzugefügt:

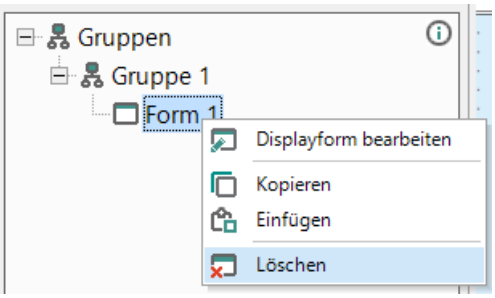
– Bearbeitungsmöglichkeiten für Gruppe 1 sind verfügbar.
– Gruppe 1 kann nicht gelöscht werden.

Displayform hinzufügen

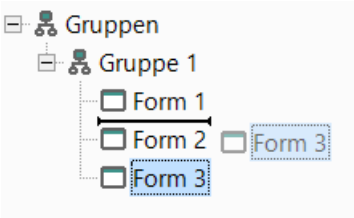
Das Display kann über eine oder mehrere Displayformen mit „Sprüngen“ dazwischen programmiert werden, so dass die angezeigten Informationen durch Programm-Ereignisse (Änderung der Variablen) oder durch den Bediener (Tastenereignis) geändert werden können.
Um eine Displayform hinzuzufügen, klicken Sie mit der rechten Maustaste auf das Element ***Gruppe 1*** und wählen Sie im Kontextmenü ***Displayform hinzufügen***.



Um eine Displayform zu löschen, klicken Sie mit der rechten Maustaste auf die gewünschte Form und wählen Sie im Kontextmenü ***Löschen***.



Um die Position der Displayform zu ändern, ziehen Sie sie bei gedrückter ***Umschalttaste*** an die neue Stelle.



Beim Ziehen wird die neue Position als horizontale Markierung angezeigt.

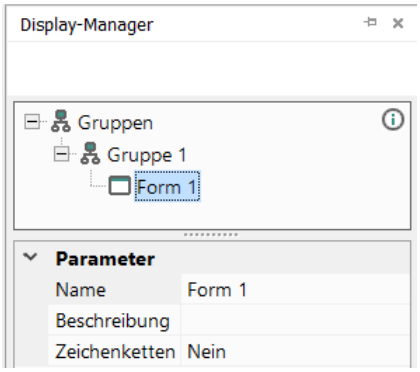
Displayform-Bearbeiter

Die Funktionalität des *Displayform-Bearbeiter* hängt vom Displaytyp des Geräts ab:
– *Monochromes Text-LCD 4.4.1*.
– *Grafisches Farb-LCD 4.4.2*.

4.4.1 LCD mit monochromem Text

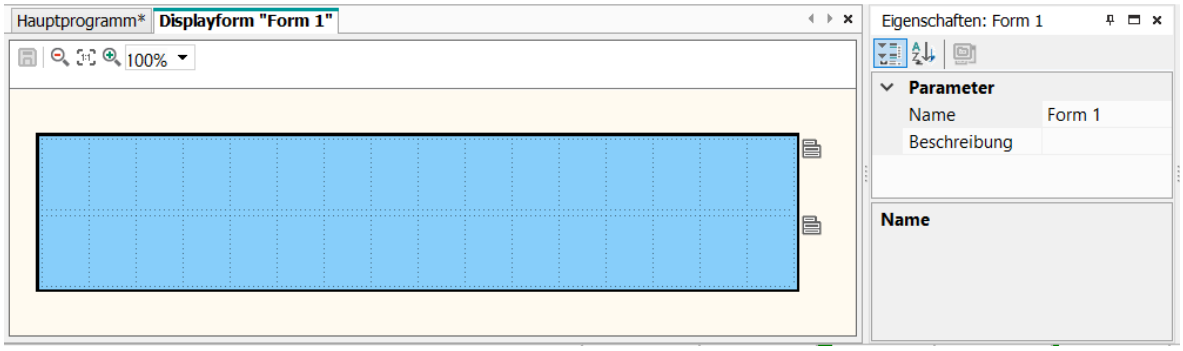
Bildschirmparameter

Parameter die Bildschirminhalte werden im Eigenschaftenfeld angezeigt. **Display-Editor**.



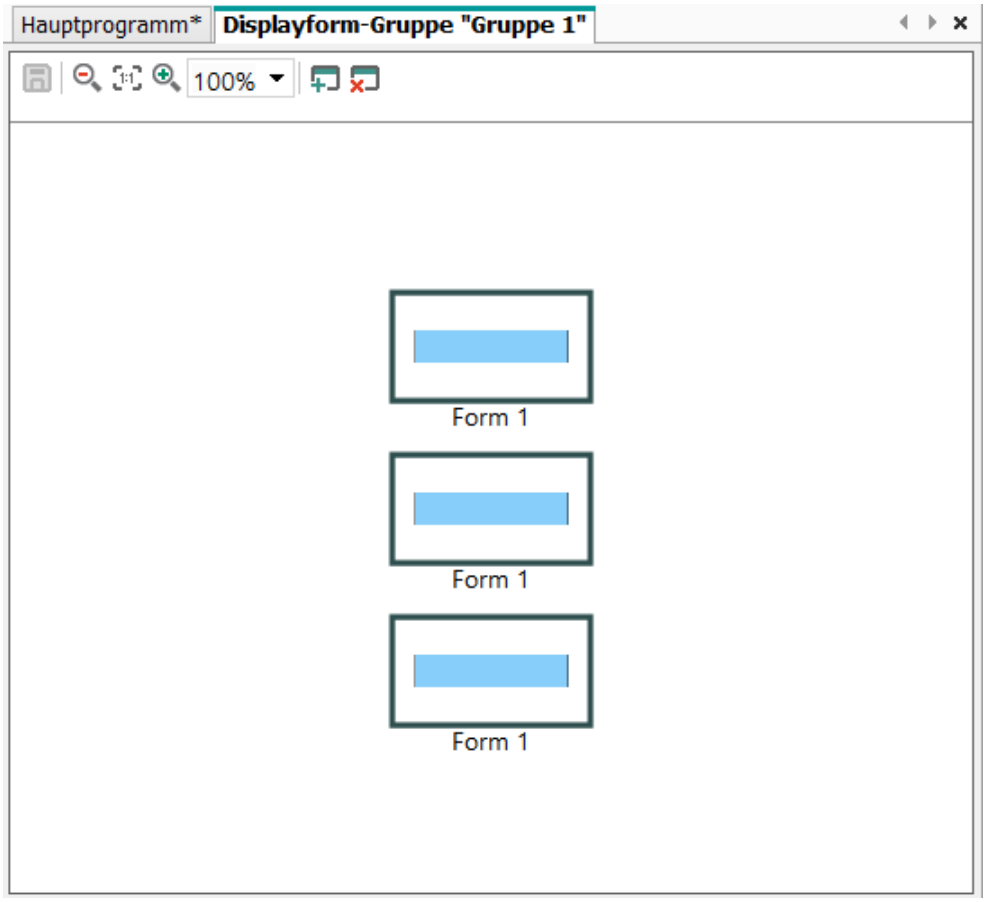
– **Name** – zur Anzeige in **Display-Editor** und im Titel des Bildschirmeditors.
– **Beschrebung** – Textbeschreibung des Bildschirms.
– **Zeichenketten ausblenden** – Ausblenden/Anzeigen von leeren Zeilen auf dem Gerätebildschirm.

Zur Präsentation **parameter** Doppelklicken Sie im Eigenschaftenfenster auf den gewünschten Bildschirm **Display-Editor**.



Display-Editor

Für Geräte mit monochromem Text-LCD steht außerdem ein Bildschirmgruppeneditor zur Verfügung, der durch Auswahl des Befehls geöffnet wird. **Gruppe bearbeiten** im Kontextmenü der Bildschirmgruppe in **Display-Editor**.



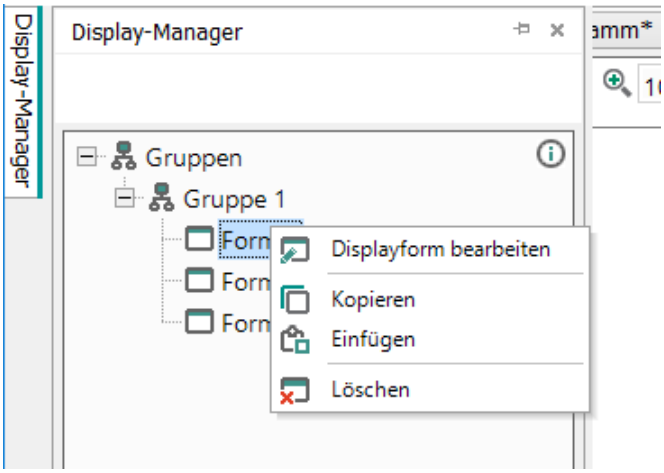
Im oberen Bereich des Bildschirmeditors und der Bildschirmgruppe befinden sich die folgenden Schaltflächen:

	Arbeitsbereich speichern
	Zoom -
	Originalgröße
	Zoom +

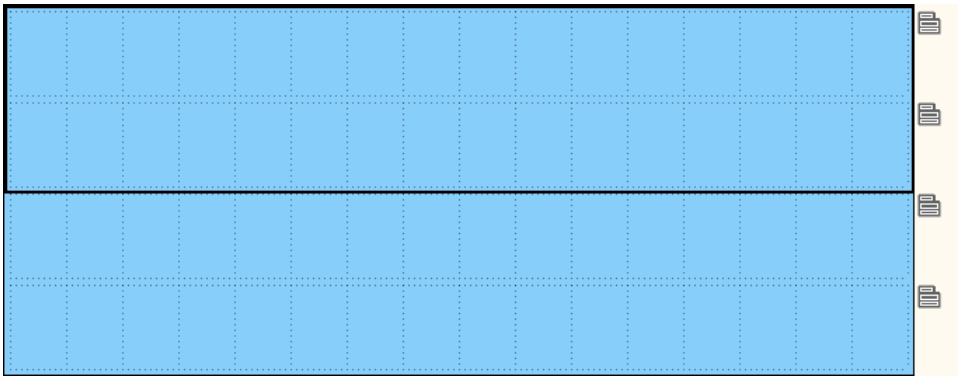
Der Maßstab kann über das Dropdown-Menü rechts neben den Schaltflächen geändert werden.

Bildschirmgruppen-Editor

Um die Informationsdarstellung auf dem Bildschirm anzupassen, öffnen Sie den Bildschirmeditor durch Doppelklicken auf den Bildschirm im Bildschirmmanager oder durch Klicken auf die Schaltfläche **Displayform bearbeiten** aus dem Kontextmenü des Bildschirms.



Der Bildschirm kann mehrere Zeilen enthalten, auf die über die Tasten des Geräts zugegriffen werden kann. Um Zeilen hinzuzufügen oder zu entfernen, klicken Sie auf die Schaltfläche **Zeilenkontextmenü** und wählen Sie den gewünschten Befehl aus.

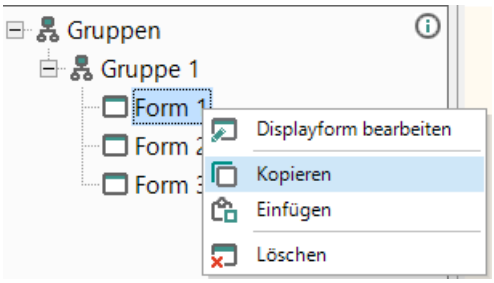


Die Zeilen enthalten Steuerelemente, die aus dem Bedienfeld gezogen werden. **Bibliothek** mithilfe der Drag-&-Drop-Methode. Beschreibungen der Blöcke finden sich im Abschnitt *Steuerelement*.

HINWEIS
Der Zeichensatz ist in der Windows-1252-Codierung implementiert.

Displayform kopieren und einfügen

Im Bildschirmmanager können Sie Bildschirme kopieren und in das aktuelle oder ein anderes Projekt einfügen. Um die ausgewählten Bildschirme zu kopieren, wählen Sie den Befehl **Kopieren** im Kontextmenü eines Bildschirms oder einer Gruppe von Bildschirmen oder durch Drücken einer Tastenkombination **Strg+C**. Mehrere Bildschirme können durch Gedrückthalten der Taste ausgewählt werden **Strg** oder **Schicht**.



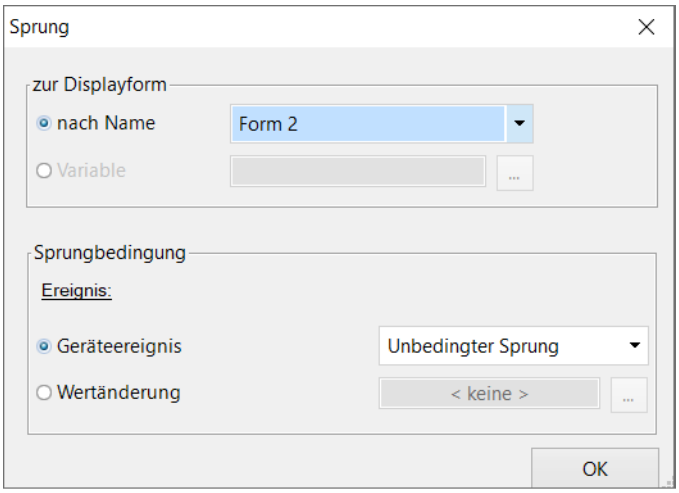
Um kopierte Bildschirme einzufügen, wählen Sie den entsprechenden Befehl im Kontextmenü des Bildschirms oder der Bildschirmgruppe aus. **Einfügen**oder drücken Sie die Tastenkombination **Strg + V**.
Alle auf dem Bildschirm platzierten Steuerelemente und Bildschirmeigenschaften werden zusammen mit dem Bildschirm kopiert. Die an den Bildschirm gebundenen Variablen werden gemäß dem Abschnitt in ein anderes Projekt kopiert. *Variablen kopieren*.
Falls Übergänge zwischen den ausgewählten Bildschirmen vorhanden sind, werden diese zusammen mit den Bildschirmen kopiert. Falls der Bildschirm einen Übergang außerhalb des ausgewählten Bereichs aufweist, wird dieser Übergang entfernt.

Sprünge

Wenn das Projekt zwei oder mehr Bildschirme umfasst, müssen Übergänge (Schaltbedingungen) zwischen ihnen eingerichtet werden.
Um einen Übergang zu erstellen, sollten Sie Folgendes tun:

1. Klicken Sie mit der rechten Maustaste auf das Element ***Gruppe 1*** im Display-Manager-Baum und wählen Sie im Kontextmenü ***Gruppe bearbeiten***. Die Registerkarte des Displaygruppen-Editors wird geöffnet.
2. Wählen Sie im Displaygruppen-Editor die Startform aus.
3. Klicken Sie auf das Symbol «**…**» in der Zeile ***zur Displayform*** im ***Eigenschaftenfeld***. Das Dialogfenster „Sprung“ wird geöffnet.

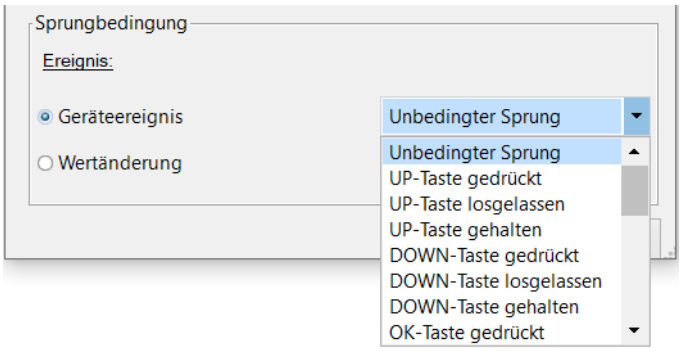
   Wählen Sie im Dropdown-Menü eine andere Form aus, zu der Sie wechseln möchten.



4. Wählen Sie im Bereich ***Sprungbedingung*** das Ereignis aus, als Geräteereignis oder als Änderung einer Variablen durch Wert.

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

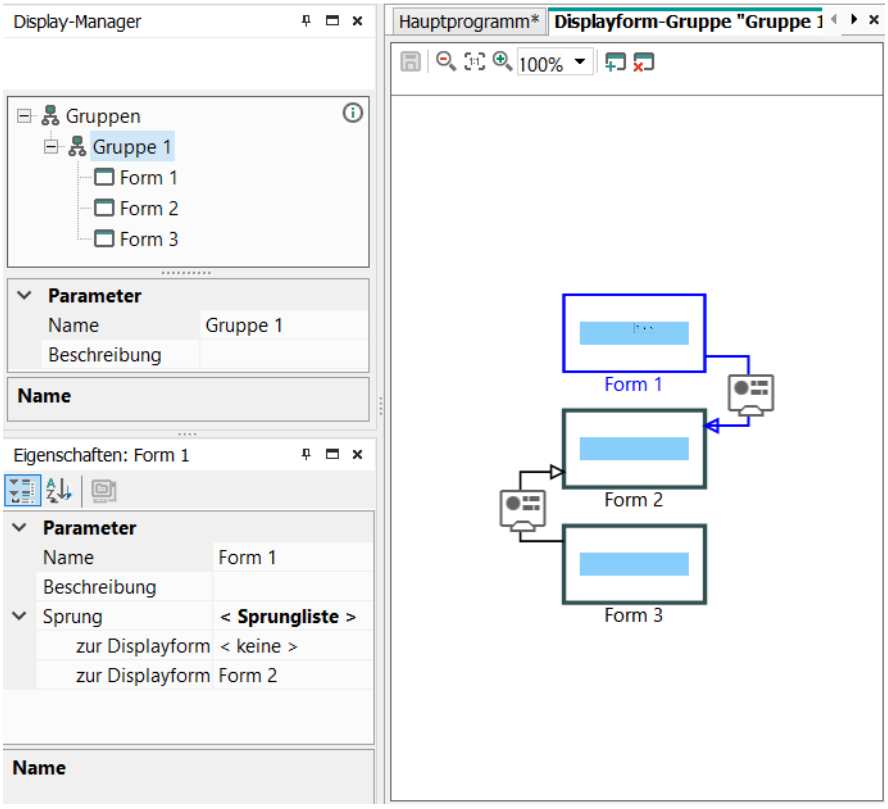
   Für das Ereignis ***Wertänderung*** kann eine BOOL-Variable ausgewählt werden.



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

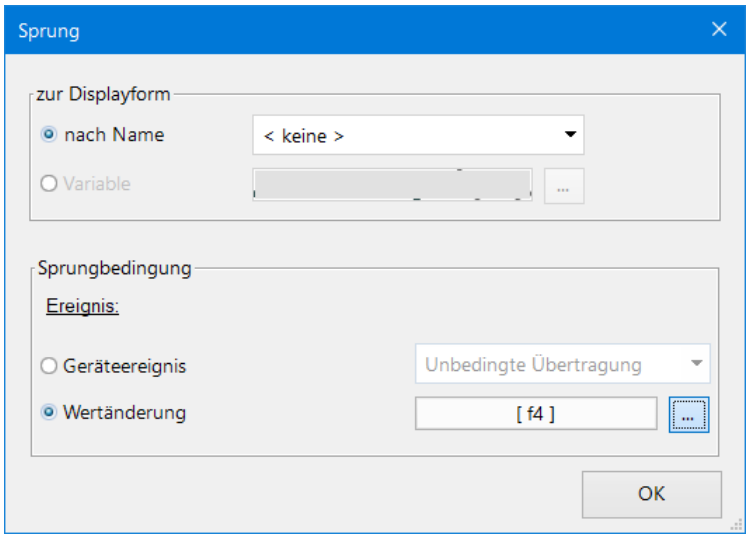
> **HINWEIS**
> Ein unbedingter Sprung wird verwendet, um einen Übergang zu konfigurieren, wenn keine Ereignisse oder Änderungen im Wert einer Variablen auftreten, beispielsweise um zu einem Alarmbildschirm zu gelangen.

6. Nach der Erstellung wird der Übergang im Diagramm im Bildschirmgruppeneditor und in den Bildschirmeigenschaften angezeigt.



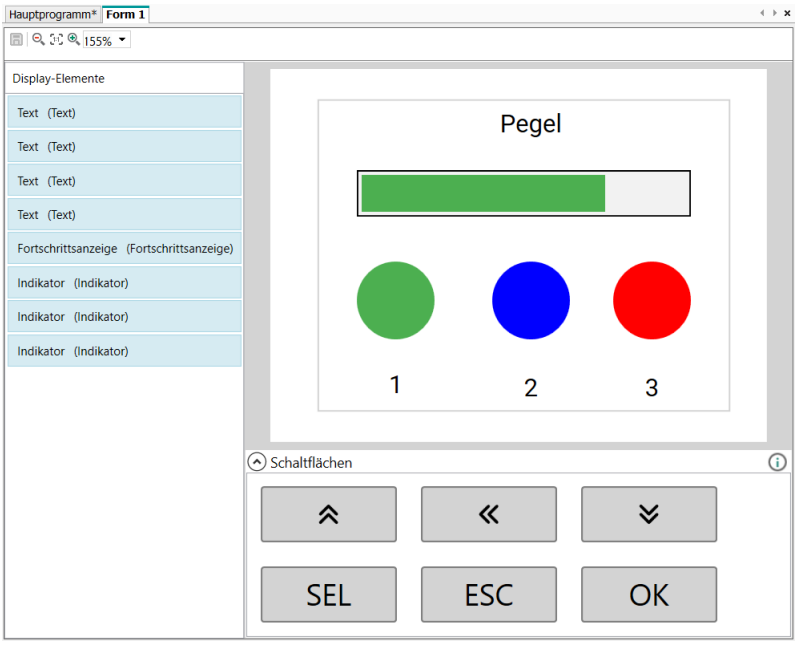
Der Übergang von Bildschirm zu Bildschirm kann durch verschiedene Ereignisse ausgelöst werden; hierfür können Sie mehrere einrichten (**Sprungliste**) Übergänge. Übergänge werden auf die gleiche Weise hinzugefügt wie oben beschrieben.

Um einen Übergang zu bearbeiten oder zu löschen, wählen Sie ihn in den Bildschirmeigenschaften aus und konfigurieren Sie die entsprechenden Einstellungen. Der Übergang wird entfernt, wenn das Feldlänge **nach Name** Das Menü zur Übergangserstellung hat folgenden Wert: **< kein >**.



4.4.2 Grafisches Farb-LCD

Bei Geräten mit grafischem Farb-LCD wird die Displayausgabe über grafische Elemente programmiert.

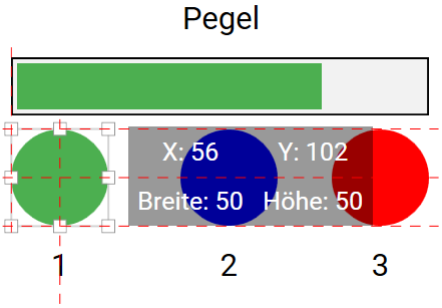


Displayeditorbereiche:

- **Display-Elemente** dienen zur Anzeige der Komponentenliste im Bildschirmfeld. Dort können Sie auch die Reihenfolge der Grafikelemente ändern und ihnen Namen zuweisen.
- **Display-Feld** dient zur Anordnung der Grafikelemente.
- **Schaltflächen** dienen zur Programmierung der Gerätetasten.

Grafische Elemente, die im Display platziert werden sollen, können aus der **Komponentenbibliothek** ausgewählt werden. Elementbeschreibungen finden Sie unter *Basiselemente*.
Im *Simulatormodus* ist ein Bildschirmsimulationsfenster verfügbar.

Display-Feld



Wenn grafische Elemente in das Feld verschoben werden, werden Tooltips mit Koordinaten und Abmessungen des Elements sowie Hilfslinien relativ zu anderen Elementen angezeigt.
Für Grafikelemente und Grafikelementgruppen stehen folgende Kontextmenüaktionen zur Verfügung:

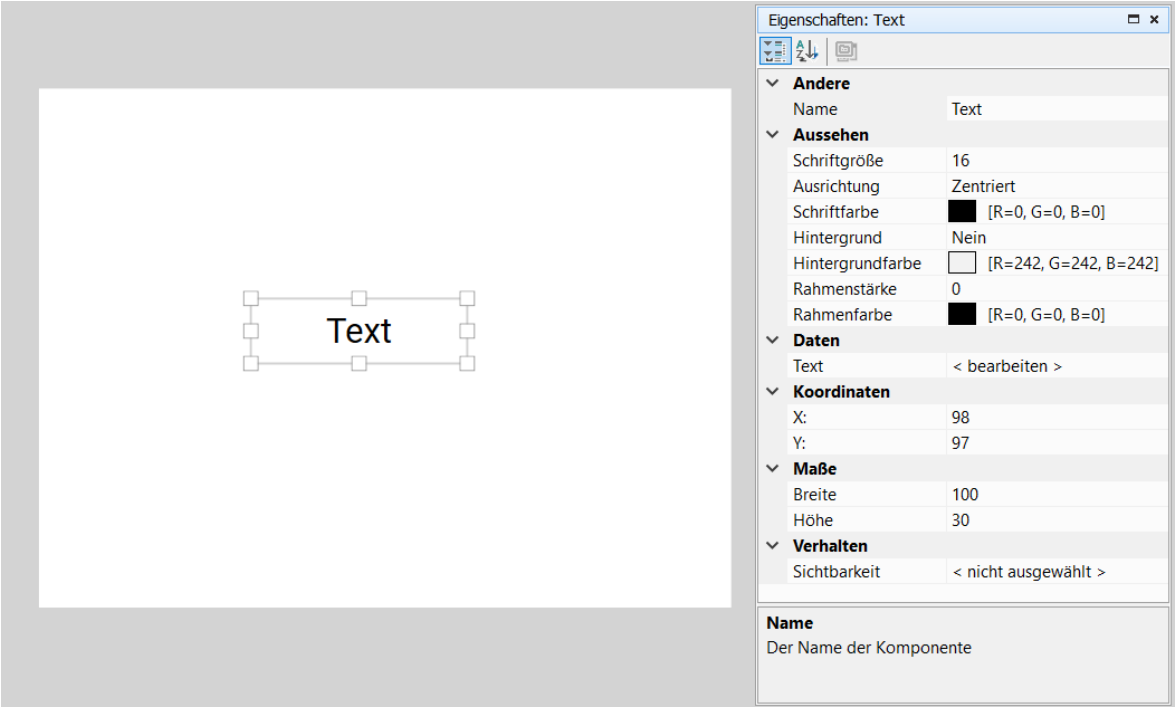
– Kopieren
– Einfügen
– Löschen
– In den Vordergrund und Hintergrund verschieben
– Ausrichten

Doppelklicken Sie auf die grafischen Elemente **Text** und **Dynamischer Text**, um den Texteingabemodus zu aktivieren, oder auf andere Elemente, um das Fenster zur Variablenanbindung zu öffnen.
Grafische Elemente können mit den Pfeiltasten der Tastatur verschoben werden, und andere Aktionen können mit *Tastaturkürzeln* ausgeführt werden.

Eigenschaften grafischer Elemente

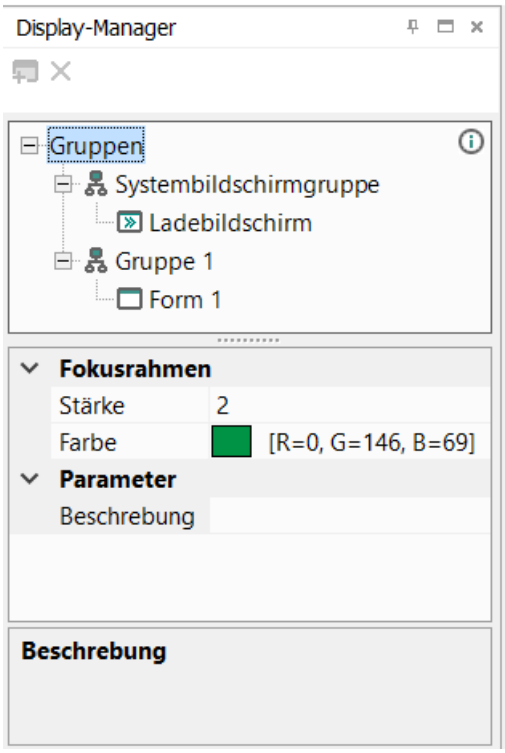
Um die Eigenschaften eines Grafikelements im Eigenschaften-Panel anzuzeigen, klicken Sie in der Elementliste links neben dem Feld oder im Display-Feld auf das gewünschte Element.
Gemeinsame Eigenschaften aller Grafikelemente:

– **Elementname**
– **Größe** – Breite und Höhe in Pixeln
– **Koordinaten** – Auf der X- und Y-Achse ist der Bezugspunkt die obere linke Kante des Elements.



Bildschirmgruppen

Im Eigenschaftenfenster des Anzeigemanagers für den Ordner „Gruppen“ können Sie Parameter konfigurieren, die auf alle Visualisierungselemente auf allen Bildschirmen innerhalb dieser Gruppe angewendet werden. Änderungen an den Einstellungen auf Gruppenebene wirken sich automatisch auf alle grafischen Elemente auf allen Bildschirmen der Gruppe aus.



– **Parameter**:

  Beschreibung – Textbeschreibung der Bildschirmgruppe.

– **Fokusrahmen**:

Stärke – die Stärke des Rahmens um ein fokussiertes grafisches Element, sowohl im Simulationsmodus als auch auf dem Gerätebildschirm.

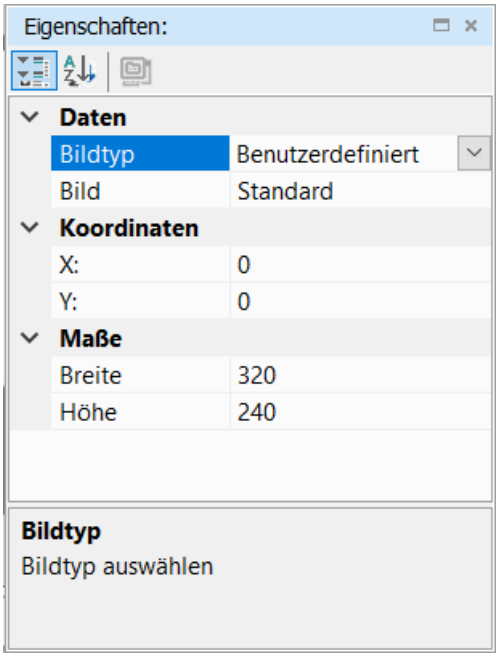
Farbe – die Farbe des Rahmens um ein fokussiertes grafisches Element, sowohl im Simulationsmodus als auch auf dem Gerätebildschirm.

Ladebildschirm

Bei Geräten mit grafischem Farb-LCD können Sie ein benutzerdefiniertes Bild für den Bildschirm hochladen, das während der Geräteinitialisierung nach dem Einschalten angezeigt wird.

HINWEIS
Standardmäßig wird auf dem Ladebildschirm ein vorinstalliertes Bild angezeigt.

Um ein benutzerdefiniertes Bild hinzuzufügen, doppelklicken Sie mit der linken Maustaste auf das Element „Ladebildschirm“ im Register „Anzeigemanager“. Das Fenster „Ladebildschirmbild“ wird geöffnet.



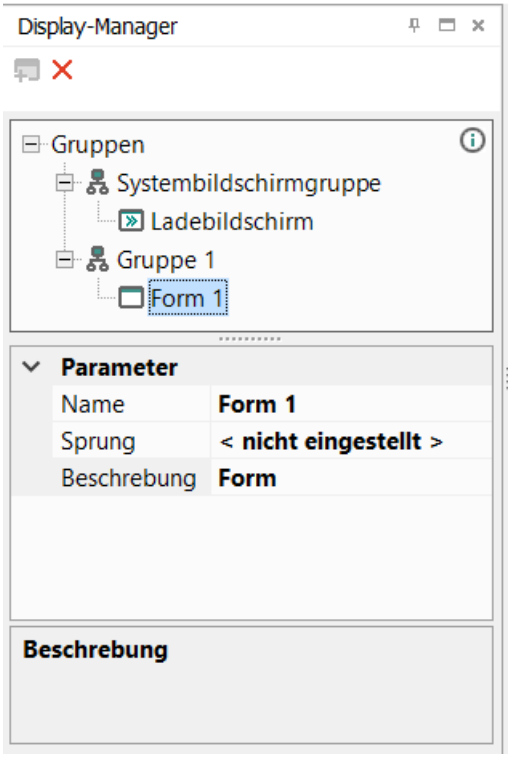
Gruppe	Parameter	Beschreibung	Zusätzliche Optionen
Koordi-naten	**X**	Gibt die X-Position des Elements an. Die Position wird durch die obere linke Ecke bestimmt. Der Wert beginnt bei 0 und steigt von links nach rechts. Der Maximalwert ergibt sich aus der Bildschirmbreite abzüglich der Elementbreite.	Die Position kann auch durch Ziehen des Elements auf der Bildschirmfläche festgelegt werden.
	Y	Gibt die Y-Position des Elements an. Die Position wird durch die obere linke Ecke bestimmt. Der Wert beginnt bei 0 und steigt von oben nach unten. Der Maximalwert ergibt sich aus der Bildschirmhöhe abzüglich der Elementhöhe.	
Daten	**Bildtyp**	– Bei Auswahl von „Standard“ wird das vorinstallierte Bild auf dem Startbildschirm angezeigt. Weitere Parameter sind deaktiviert. – Die Auswahl von „Benutzerdefiniert“ ermöglicht die Konfiguration der Bildparameter.	

Gruppe	Parameter	Beschreibung	Zusätzliche Optionen
	Bild	Wählen Sie die Bilddatei aus. Folgende Dateitypen stehen zur Auswahl: *.jpg, *.jpeg, *.jpe, *.bmp. Die maximale Bildauflösung ist durch die Bildschirmauflösung des Geräts begrenzt. Im ALP-Bildschirmeditor kann die Bildgröße verkleinert werden.	
Abmes-sungen	**Breite**	Gibt die Breite des Elements entlang der X-Achse an. Der Maximalwert darf die Breite des geladenen Bildes nicht überschreiten.	Die Elementgröße kann durch Anpassen der Objektgrenzen auf der Bildschirmfläche festgelegt werden. Breite und Höhe ändern sich proportional zu den Abmessungen des geladenen Bildes.
	Höhe	Gibt die Höhe des Elements entlang der Y-Achse an. Der Maximalwert darf die Höhe des geladenen Bildes nicht überschreiten.	

Nach dem Ladebildschirm wird standardmäßig der erste Bildschirm aus der Liste Gruppe 1 angezeigt.

Anzeigeformularparameter

Anzeigeformularparameter werden im Eigenschaftenbereich des **Anzeigemanagers** angezeigt.



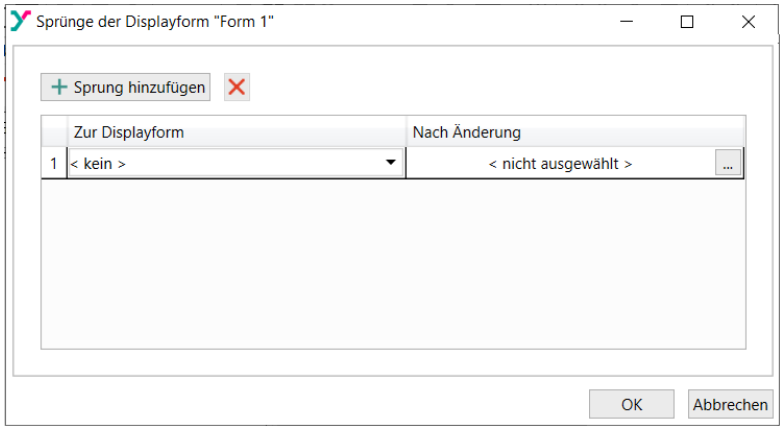
– **Name**: Wird im Display-Manager und in der Titelleiste des Anzeigeeditors angezeigt.
– **Sprung**: Dient zum Erstellen von Übergängen zwischen Anzeigeformularen.
– **Beschreibung**: Enthält eine Textbeschreibung des Anzeigeformulars.

Sprünge

So erstellen Sie einen Übergang (Sprung) zwischen den Display-Formen für ein LCD:

1. Klicken Sie mit der rechten Maustaste auf den Display-Form im Display-Manager-Baum und wählen Sie im Kontextmenü **Sprünge einstellen** oder klicken Sie in den Display-Eigenschaften

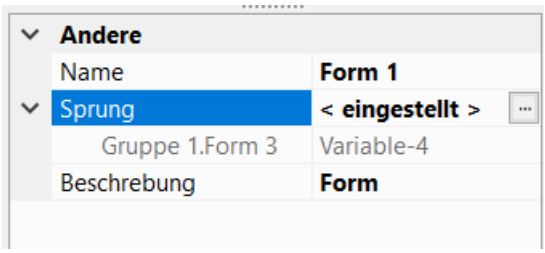
im Feld **Sprünge** auf die Schaltfläche **...**. Das Fenster „Sprungseditor“ wird geöffnet. Der ausgewählte Displayform dient als Ausgangspunkt für den Sprung.



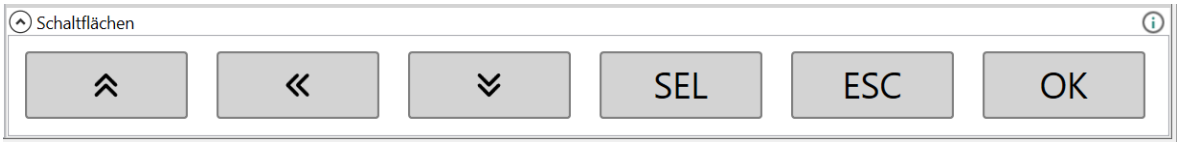
2. Wählen Sie aus dem Dropdown-Menü **Zur Displayform** eine andere Form aus, zu dem gewechselt werden soll.
3. Geben Sie in der Spalte **Variable ändern** durch Klicken auf die Schaltfläche **...** eine boolesche Variable an, deren Wertänderung den Übergang auslöst. Nach Ausführung des Übergangs wird der Wert der gebundenen Variable auf „0“ geändert.

HINWEIS
Informationen zum Erstellen eines per Schaltfläche ausgelösten Sprungs finden Sie unten.

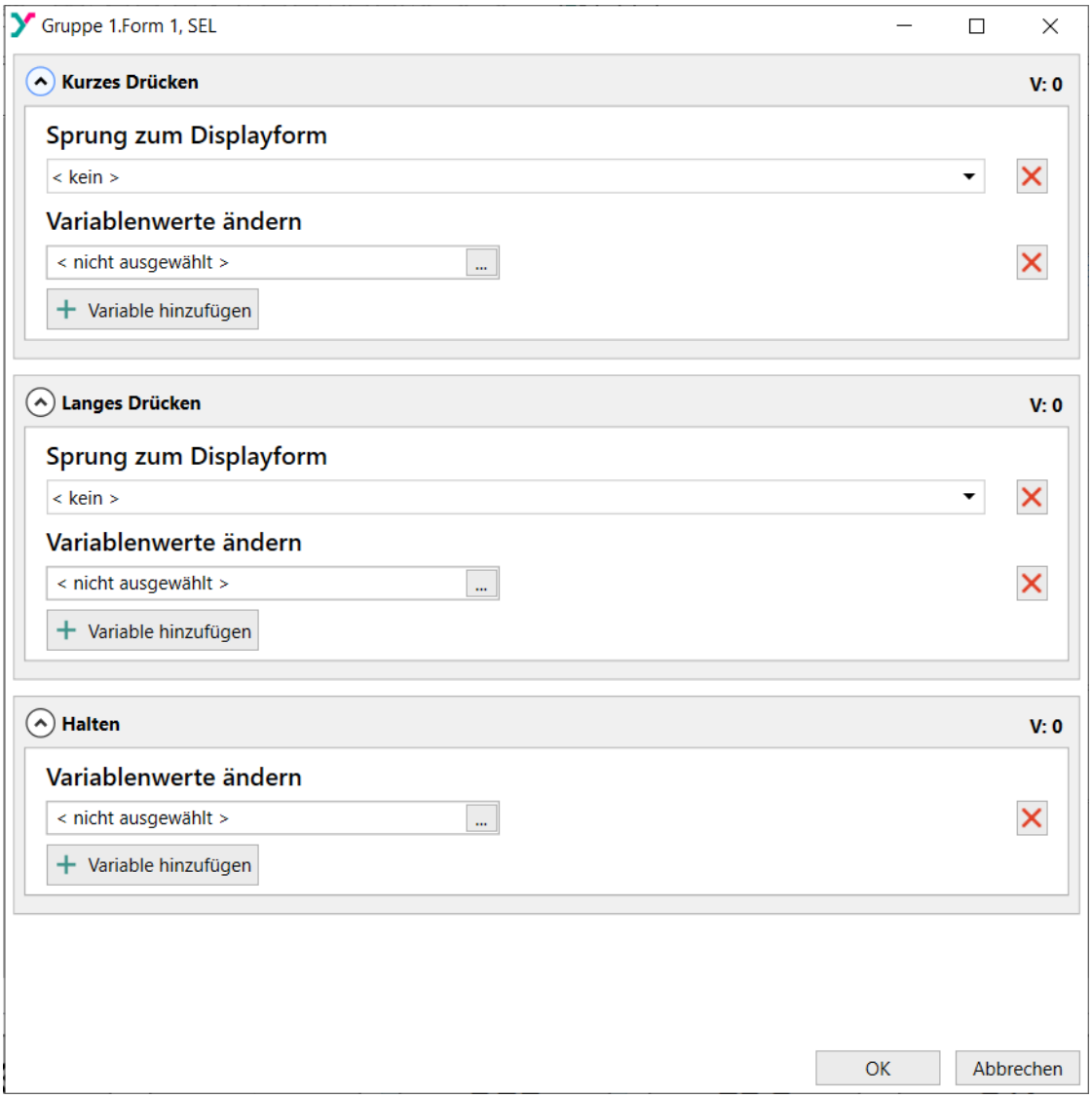
4. Um einen weiteren Sprung hinzuzufügen, klicken Sie auf + **Sprung hinzufügen**. Um einen ausgewählten Sprung zu löschen, klicken Sie auf ✕.
5. Klicken Sie auf **OK**. Die erstellten Übergänge werden in den Bildschirmeigenschaften angezeigt. Die Reihenfolge, in der die Sprünge ausgeführt werden, entspricht der Reihenfolge in den Displayformeigenschaften.



Durch einen Knopf zwischen Bildschirmen springen



Im unteren Bereich des Visualisierungseditors befindet sich ein Menü zur Programmierung der Gerätetasten. Durch Doppelklicken auf eine Taste öffnet sich das Fenster zur Ereignisprogrammierung der Taste. Die Tastenprogrammierung erfolgt für jede Displayform separat.



Für die Tastenprogrammierung stehen folgende Ereignisse zur Verfügung:

– Kurzes Drücken (weniger als 3 Sekunden)
– Langes Drücken (mindestens 3 Sekunden)
– Halten (mindestens 5 Sekunden)

Sie können mehrere Handler für kurze und lange Tastendrücke programmieren (Übergänge zu anderen Bildschirmen und Änderungen des Wertes boolescher Variablen). Nur ein Handler zum Ändern der Werte boolescher Variablen kann so programmiert werden, dass er hält. Handler werden im Programm in der Reihenfolge ausgeführt, in der sie hinzugefügt werden.

Um eine Variable hinzuzufügen, klicken Sie auf + ***Variable hinzufügen***. Um eine programmierte Aktion zu löschen, drücken Sie ✕.
Durch die Programmierung wird das Standardverhalten der Schaltflächen außer Kraft gesetzt. Um zum Standardverhalten der Schaltflächen zurückzukehren, löschen Sie die Ereignishandler und schreiben Sie das Programm auf das Gerät.
Wenn vom Benutzer keine Aktionen für die Schaltflächen ***Auf*** und ***Ab*** festgelegt wurden, wechselt das Gerät beim Drücken dieser Schaltflächen entsprechend der Reihenfolge im Bildschirmmanagerfenster zu anderen Bildschirmen.
Nach der Programmierung zeigt das Tastenmenü die Anzahl der Ereignisse für jede Taste an.

4.5 Simulation

Nutzen Sie die Simulation, um die Richtigkeit des erstellten Programms zu prüfen. Derzeit ist nur die Offline-Simulation 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.
Um den Simulationsmodus zu starten/stoppen, klicken Sie auf das Symbol ▶ in der Symbolleiste ***Service > Simulation***. Eine neue Symbolleiste ***Simulation*** wird angezeigt.

Symbolleiste „Simulation“

▶	**Start**	Startet die Simulation.
↪	**Einzelzyklus**	Führt die Simulation schrittweise aus und startet einen einzelnen Programmzyklus.
❚❚	**Pause**	Pausiert die Simulation.
■	**Stop**	Stoppt die Simulation.
	Aktualisierung	Ermöglicht das Einstellen der Aktualisierungsperiode der Informationen im Schema in Millisekunden.
	Zykluszeit	Ermöglicht das Einstellen der Zykluszeit der Programmausführung im Simulationsmodus. Die Zeit kann in Millisekunden, Sekunden, Minuten oder Stunden angegeben werden.
	Beobachtungs-fenster	Öffnet oder schließt das Beobachtungsfenster zur Anzeige der Variablenwerte bei jedem Programmschritt.
	Anzeigesimulation	Öffnen Sie das Anzeigesimulationsfenster

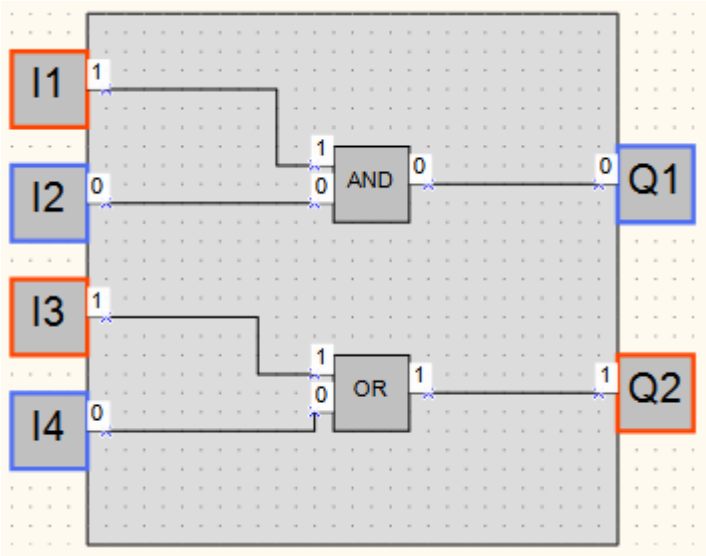
VORSICHT
*Die Parameter **Zykluszeit** im ALP-Simulationsmodus und **Zykluszeit** im Gerät sind trotz gleichem Namen unterschiedlich. Die **Zykluszeit** kann beliebig eingestellt werden und ist erforderlich, um die Funktionsweise von Timing-Funktionsblöcken zu simulieren. – TON, TOF, BLINK. Alle Zeitparameter dieser Funktionsbausteine werden während der Zwischenschritte des Simulators verarbeitet.*

Simulation von Taktblöcken

Bei Geräten mit Echtzeituhrfunktion öffnet sich ein zusätzliches Bedienfeld. Dieses Bedienfeld ermöglicht es Ihnen, das Verhalten des *CLOCK* und *CLOCK WEEK*. Die Funktion blockiert mit der Zeit.

Ablauf der Simulation

1. Führen Sie die Simulation in einem der Modi aus: Echtzeit (▶) oder Schritt-für-Schritt (↪).
2. Legen Sie die Eingangswerte für Programmblöcke fest.



3. Wählen Sie aus praktischen Gründen die Werte der Parameter ***Aktualisierung***, ***Zykluszeit*** und **Zykluseinheitszeit** für eine Simulation aus.
4. Um das Programm zu korrigieren, beenden Sie den Simulationsmodus.

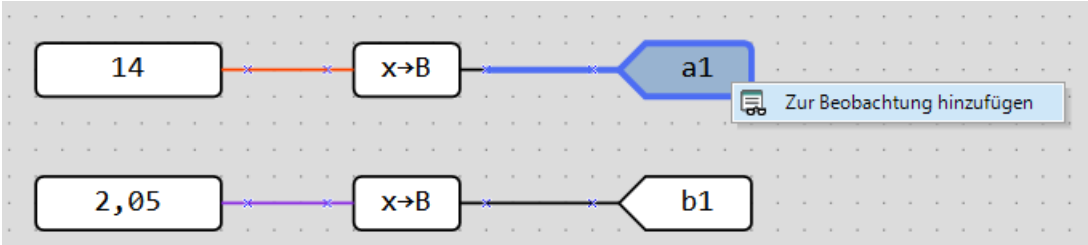
Der Simulationsmodus funktioniert nach folgenden Regeln:

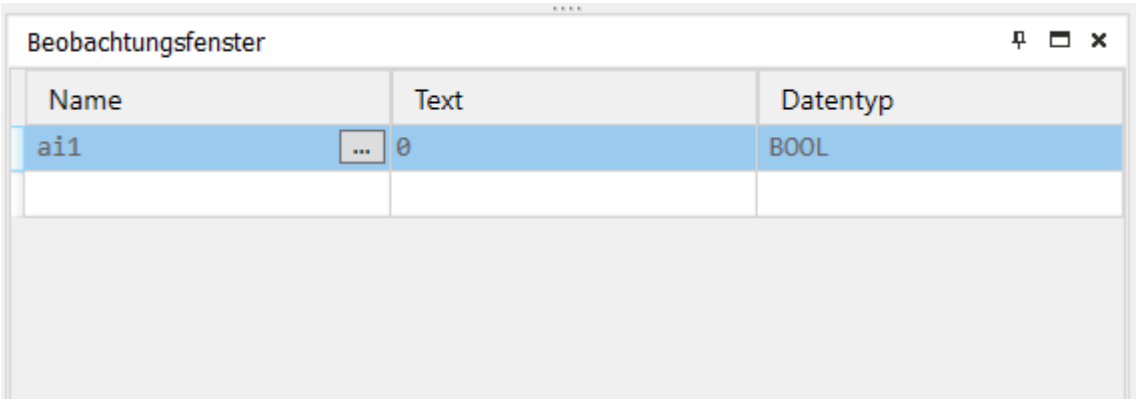
– Das Programm (auf der Registerkarte „Programm“) und die Makros werden separat simuliert.
– Der Betrieb von Blöcken, die nicht mit einem der Ausgänge des Geräts oder mit einem Ausgabeblock einer Netzwerkvariablen verbunden sind, wird nicht simuliert.
– Die Simulation funktioniert nicht bei volatilen Variablen oder Variablen, die falsch verknüpft sind.

Beobachtungsfenster

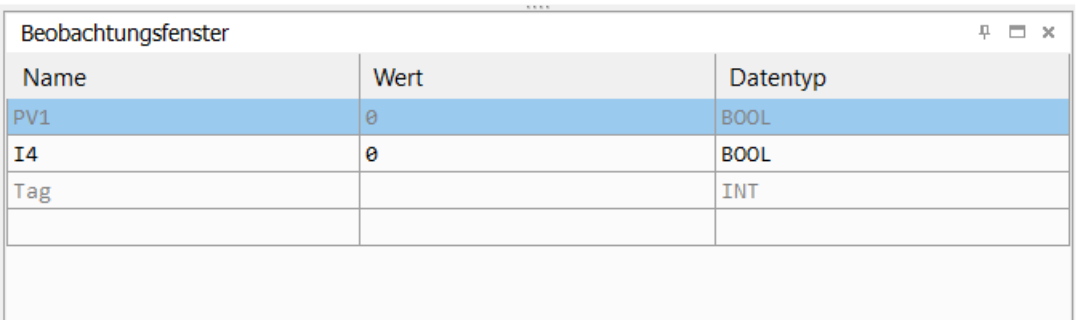
Klicken Sie auf das Symbol in der Simulationssymbolleiste, um die Eingangs-, Ausgangs- oder Variablenwerte bei jedem Programmschritt zu beobachten.

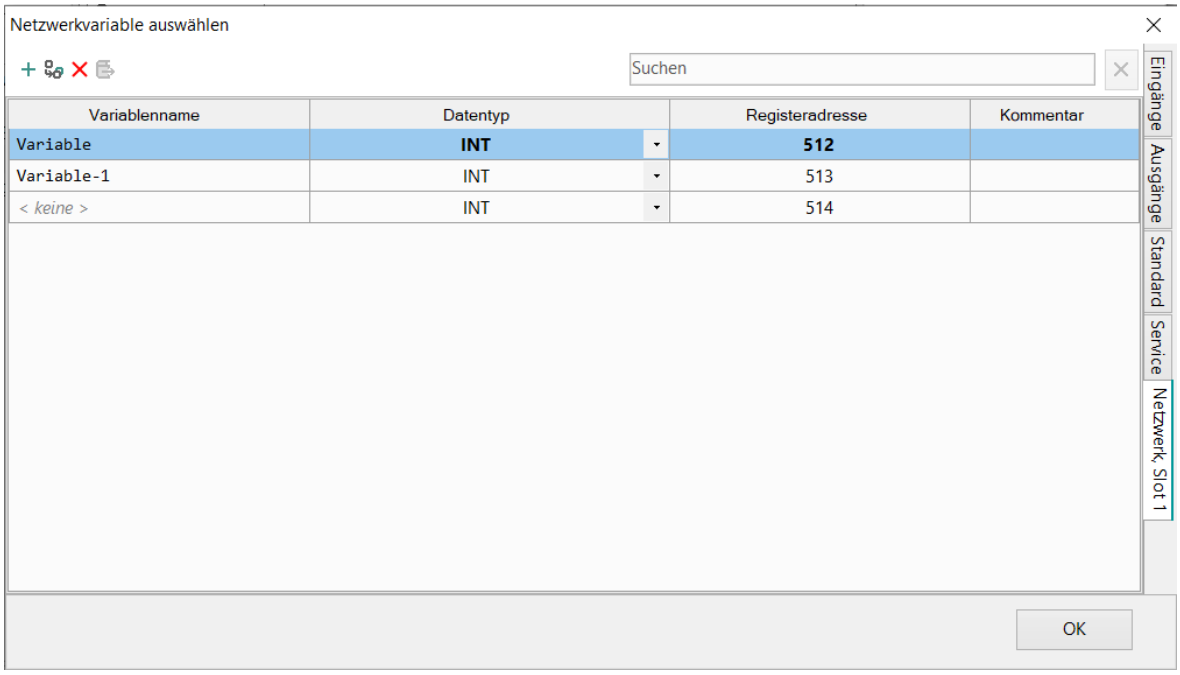
Um der ***Beobachtungsliste*** einen Eingang, einen Ausgang oder eine Variable hinzuzufügen, klicken Sie auf das leere Feld in der Spalte ***Name*** und dann auf das links angezeigte Symbol «**...**».





Es öffnet sich die ***Variablentabelle***. Hier können Projektvariablen, Ein- und Ausgänge ausgewählt werden

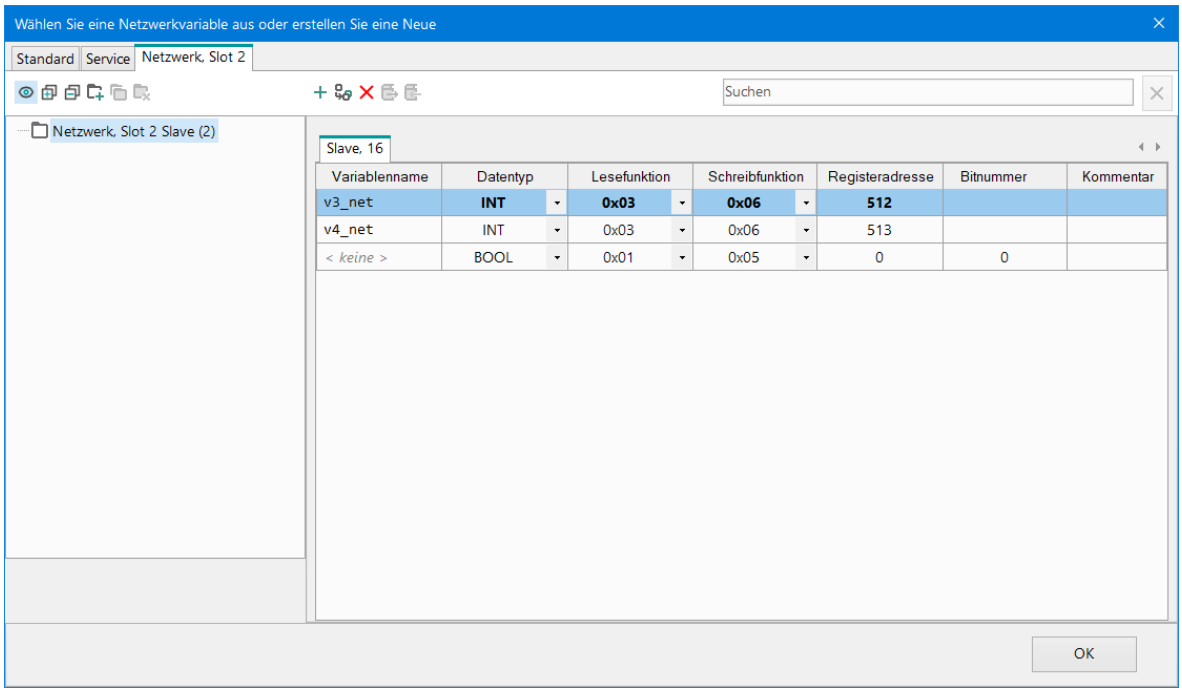




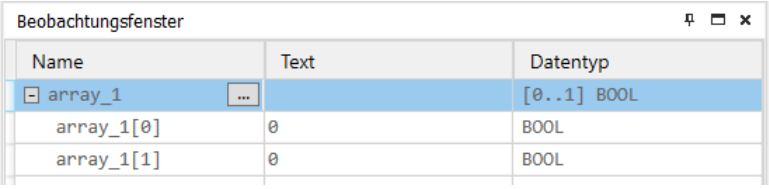
Die ausgewählten Variablen werden dem Beobachtungsfenster hinzugefügt.

Auch das Baustein-Kontextmenü kann genutzt werden.
Die Werte der Variablen, Eingänge und Ausgänge können während der Simulation in der Spalte ***Wert*** eingestellt werden.
Einschränkungen bei der Arbeit mit einer Variablentabelle, die aus dem Variablenanzeigefenster geöffnet wurde:

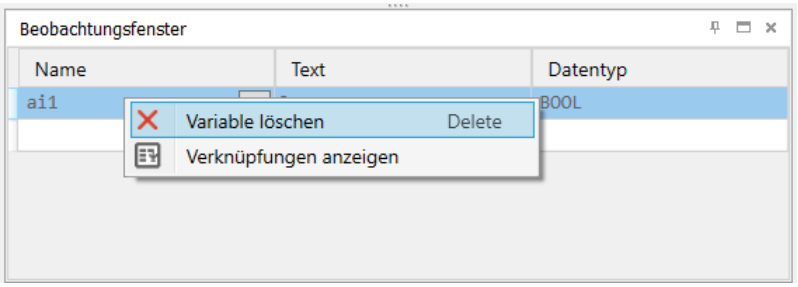
– Das Erstellen, Duplizieren und Löschen von Variablen ist nicht möglich
– Eine Änderung des Namens, des Datentyps und der Parameter für Energieunabhängigkeit ist nicht möglich
– Für Standardvariablen stehen die Konfiguration von Array-Parametern sowie die Funktionen zum Importieren und Exportieren von Variablen nicht zur Verfügung
– Für Netzwerk-Slave-Variablen sind keine Parametereinstellungen verfügbar



Wählen Sie die Variablen aus und klicken Sie auf die Schaltfläche. **OK**. Die ausgewählten Variablen werden dem Ansichtsfenster hinzugefügt.
99 Der Variablenanzeigetabelle wird eine Array-Variable zusammen mit den Array-Elementen hinzugefügt. In diesem Fall ist es möglich, einzelne Array-Elemente hinzuzufügen, selbst wenn das entsprechende Array bereits in der Tabelle des gesamten Anzeigefensters vorhanden ist.



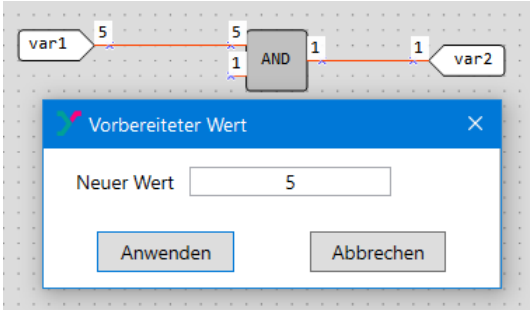
Unbenutzt Die Variablen werden im Anzeigefenster grau dargestellt, ihr Wert wird immer durch die Zahl 0 angegeben, eine Eingabe eines Wertes für diese Variablen ist nicht möglich.
Um eine Variable aus dem Ansichtsfenster zu löschen, wählen Sie die Variable aus und drücken Sie die Entf-Taste oder verwenden Sie das Kontextmenü der Variable. **Ansichtsfenster**:



Es ist nicht möglich, ein Array-Element zu löschen, das innerhalb eines anderen Arrays verschachtelt ist.

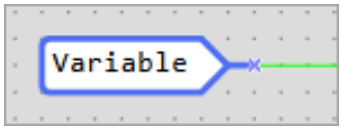
Eingabe von Variablenwerten

Im Simulatormodus können Variablenwerte im Vorschaufenster in der Spalte eingestellt werden. **Wertes**.
Variablen können auch im Diagramm Werte haben. Durch Doppelklicken auf eine Variable öffnet sich ein Fenster mit einem Feld zur Eingabe eines neuen Wertes.

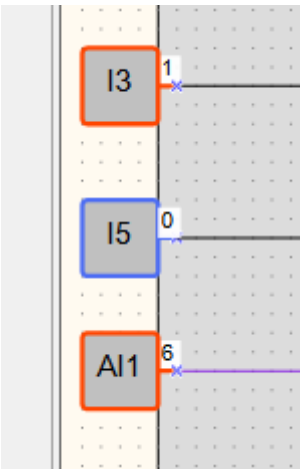


Nach Eingabe eines Variablenwerts ungleich Null ändert der Block im Diagramm für einen Programmdurchlauf seine Farbe in Rot. Wenn die Simulation gestoppt wird, bleibt der Block rot, bis die Simulation gestartet oder ein Befehl ausgeführt wird. **Schritte pro Zyklus:**

Nach Eingabe des Wertes Null für eine Variable (bei einer Array-Variablen muss der Wert 0/False für alle Array-Elemente gelten) ändert der Block im Diagramm seine Farbe für einen Programmdurchlauf zu Blau. Nach dem Stoppen der Simulation bleibt der Block blau, bis die Simulation erneut gestartet oder der entsprechende Befehl ausgeführt wird. **Schritte pro Zyklus:**



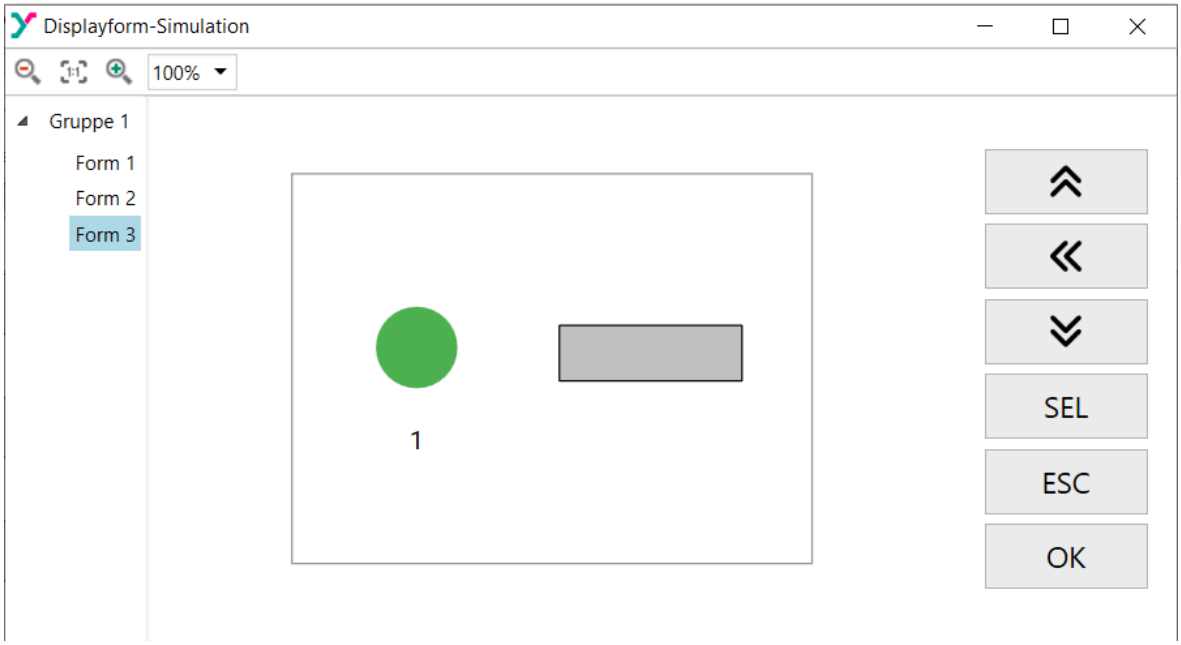
Im Simulationsmodus können Sie die Werte der Geräteeingänge durch Anklicken änderneingestellt. Bei analogen Eingängen wird der Wert ähnlich wie bei Variablen in einem Fenster mit Eingabefeld festgelegt.



4.5.1 Visualisierung-Simulation

Im Simulatormodus für Geräte mit grafischem Farb-LCD steht die Simulation des Gerätedisplays und der Tasten zur Verfügung. Nach dem Aufrufen des Modus öffnet sich ein Fenster mit dem Startbildschirm des Geräts.

HINWEIS
Das *Diagrammelement* wird im Simulationsmodus nicht unterstützt.



Zoom-Schaltflächen befinden sich in der oberen linken Ecke des Fensters.
Auf der linken Seite befindet sich eine Liste der im Projekt verfügbaren Displayformen.
In der Mitte des Fensters wird eine Simulation des Gerätedisplays angezeigt.
Auf der rechten Seite befinden sich Schaltflächen zur Simulation von Aktionen mit echten Schaltflächen des Geräts.

Arbeiten mit dem Bildschirm

Folgende Visualisierungselemente können im Simulationsmodus bearbeitet werden:
- *Dynamische Box*
- *REAL/INT Ein-/Ausgabe*
- *Zeiteingabe/-ausgabe*
- *IP-Adresse*

Arbeiten mit den Schaltflächen

Die Schaltflächen im rechten Teil des Fensters dienen zur Simulation von Aktionen mit echten Schaltflächen des Geräts.
Die Schaltflächenaktionen werden für jeden Bildschirm festgelegt. Schaltflächenaktionen werden für den Bildschirm verarbeitet, der aktuell im Hauptbereich des Fensters angezeigt wird.
Simulationstasten können mit den Tasten ***Q, W, E, A, S, D, Z, X, C*** gesteuert werden. Das Drücken von Tasten auf der Tastatur ist nur möglich, wenn das Visualisierungssimulationsfenster im Fokus ist.
Bei einer schrittweisen Simulation ist das Halten der Taste nicht aktiviert.

4.5.2 ST-Code-Simulation

Die Simulation von Funktions- oder Funktionsblockcode in ST ist in ALP verfügbar. Öffnen Sie dazu den **Funktionseditor** (oder **Funktionsblockeditor**) und klicken Sie in der Symbolleiste auf die Schaltfläche ▶ oder im Hauptmenü auf **Service** → **Simulatormodus**. Das Simulationsfenster wird geöffnet. Im **Editor** werden die Variablenwerte in den Anzeigefeldern neben der Variable angezeigt. Wenn im Programm Arrays deklariert sind, wird im Anzeigefenster der Eintrag **Ändern** angezeigt.

Hauptprogramm* | **Funktionsblock-Editor "MaxI_MinI"**

```
1  FUNCTION_BLOCK MaxI_MinI
2  //Maximale und minimale Zahlen mithilfe der Bubblesortierung vom Kleinsten zum Größten
3
4      VAR_INPUT //Deklaration von Eingabevariablen
5          udiX1 [ 0 ], udiX2 [ 0 ], udiX3 [ 0 ], udiX4 [ 0 ], udiX5 [ 0 ] : UDINT; //Hinzufügen der
6      END_VAR
7
8      VAR_OUTPUT //Deklaration von Ausgabevariablen
9          udiMaxI [ 0 ], udiMinI [ 0 ] : UDINT;
10     END_VAR
11
12     VAR //Deklaration lokaler Variablen
13         udiI [ 5 ], udiJ [ 2 ], udiN [ 5 ], udiK [ 0 ] : UDINT;
14         audiX [ Ändern ] : ARRAY [1..5] OF UDINT; //Gibt den Bereich des Arrays und den Datentyp des Arr
15     END_VAR
16
17     //Codebereich
18     //Array-Deklaration
19     audiX[1] [ 0 ] := udiX1 [ 0 ];
20     audiX[2] [ 0 ] := udiX2 [ 0 ];
21     audiX[3] [ 0 ] := udiX3 [ 0 ];
22     audiX[4] [ 0 ] := udiX4 [ 0 ];
23     audiX[5] [ 0 ] := udiX5 [ 0 ];
24     udiN [ 5 ] := 5; // die Anzahl der zu sortierenden Zahlen wird angegeben (Array-Größe)
```

Farbcodierung der Ansichtsfenster:

[13.3] – Der aktuelle Wert der Variable
[12] – Der Wert der Variable wurde vom Benutzer geändert
[13.3] – Der Wert der Variable wurde vom Benutzer festgelegt
[Ändern] – Der Wert eines oder mehrerer Elemente des Arrays (nicht aller) wurde vom Benutzer festgelegt
[???] – Fehler: Falscher Variablenwert (z. B. wenn mithilfe einer Variablen ein nicht vorhandenes Array-Element angegeben wird)
Wenn Sie im Simulationsmodus mit der linken Maustaste in das Ansichtsfenster klicken, um den Wert einer Integer-Variablen (udint) oder einer Gleitkomma-Variablen (real) einzugeben, erscheint ein Fenster, in dem Sie einen neuen Wert der Variable eingeben und den Wert gegebenenfalls fixieren können.

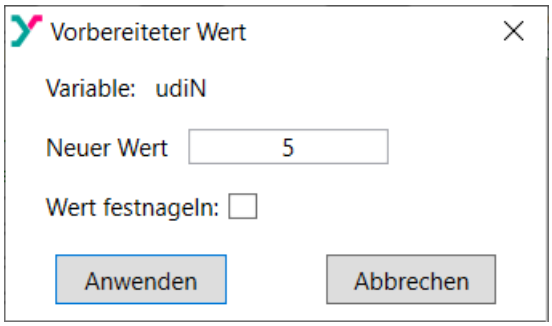


Abb. 4.3 Ändern des Werts einer UDINT-Variablen

Wenn Sie im Simulationsmodus mit der linken Maustaste in das Ansichtsfenster klicken, um den Wert einer booleschen Variable (bool) einzugeben, erscheint ein Fenster, in dem Sie „TRUE“ oder „FALSE“ auswählen und den Wert gegebenenfalls fixieren können.

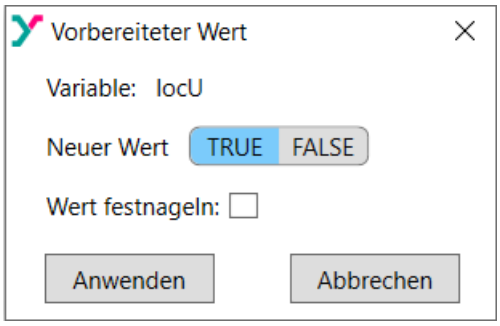


Abb. 4.4 Ändern des Werts einer BOOL-Variablen

Wenn Sie im Simulationsmodus mit der linken Maustaste in das Ansichtsfenster klicken, um den Wert einer Zeitvariablen (Zeit oder DT) einzugeben, erscheint ein Fenster, in dem Sie den Wert im Format T#0d0h0m0s0ms (für Zeit) oder DT#2000-01-01-0:00:00 (für dt) eingeben und gegebenenfalls korrigieren können.

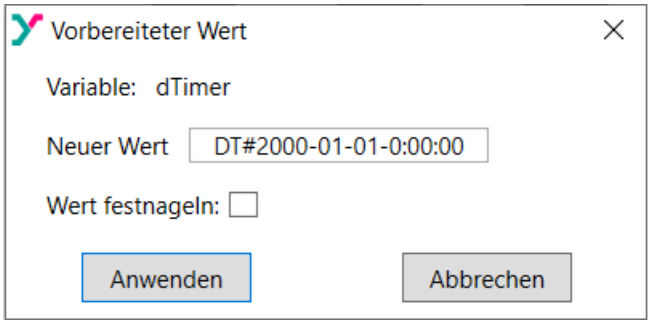


Abb. 4.5 Ändern des Werts einer Zeitvariablen

HINWEIS
Die Zeit wird automatisch neu berechnet, nachdem die Schaltfläche **Anwenden** gedrückt wurde. Beispielsweise sieht der eingegebene Wert der Variable T#80m nach erneutem Öffnen des Fensters **Wert ändern** wie folgt aus: T#1h20m.

Wenn kein oder ein falscher Wert eingegeben wird, wird eine Fehlermeldung angezeigt. Die Schaltfläche **Anwenden** ist deaktiviert.

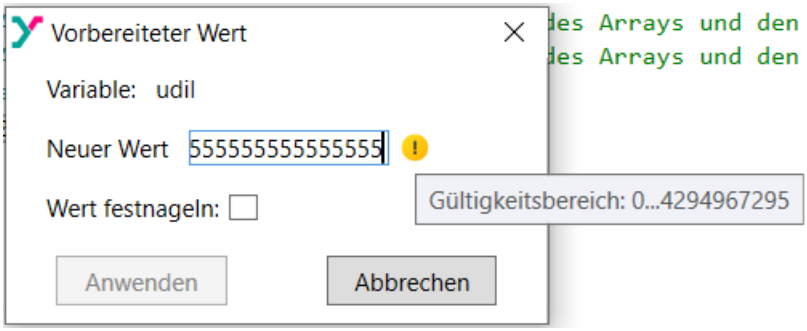


Abb. 4.6 Werteingabefehler

Um Werte in die Array-Elemente einzugeben, klicken Sie mit der linken Maustaste auf das Ansichtsfenster mit der Bezeichnung **Ändern**:

– Wenn der Elementindex durch eine Konstante oder mithilfe einer Variable angegeben wird, wird das Fenster „Wert für bestimmtes Array-Element ändern“ angezeigt.
– Wird der Index eines Array-Elements als Ergebnis eines mathematischen Ausdrucks oder als Ergebnis einer Funktion/eines Funktionsbausteins angegeben, erscheint ein Fenster mit allen zur Eingabe verfügbaren Elementen des Arrays und Variablenwerten:

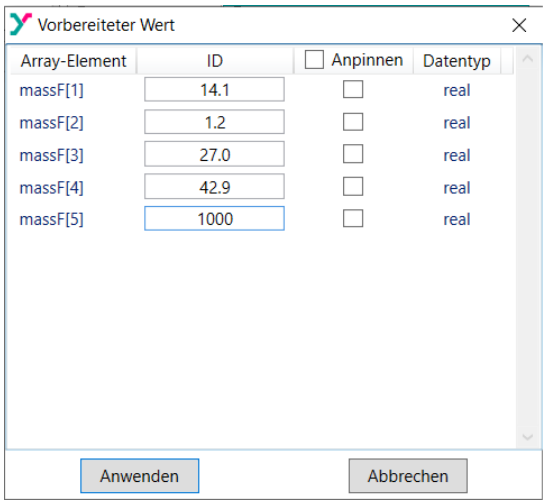


Abb. 4.7 Ändern der Werte von Array-Elementen mit REAL-Variablen

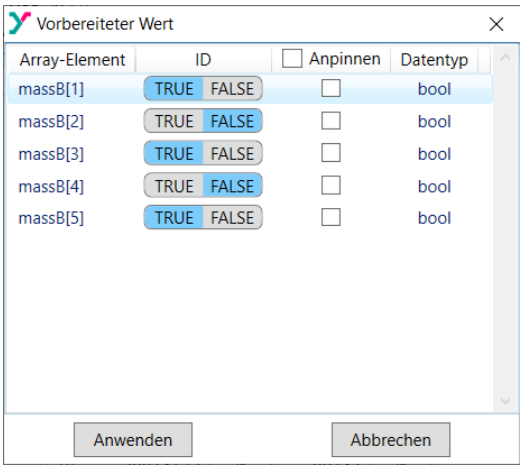


Abb. 4.8 Ändern der Werte von Array-Elementen mit BOOL-Variablen

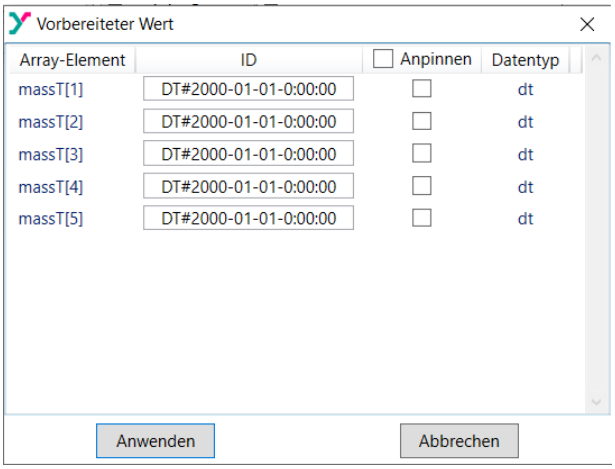


Abb. 4.9 Ändern der Werte von Array-Elementen mit DT-Variablen

Wenn kein oder ein falscher Wert eingegeben wird, wird eine Fehlermeldung angezeigt. Die Schaltfläche **Akzeptieren** ist deaktiviert.

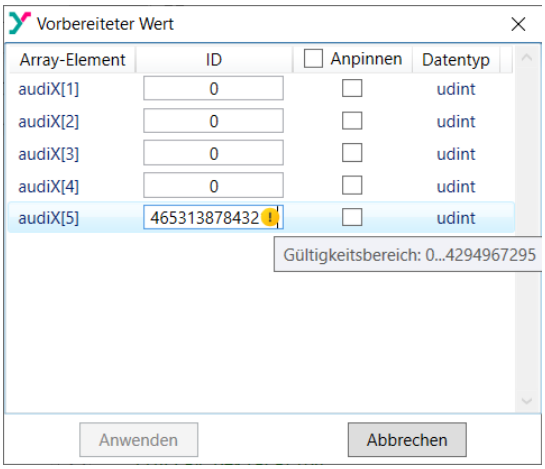


Abb. 4.10 Werteingabefehler

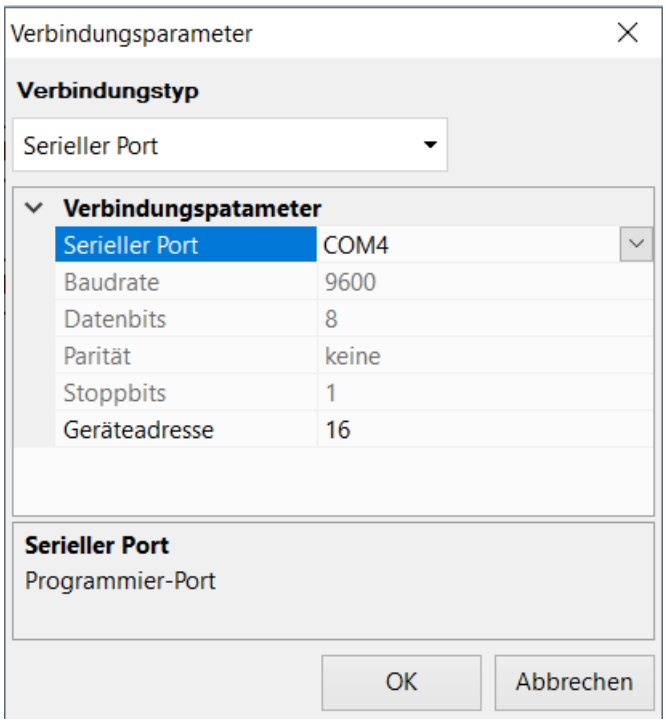
4.6 Verbindung zum Geraet

VORSICHT
Das Gerät muss vor dem Verbinden mit dem PC ausgeschaltet werden.

Alle Geräte können über USB an den PC angeschlossen werden. Wenn das Gerät über eine Ethernet-Schnittstelle verfügt, kann es über Ethernet verbunden werden. Um die Verbindung vorübergehend zu unterbrechen, verwenden Sie den ***OFFLINE-Modus***.

Verbindung über USB

HINWEIS
Für Geräte der zweiten Generation (PR103, PR205, PR225) mit Anschluss an RS-485 mit USB-Konverter IC4M. In diesem Fall wurde eine Datei mit einer Geschwindigkeit von 9600 Bit/s erstellt.

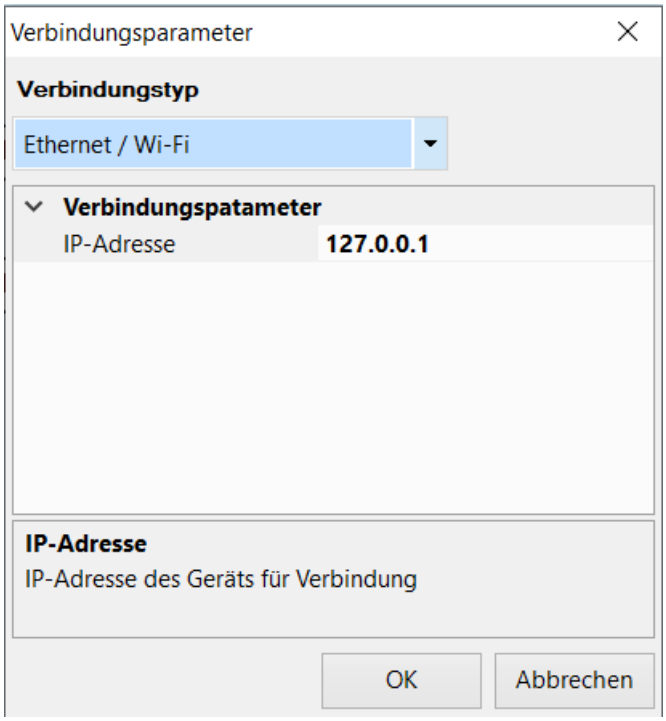


Geräte können über USB an den PC angeschlossen werden.

1. Schließen Sie das Gerät an einen USB-Anschluss des PCs an und schalten Sie das Gerät ein.
2. Starten Sie ALP und wählen Sie den Menüpunkt ***Gerät > Porteinstellungen***.
3. Wählen Sie ***Serieller Port*** für ***Verbindungstyp***.
4. Wählen Sie im geöffneten Dialog den seriellen Port aus. Die Nummer des emulierten COM-Ports finden Sie im Windows-Gerätemanager unter „Anschlüsse (COM und LPT)“.
5. Geben Sie die ***Geräteadresse*** (standardmäßig 16) ein und bestätigen Sie mit ***OK***. Alle weiteren Parameter werden nur zu Ihrer Information angezeigt.

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

Verbindung über Ethernet



Um das Gerät über die Ethernet-Schnittstelle mit einem PC zu verbinden, beachten Sie die folgenden Schritte:

1. Verbinden Sie das Gerät mit demselben lokalen Netzwerk wie den PC.
2. Ermitteln Sie die IP-Adresse des angeschlossenen Geräts. Die Standard-IP-Adresse ist im Benutzerhandbuch des Geräts angegeben. Die aktuelle IP-Adresse des Geräts kann mit der Software ausgelesen werden..
3. Wählen Sie ***Ethernet / Wi-Fi*** als ***Verbindungstyp***.
4. Geben Sie die ***IP-Adresse*** des verbundenen Gerätes ein und bestätigen Sie mit ***OK***.

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

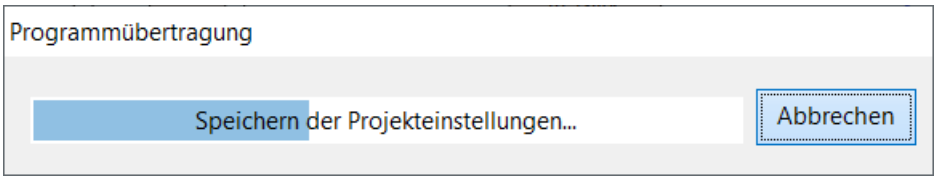
4.7 Projekt auf Geraet uebertragen

VORSICHT
Wenn ein neues Projekt auf das Gerät hochgeladen wird, wird das bereits im Gerätespeicher (ROM) gespeicherte Programm durch das neue ersetzt.

Gehen Sie wie folgt vor:

1. *Verbinden* Sie das Gerät mit dem PC.
2. Schalten Sie das Gerät ein.
3. Passen Sie bei Bedarf die Porteinstellungen an.
4. Laden Sie das Projekt auf das Gerät hoch.

   Über den Menüpunkt ***Datei > Anwendung auf Gerät übertragen*** oder durch Klicken auf das Symbol in der Symbolleiste kann das Projekt auf das Gerät hochgeladen werden. Nach Abschluss des Hochladens kann das Gerät ausgeschaltet und vom PC getrennt werden.



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

HINWEIS
Wenn die Programmübertragung abgeschlossen ist, geht das Gerät in den Betriebsmodus und das Programm startet automatisch.

OFFLINE-Modus

Im ***OFFLINE-Modus*** wird die Verbindung zwischen ALP und dem Gerät unterbrochen.
Der Modus ist hilfreich, wenn Sie mit zwei ALP-Instanzen auf dem PC arbeiten und versuchen, mit demselben Gerät zu kommunizieren. Beide Anwendungen belegen abwechselnd den Port und die Verbindung zum Gerät wird ständig unterbrochen.
Die ALP-Instanz, die nicht mit dem Gerät interagieren soll, muss in den OFFLINE-Modus versetzt werden.
Der OFFLINE-Modus kann über den Menüpunkt ***Service > OFFLINE-Modus*** oder durch einen Klick auf die *Statusanzeige* ***Gerät*** aktiviert/deaktiviert werden. Mit dem nächsten Klick wird der OFFLINE-Modus deaktiviert.

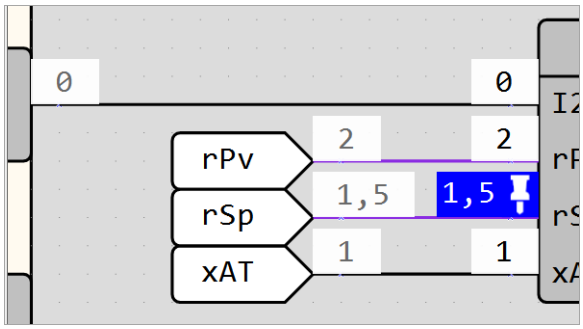
4.8 Online-Debugging

Der Online-Debug-Modus ist für das Testen des Programms in Echtzeit mit tatsächlichen Werten der Geräteeingänge konzipiert. Online-Debugging ist nur möglich, wenn das Gerät an einen PC angeschlossen ist. Im Online-Debugging-Modus können keine Änderungen am Projekt vorgenommen werden.

Ablauf und Regeln der Arbeit im Online-Debugging-Modus

Online-Debugging kann erst durchgeführt werden, nachdem das Programm auf das Gerät geladen wurde. Das Programm im Gerät muss mit dem Programm im Projektdiagramm übereinstimmen. Das Gerät muss die entsprechende Version aufweisen. *Firmware*.
Für die Online-Debugging sollten Sie Folgendes tun:

1. Starten Sie das Online-Debugging durch Klicken auf die Schaltfläche In der Symbolleiste wird das Online-Debugging-Panel geöffnet.
2. Überprüfen Sie die Logik des Programmablaufs mit realen Werten an den Geräteeingängen, die im Diagramm dargestellt sind.

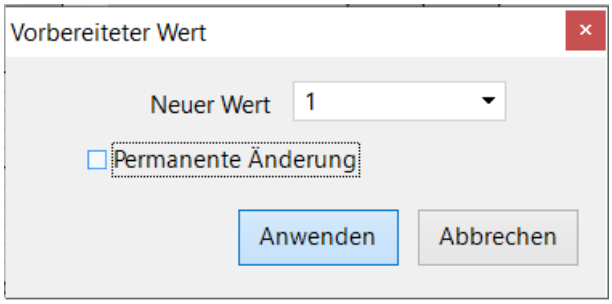


3. Um Anpassungen vorzunehmen, verlassen Sie den Online-Debugging-Modus durch Drücken der Taste. .

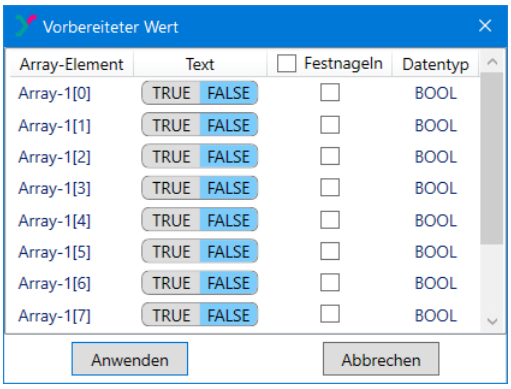
Online-Debugging ist nur für das Projektdiagramm verfügbar (um die Funktionsweise von Makros, Funktionen und Funktionsbausteinen in der ST-Sprache zu überprüfen, sollten Sie verwenden). *Simulationsmodus*).

Manuelle Zuordnung von Werten

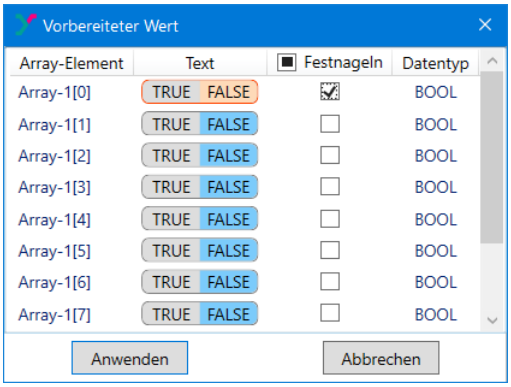
Im Online-Debugging-Modus können Sie die erforderlichen Werte für Ausgabevariablenblöcke sowie für Funktionseingaben, Funktionsblöcke und Makros festlegen. Um im Online-Debugging-Modus einen Wert festzulegen, klicken Sie auf die Variable-Wert-Platte und geben Sie den neuen Wert in das sich öffnende Fenster ein.



Bei einer Variablen vom Typ Array können die Werte aller Array-Elemente eingegeben werden:

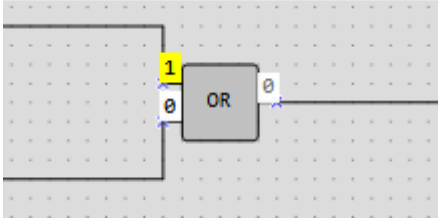


Die im Fenster „Wert ändern“ eingegebenen Werte eines Array-Variablenelements werden orange hervorgehoben:

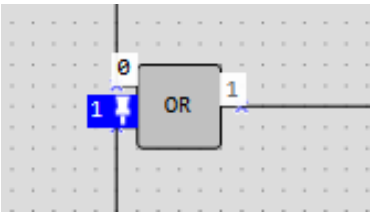


Werte in einem Diagramm anzeigen

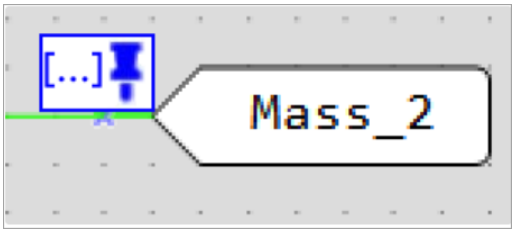
Wenn Sie das Kästchen nicht ankreuzen **Wert festnageln**Der neue Wert ist dann für einen Arbeitszyklus gültig. Am Ende des aktuellen Zyklus oder zu Beginn des nächsten kann der angegebene Wert überschrieben werden. Während der Aktion wird der eingestellte Wert gelb hervorgehoben.



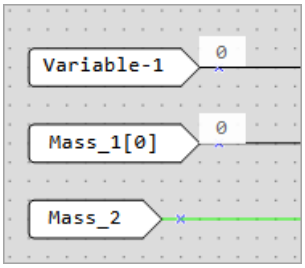
Wenn Sie einen festgelegten Wert sperren, bleibt dieser Wert in jedem Arbeitszyklus so lange aktiv, bis Sie ihn manuell auf einen anderen Wert ändern oder die Sperre aufheben. Die Wertkorrektur ist beim Verlassen des Online-Debugging-Modus deaktiviert. Bei Variablen vom Typ Array ist es möglich, einen, mehrere oder alle Werte der Array-Elemente festzulegen. Der feste Wert (bei Arrays – Festlegung der Werte aller Elemente) wird im Diagramm als blaues Kästchen dargestellt:



Wenn Sie den Wert eines oder mehrerer Elemente des Arrays (nicht aller) festlegen, wird die Leiste weiß mit blauem Rand angezeigt:

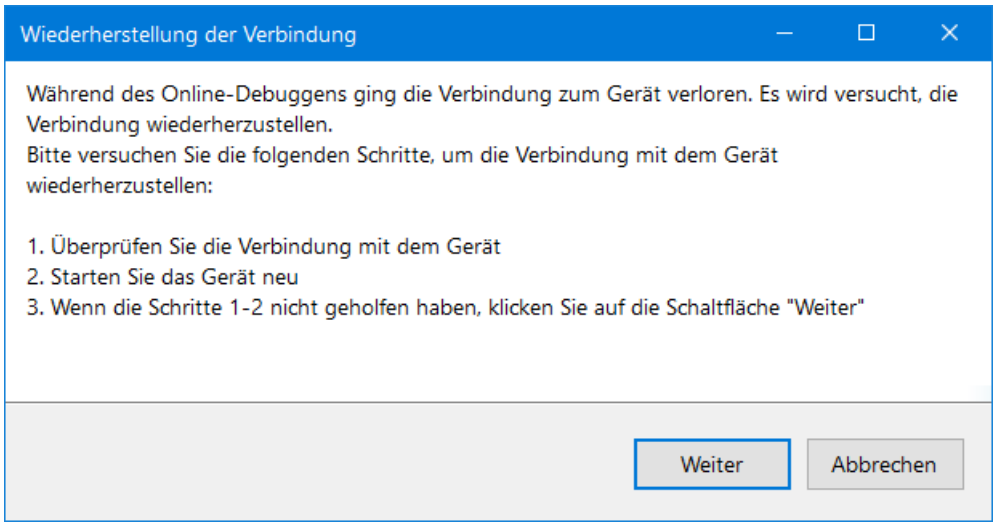


Es ist nicht möglich, die Werte der Eingangsvariablenblöcke festzulegen, daher werden die Werte dieser Eingangsvariablen im Diagramm grau dargestellt. Bei Eingabeblöcken mit Variablen vom Typ Array wird der Wert nicht angezeigt.



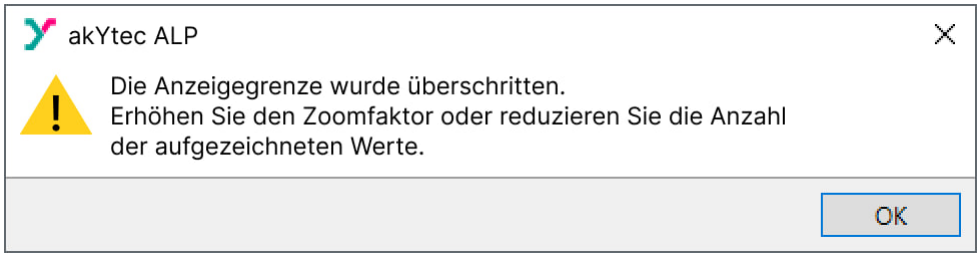
Fehlerbehebung

Wenn die Verbindung zum Gerät abbricht, wird der Online-Debugging-Modus nach 10 Sekunden deaktiviert und das Gerät schaltet in den Betriebsmodus. Wenn Sie die Verbindung wiederherstellen, wird das Online-Debugging fortgesetzt, jedoch werden zuvor aufgezeichnete Werte deaktiviert.



Grenzwert der übertragenen Werten

Jede Gerätemodifikation hat ihr eigenes Limit für den Speicherpuffer, der für die Datenübertragung im Online-Debugging-Modus verwendet wird. Wenn das Diagramm leere Balken anzeigt, wird eine Einschränkung ausgelöst. Sie sollten den Diagrammmaßstab erhöhen, um die Anzahl der im Diagrammfenster angezeigten Werte zu reduzieren. Festgelegte Werte bleiben auch dann festgelegt, wenn sie nicht im Schaltplanfenster angezeigt werden, und belegen weiterhin den Speicherpuffer. Wenn der Speicherpuffer des Geräts vollständig gefüllt ist, öffnet ein Klick auf eine leere Platte nicht das Fenster „Wert ändern“. Stattdessen erscheint eine Informationsmeldung über das Überschreiten des Grenzwerts der angezeigten Werte:



Stattdessen erscheint eine Informationsmeldung über das Überschreiten des Grenzwerts der angezeigten Werte:

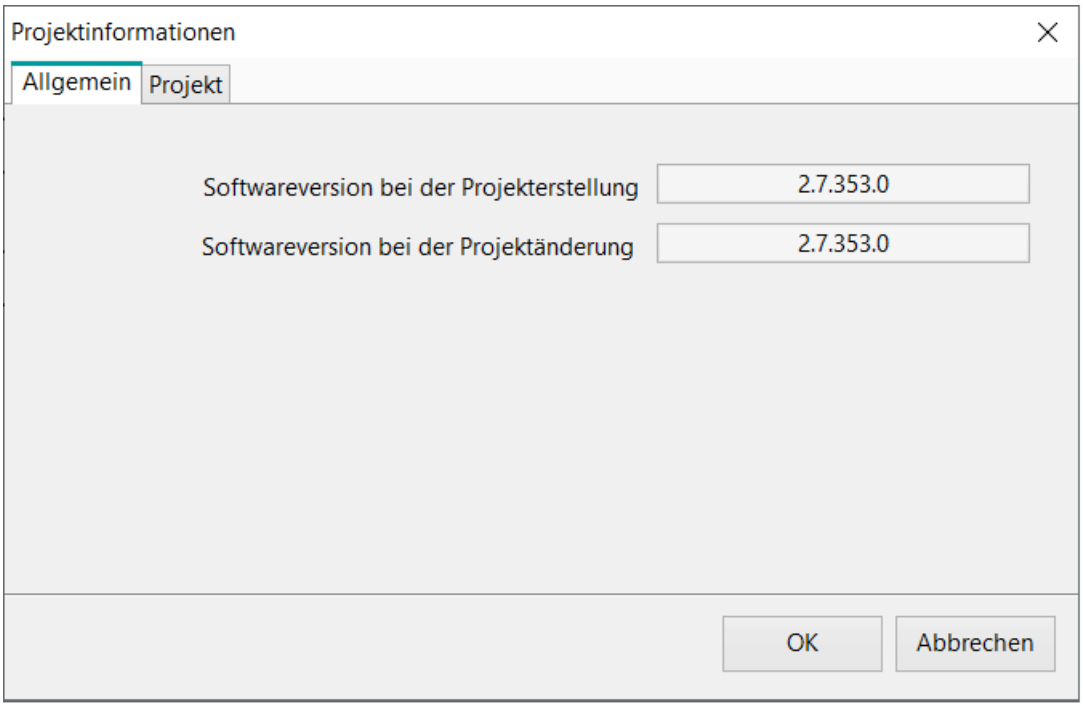
– Der festgelegte Wert wird im Fenster „Wert ändern“ angezeigt und kann geändert oder freigegeben werden.
– Die Werte nicht zugewiesener Array-Elemente im Fenster „Wert ändern“ werden nicht angezeigt und können nicht eingegeben werden.

4.9 Projektinformationen

Über den Menüpunkt ***Datei*** → ***Projektinformationen*** können Sie die Informationen zum Programm einsehen und ändern.

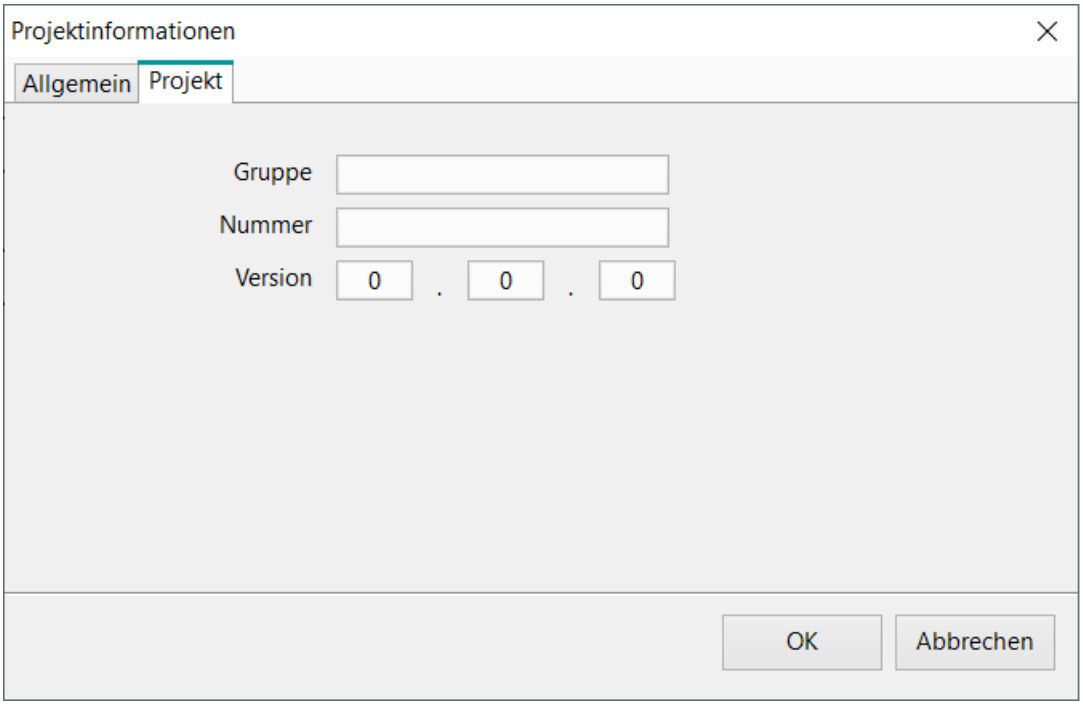
Allgemein

Der Reiter ***Allgemein*** enthält die Informationen zur Softwareversion.
Softwareversion bei Projekterstellung – Version der Software, in der das Projekt erstellt wurde.
Softwareversion bei Projektänderung – Version der Software, in der das Projekt geändert wurde.



Projekt

Die Registerkarte ***Projekt*** ist nicht für jedes Gerät verfügbar. In der Registerkarte können Sie Informationen über die Gruppe, Nummer und Version des Programms angeben, die im Fenster ***Geräteinformationen*** des angeschlossenen Geräts angezeigt werden sollen, nachdem das Projekt darauf gespeichert wurde.



– ***Gruppe***: Gibt den Namen der Projektgruppe an.
– ***Nummer***: Gibt die Projektnummer innerhalb der Gruppe an.
– ***Version***: Gibt die Projektversion an.

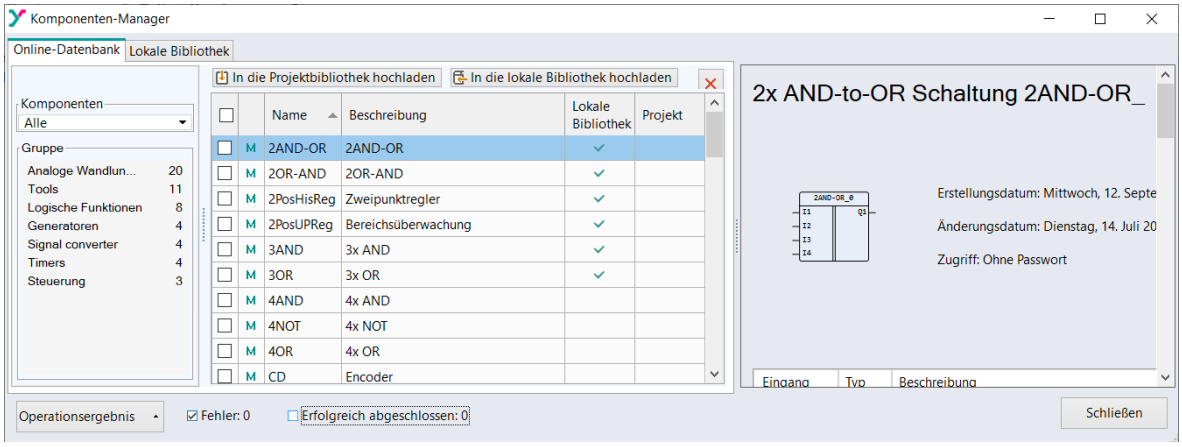
Klicken Sie auf ***OK***, um die Informationen im Projekt zu speichern, oder klicken Sie auf ***Abbrechen***, um sie zu verwerfen.

4.10 Komponenten-Manager

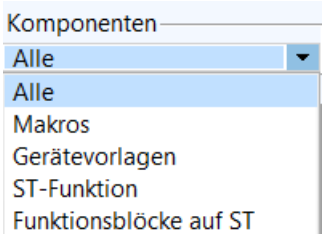
Neue Makros und Gerätevorlagen können aus der akYtec Online-Datenbank heruntergeladen werden. Der Komponenten-Manager ist das Werkzeug für alle Interaktionen mit dieser Datenbank. Für diese Interaktion ist ein Internetzugang erforderlich.
Wählen Sie den Menüpunkt ***Datei*** → ***Komponenten-Manager***, um ihn in einem separaten Fenster zu öffnen.

Online-Datenbank

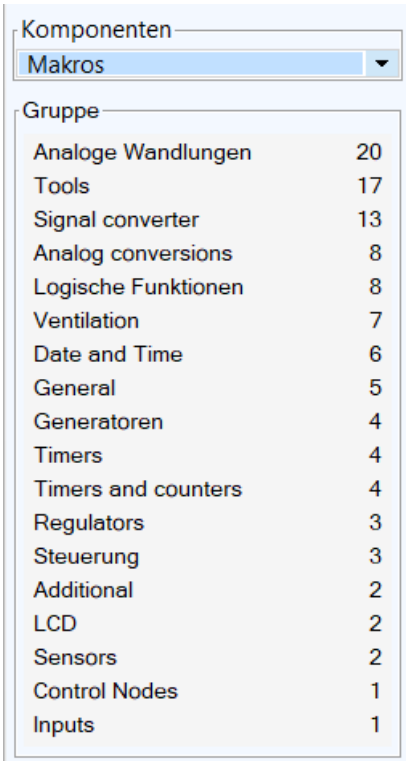
Auf der Registerkarte **Komponenten-Manager** wird eine Liste der Komponenten aus der akYtec-Online-Datenbank angezeigt.



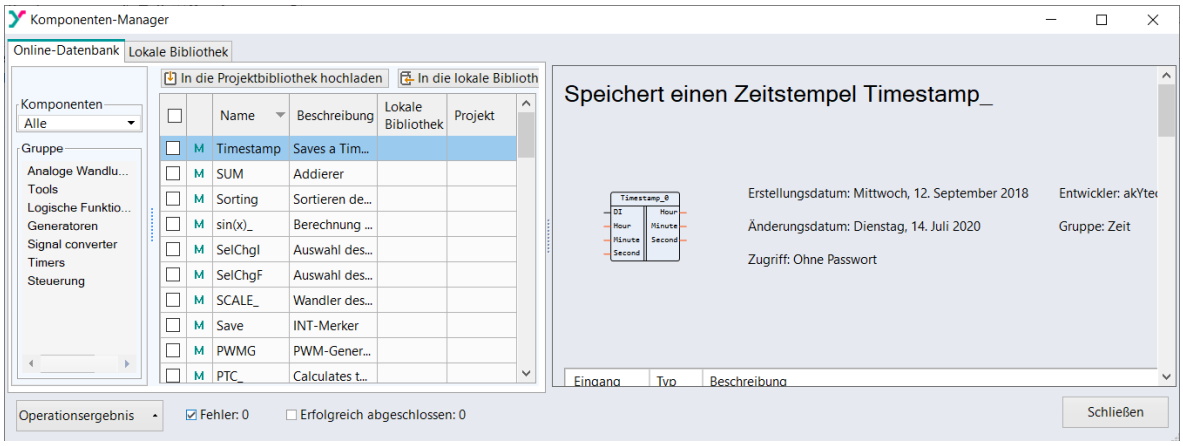
Über das Dropdown-Menü ***Komponenten*** können Sie die Liste nach Typ filtern.



Makros werden je nach Zweck weiter in Kategorien unterteilt.



Eine vollständige Beschreibung einer Komponente kann durch Auswahl in der Liste angezeigt werden. Die Komponentenbeschreibung wird im rechten Fensterbereich angezeigt. Zum Anzeigen der vollständigen Komponentendokumentation ist Acrobat Reader erforderlich.



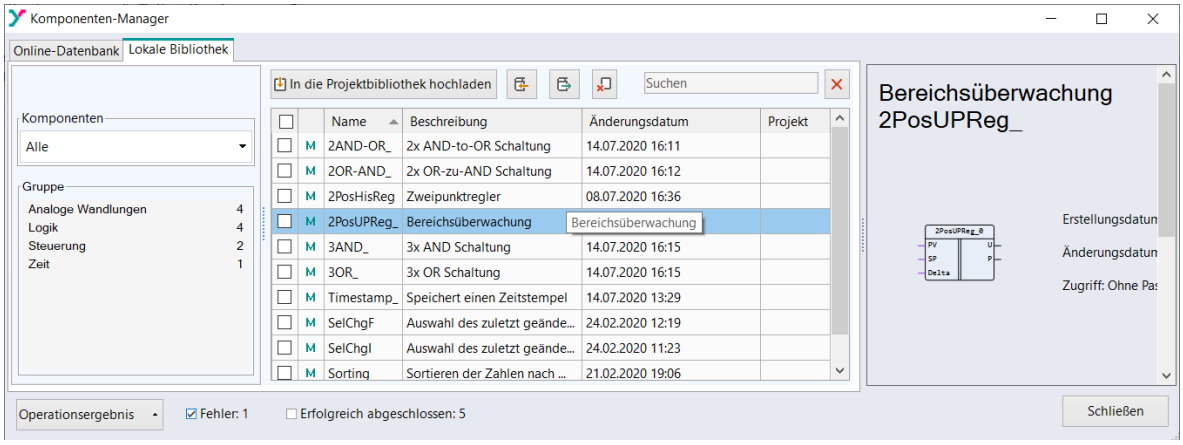
Um Komponenten auch ohne Internetverbindung verfügbar zu machen, aktivieren Sie die Kontrollkästchen daneben und klicken Sie auf die Schaltfläche „In lokale Bibliothek hochladen“. Die Komponenten werden dem lokalen Speicher hinzugefügt und erscheinen dann im Tab „Bibliothek“.
Um Komponenten sofort in das aktuelle Projekt zu laden, aktivieren Sie die Kontrollkästchen daneben und klicken Sie auf die Schaltfläche „In Projektbibliothek hochladen“. Die Komponenten werden dann im Bereich „Projektmakros“ dem Komponentenbibliotheksfenster hinzugefügt.
Der erfolgreiche Download von Komponenten in den lokalen Speicher oder in das aktuelle Projekt wird durch ein grünes Häkchen in den Spalten „Bibliothek“ bzw. „Projekt“ angezeigt.
Um eine Netzwerkvorlage mit der Gerätekonfiguration zu verbinden, befolgen Sie die Empfehlungen im Abschnitt *Netzwerkgerätevorlagen*.

Lokale Bibliothek

Die Registerkarte „Lokale Bibliothek“ zeigt den Inhalt des lokalen Komponentenspeichers auf dem PC des Benutzers an. Um Komponenten in das aktuelle Projekt zu laden, wählen Sie die gewünschten Komponenten aus und klicken Sie auf die Schaltfläche „In Projektbibliothek hochladen“. Die Komponenten werden dann im Bereich „Projektmakros“ dem

Komponentenbibliotheksfenster hinzugefügt. Das erfolgreiche Laden von Komponenten in das aktuelle Projekt wird durch ein grünes Häkchen in der Spalte „Projekt“ der Liste angezeigt.

HINWEIS
Das Laden von Komponenten, die Arrays enthalten, in die Projektbibliothek ist ab ALP Version 3.0 möglich.



HINWEIS
Bibliotheksdateien werden unter der lokalen Adresse gespeichert:
**C: \ Benutzer \ [Benutzername] \ Dokumente \akYtec ALP \ Library **

Der Komponentenmanager unterstützt die folgenden Komponentenverwaltungsfunktionen:

– – Eine Datei in den lokalen Speicher importieren.

– – Eine Datei aus dem lokalen Speicher exportieren.

– – Ausgewählte Komponenten löschen.

Komponentenimport

Mit dem Import können Sie Komponenten laden, die auf Ihrem PC gespeichert sind, aber noch nicht Teil des lokalen Speichers sind.
So importieren Sie eine Komponente:

1. Klicken Sie auf die Schaltfläche .
2. Wählen Sie im daraufhin angezeigten Fenster die zu importierenden Dateien aus.
3. Klicken Sie auf die Schaltfläche **Öffnen**. Die Dateien werden im Tab **Bibliothek** angezeigt.

Importbeschränkungen für frühere Versionen von ALP

Beim Importieren einer Komponente, die Funktionalitäten enthält, die von der Zielversion nicht unterstützt werden:

– Beim Import in ALP Version 2.11.370 oder höher wird die Komponente zwar geladen, jedoch mit Einschränkungen im Komponenten-Manager angezeigt. Die Beschreibung gibt die ALP-Version an, ab der die Komponente unterstützt wird.
– Beim Import in eine ALP Version niedriger als 2.11.370 – die Komponente wird zwar geladen, aber beim Versuch, sie aus der Bibliothek auszuwählen, erscheint ein Fehlerfenster.

Jegliche Operationen mit solchen Komponenten führen zu Fehlern. Es ist nicht möglich, in einem Projekt Komponenten zu verwenden, die von der aktuellen Version von ALP nicht unterstützt werden.

Komponentenexport

Mit der Exportfunktion können Sie Komponenten auf Ihrem PC speichern, um sie später zu bearbeiten oder zu übertragen.
So exportieren Sie eine Komponente:

1. Markieren Sie das Kästchen neben der gewünschten Komponente auf der Registerkarte **Bibliothek**.
2. Klicken Sie auf die Schaltfläche .
3. Wählen Sie im daraufhin angezeigten Fenster den Exportpfad aus.
4. Klicken Sie auf die Schaltfläche **OK**. Die Komponente wird als Datei am angegebenen Speicherort gespeichert.

Weitere Informationen

Die Funktionen für die Komponentendokumentation sind abhängig von der installierten Version von Acrobat Reader verfügbar, einschließlich des Speicherns der Datei im PDF-Format und des Druckens.
Um die Ergebnisse der Operationen im Komponenten-Manager anzuzeigen, klicken Sie auf die Schaltfläche Operation result – eine Meldungszeile wird am unteren Rand des Fensters angezeigt..

```
cos(x)_ - Component has been successfully added to the library
Ain-to-var - Component has been successfully added to the library
AVG - Component has been successfully added to the library
2AND-OR_ - The macro is added to the project
3AND_ - The macro is added to the project
3OR_ - The macro is added to the project
```

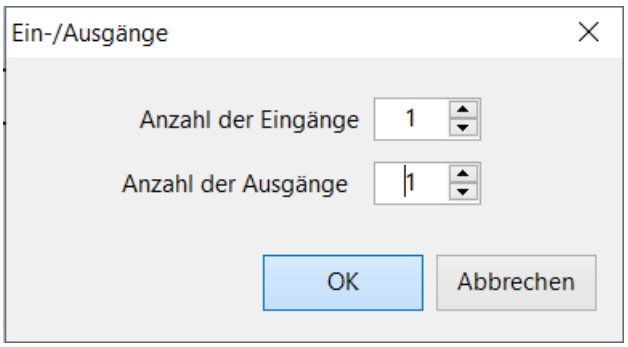
4.11 Makroentwicklung

Ein Makro ist ein benutzerdefinierter Algorithmus zur Verwendung in einem Projekt.
Hauptoperationen mit Makros:
– **Speichern** in der *Bibliothek* zur späteren Verwendung im Projekt
– **Exportieren** in eine Datei zur Verwendung in anderen Projekten
– **Importieren** zuvor erstellter Makros
– **Herunterladen** von Makros aus der Online-Datenbank in das Projekt

Neues Makro über das Hauptmenü

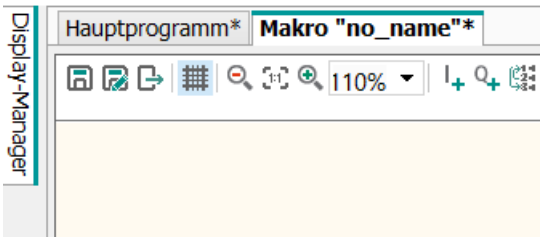
So erstellen Sie ein neues Makro:

1. Wählen Sie im Hauptmenü den Eintrag ***Datei*** → ***Neues Makro***. Anschließend legen Sie im geöffneten Dialogfenster die Anzahl der Ein- und Ausgänge fest.

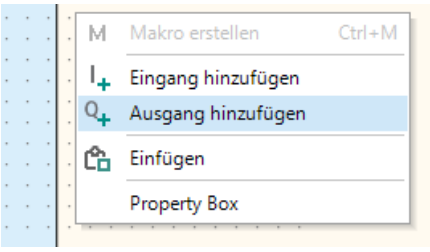


HINWEIS
Die Anzahl der Ein- und Ausgänge kann nach dem Erstellen eines Makros geändert werden.

2. Entwickeln Sie den Makroalgorithmus auf der Registerkarte **Makro-Editor**, ähnlich wie bei der Entwicklung des Programms im Diagramm.

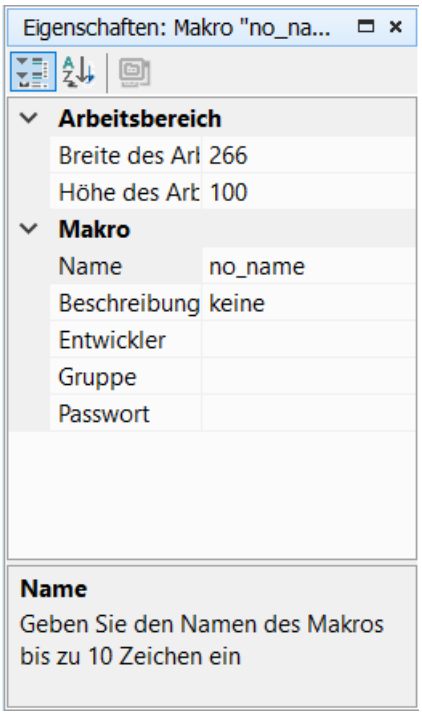


Die Anzahl der Ein- und Ausgänge kann jederzeit geändert werden. Um einen neuen Ein- oder Ausgang hinzuzufügen, verwenden Sie die Elemente oder in der Symbolleiste oder im Kontextmenü des Arbeitsbereichs.



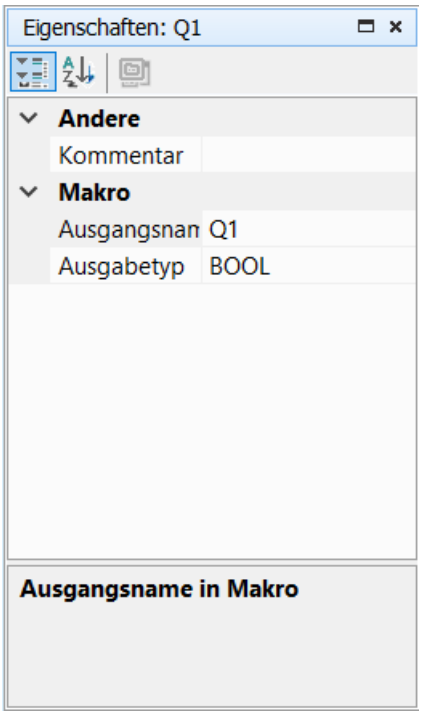
Um einen Eingang oder Ausgang zu entfernen, verwenden Sie ***Entfernen*** im Kontextmenü.

3. Geben Sie im Eigenschaftenfeld einen Namen, eine Beschreibung und eine Gruppe für das Makro ein.

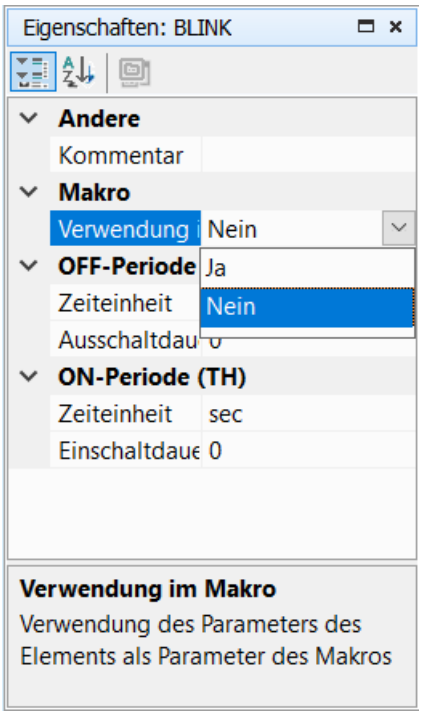


Der Name wird in der Kopfzeile der Arbeitsbereichsregisterkarte und in der Bibliothek angezeigt.

4. Im Eigenschaftenfeld können für jeden Ein- und Ausgang der Name und der Datentyp geändert werden.

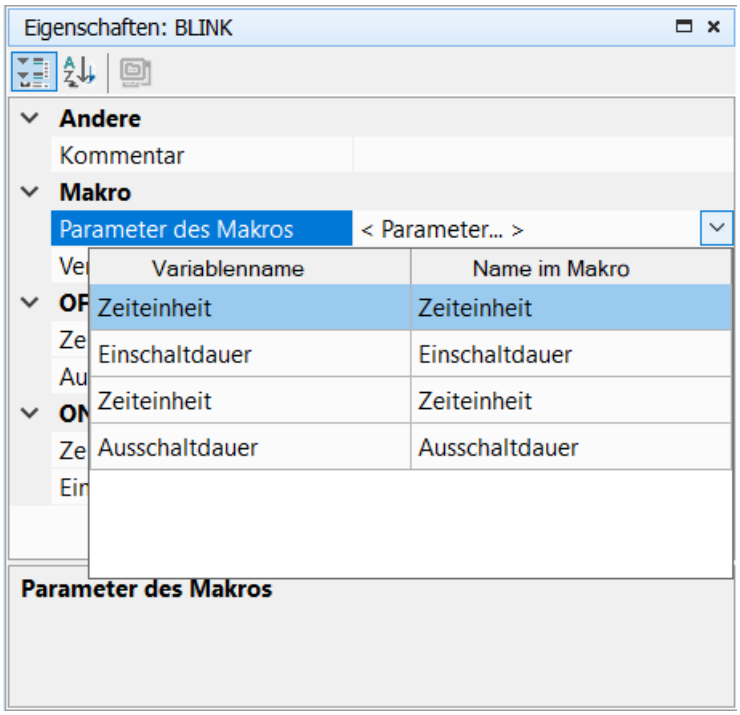


5. Als nächstes können Sie die Sichtbarkeit der im Makro verwendeten FB-Parameter im Hauptprogramm einstellen.

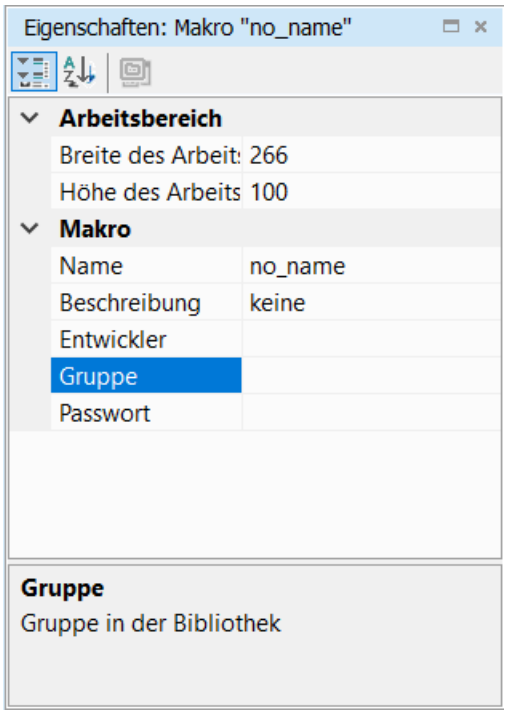


Wenn der Parameter ***In Makro verwenden*** auf ***Ja*** gesetzt ist, werden die FB-Parameter zu Parametern des Makros und dem Makro wird im Eigenschaftenfeld eine neue Option ***Parameter des Makros*** hinzugefügt.

Nach Auswahl von ***Ja*** erscheint eine Dropdown-Liste mit den verfügbaren Variablen zum Umbenennen. Zum Beispiel für die BLINK FB-Variable.



6. Das Makro kann auf die gleiche Weise wie das Hauptprogramm *simuliert 4.5* werden.
7. Vor dem Speichern des Makros können Sie folgende Felder ausfüllen: ***Name***, ***Beschreibung***, ***Entwickler***, ***Gruppe*** und ***Passwort***.



Es wird empfohlen, einen kurzen und eindeutigen Namen zu wählen. Der Text im Parameter ***Beschreibung*** wird in der Bibliotheksbox unter dem Makronamen und in einem Tooltip angezeigt, wenn sich der Mauszeiger im Hauptarbeitsbereich über dem Makro befindet.

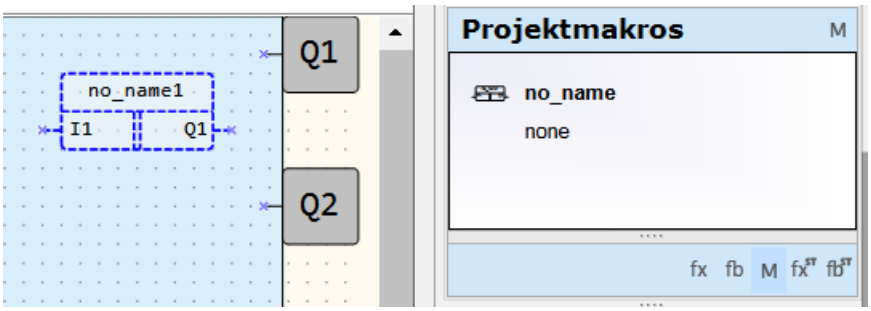
Wenn Sie das Passwort für das Makro festlegen, wird es bei jedem Aufruf des Menüpunkts Makro bearbeiten abgefragt. Andernfalls steht die Bearbeitung des Makros jedem zur Verfügung.

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

Das Makro kann durch Auswahl von ***Datei*→ *Makro speichern unter…*** oder durch Klicken auf das Symbol in der Symbolleiste des Makro-Editors gespeichert werden.

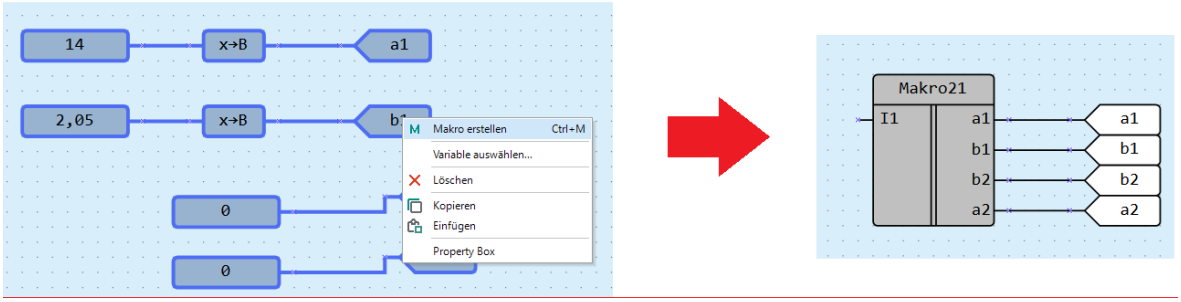
Gespeicherte Makros sind nur für ein geöffnetes Projekt verfügbar. Um das Makro in einem anderen Projekt zu verwenden, muss es in eine Datei exportiert und dann in ein anderes Projekt importiert werden.

8. Wählen Sie in der ***Bibliothek*** den Abschnitt ***Projektmakros*** aus und ziehen Sie ihn in den Arbeitsbereich.

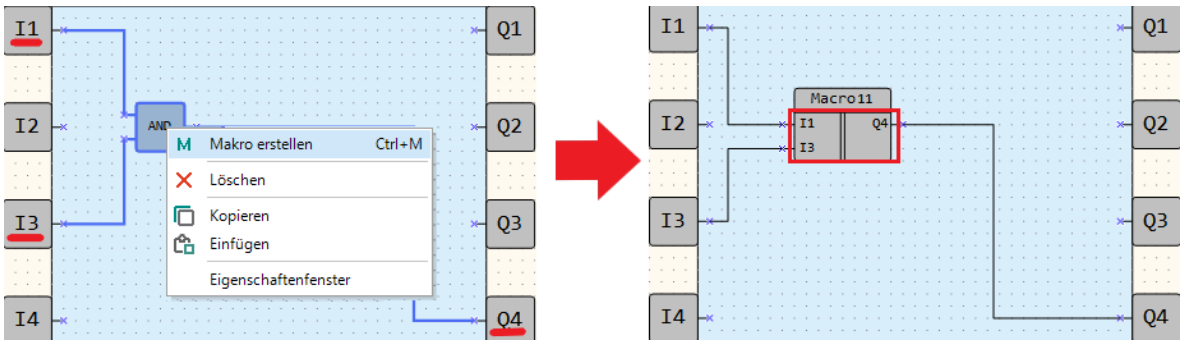


Neues Makro über Kontextmenü

Sie können ein Makro erstellen, indem Sie im Arbeitsbereich ein Auswahlrechteck aufziehen und im Kontextmenü des Arbeitsbereichs den Eintrag ***Neues Makro*** verwenden. Alle ausgewählten Bausteine werden in den neuen Makroblock verschoben, der die ausgewählten Bausteine im Hauptarbeitsbereich ersetzt. Alle externen Verbindungslinien bleiben erhalten.



Die Ein- und Ausgänge des Makros werden entsprechend den Elementen und Blöcken benannt, die zum Zeitpunkt ihrer Erstellung mit ihnen verbunden sind. Wenn ein Ausgang mit mehreren Elementen verbunden ist, wird er standardmäßig mit (Qn, wobei n die fortlaufende Nummer des Makroausgangs ist) benannt.



Das erstellte Makro und seine Eingabe-/Ausgabenamen können im Makro-Editor bearbeitet werden, indem man im Kontextmenü des Blocks die Option „Bearbeiten“ auswählt.
Die Anzahl der Ein- und Ausgänge des Makros entspricht der Anzahl der verbundenen Ein- und Ausgänge im ausgewählten Diagrammbereich. Werden Elemente ohne Verbindungen ausgewählt, wird ein Makro mit einem Eingang und einem Ausgang erstellt.
Elemente, die beim Erstellen des Makros im Diagramm nicht berücksichtigt werden:

– Ein- und Ausgänge

– Servicevariablen
– Netzwerkvariablen
– Der PID-Reglerblock (PID)

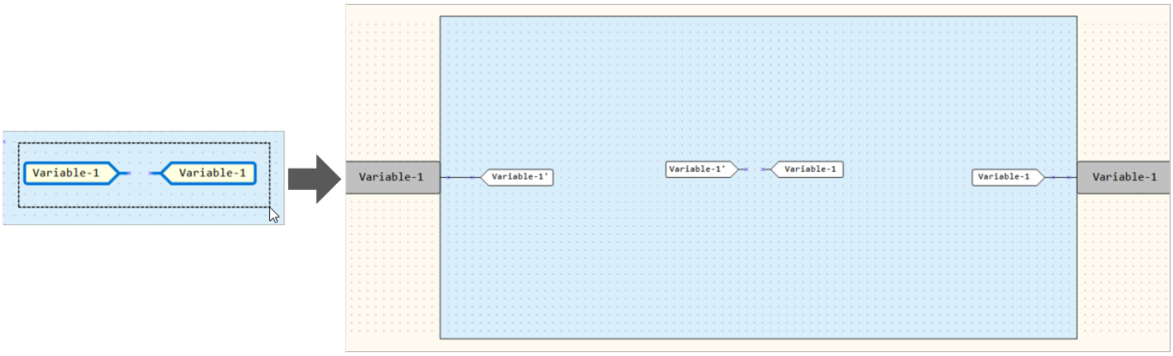
Wenn während der Makroerstellung eines der oben genannten Elemente ausgewählt wurde, wird es mit dem entsprechenden Ein-/Ausgang des Makroblocks verbunden.Wenn der ausgewählte Bereich einen Funktionsbaustein (FB) enthält, aus dem ein Wert gelesen oder in den ein Wert geschrieben wird, wird der Lese-/Schreibblock für diesen FB in das Makro aufgenommen, auch wenn er sich außerhalb des ausgewählten Bereichs befindet.
Wenn Standardvariablen innerhalb des Auswahlbereichs liegen, werden sie mit demselben Namen in das Makro kopiert.Variablen innerhalb des Makros sind nicht identisch mit den Variablen im Hauptdiagramm. Wenn alle Blöcke, die mit einer einzelnen Variable verknüpft sind, in das Makro aufgenommen werden, verschwinden sie aus dem Hauptdiagramm.
Wenn während der Makroerstellung sowohl der Eingabe- als auch der Ausgabebereich einer ***in-use*** Variable innerhalb des ausgewählten Bereichs liegen, wird eine neue Variable innerhalb des Makros erstellt. Diese neue Variable erbt alle Eigenschaften der ursprünglich ausgewählten Variable: Typ, Nichtflüchtigkeit und Standardwert.Ihr Name setzt sich aus dem Namen der ursprünglichen Variable und einem angehängten Apostroph (') zusammen.
Die gleichen Regeln gelten, wenn der bei der Makroerstellung ausgewählte Bereich Folgendes umfasst:

– Sowohl die Eingabe- als auch die Ausgabeblöcke eines verwendeten Array-Elements.
– Sowohl die Eingabe- als auch die Ausgabeblöcke eines Arrays, das mindestens ein verwendetes Element enthält.
– Ein Ein-/Ausgangsbaustein eines Arrays, das mindestens ein verwendetes Element enthält, zusammen mit dem Ein-/Ausgangsbaustein eines Elements aus diesem Array.

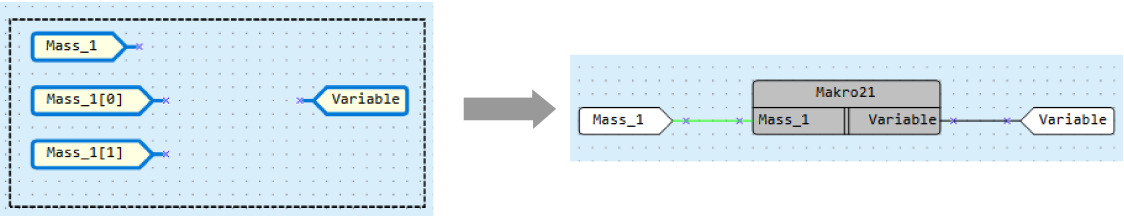
Der Name eines nach diesen Regeln erstellten Array-Elements wird wie folgt formatiert: Eckige Klammern ([ ]) werden durch geschweifte Klammern ({ }) ersetzt, und am Ende des Namens wird ein Apostroph (') hinzugefügt.

Beispiel:
Mass_1[0] → Mass_1{0}'

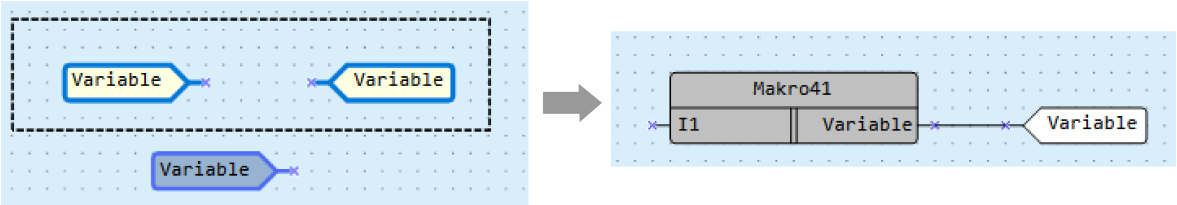
Wenn der ausgewählte Bereich während der Makroerstellung den Eingabeblock eines Arrays und Eingabeblöcke für Elemente desselben Arrays enthält, wird innerhalb des Makros nur eine Eingabe vom Typ Array erstellt, die nach dem Array benannt ist.



Wenn während der Makroerstellung der ausgewählte Bereich sowohl die Eingabe- als auch die Ausgabeblöcke einer einzelnen ungenutzten Variable enthält, die auch außerhalb des ausgewählten Bereichs im Diagramm existiert, werden diese Blöcke mit demselben Namen in das Makro kopiert, und nur der Ausgabevariablenblock verbleibt im ursprünglichen Diagramm.

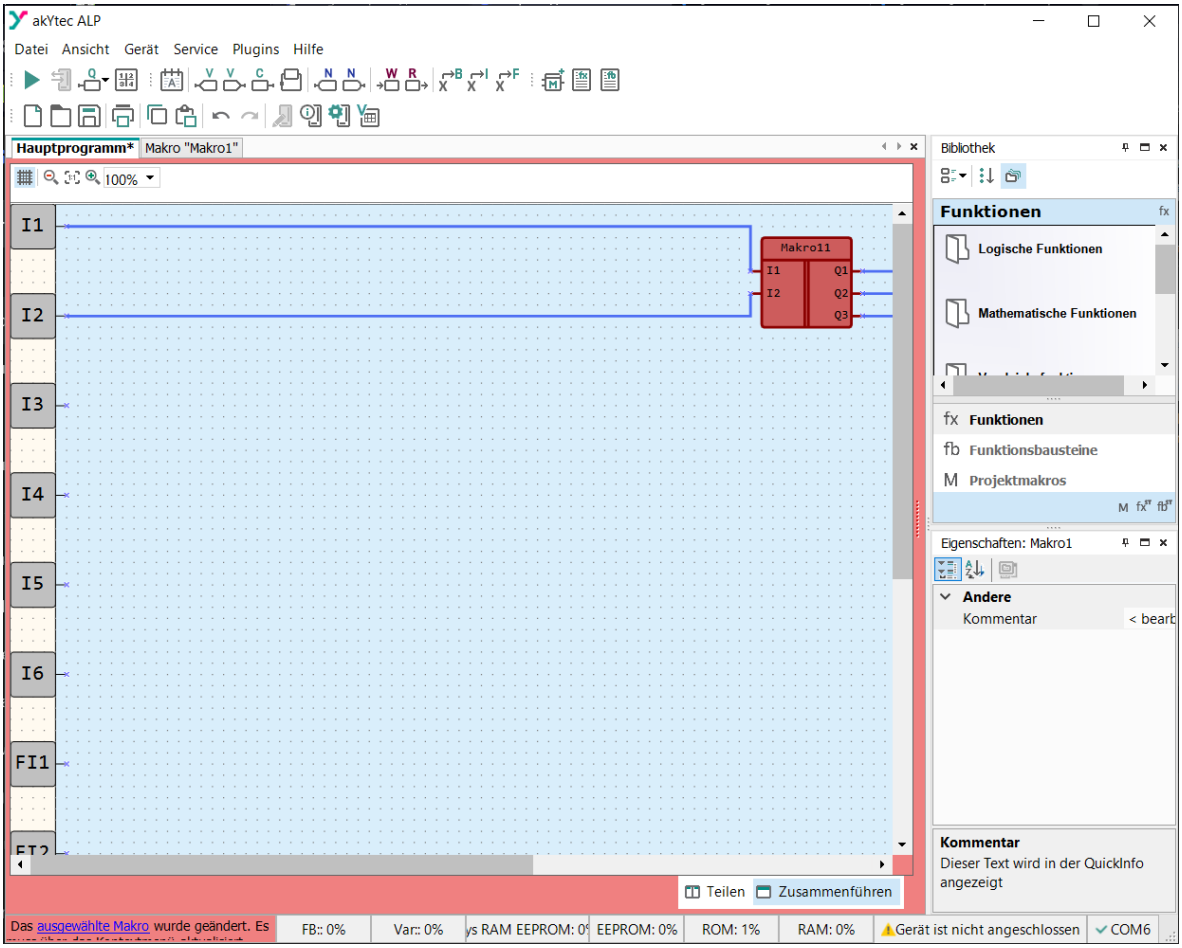


Wenn während der Makroerstellung der ausgewählte Bereich sowohl die Eingabe- als auch die Ausgabeblöcke einer einzelnen, unbenutzten Variable enthält, die auch außerhalb des ausgewählten Bereichs im Diagramm existiert, werden diese Blöcke mit demselben Namen in das Makro kopiert, und nur der Ausgabevariablenblock verbleibt im ursprünglichen Diagramm.



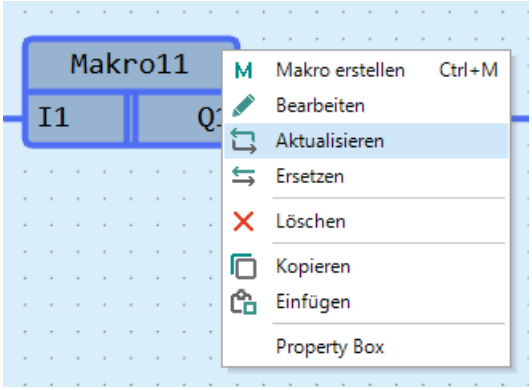
Makro aktualisieren

Wenn das im Hauptprogramm verwendete Makro geändert wurde (Name, Typ, Anzahl der E/A-Punkte, Elemente oder der Parameter ***In Makro verwenden*** eines beliebigen FB), wird es im Hauptprogramm rot hervorgehoben und der Benutzer wird aufgefordert, das Makro zu aktualisieren. Das Makro gilt als geändert, wenn die im Makro-Editor vorgenommenen Änderungen gespeichert werden.



So aktualisieren Sie ein Makro auf der Arbeitsfläche.

1. Klicken Sie mit der rechten Maustaste auf das markierte Makro und wählen Sie im daraufhin angezeigten Menü „Aktualisieren“.



HINWEIS
Im Falle eines verschachtelten Makros fordert ALP Sie auf, dorthin zu navigieren. Die Aktionen für ein verschachteltes Makro sind ähnlich.

2. Nach der Aktualisierung des Makros fährt ALP automatisch mit dem nächsten Makro fort, das eine Aktualisierung erfordert.

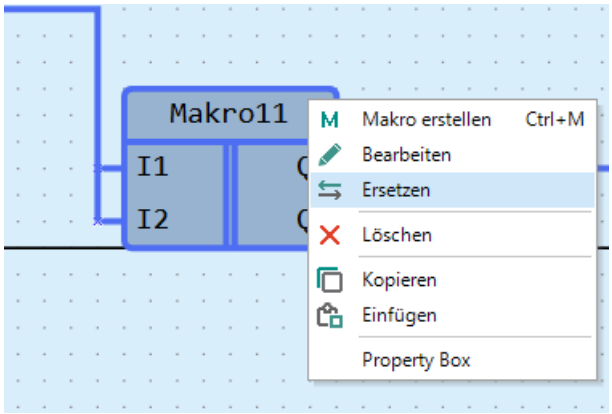
Ein Makro wird im Projekt nach folgenden Regeln aktualisiert:

– Wenn der Typ oder der Name einer Ein-/Ausgabe mit angeschlossener Verbindung im Makro geändert wurde, wird diese Verbindung beim Aktualisieren getrennt;
– Wenn dem Makro Ein-/Ausgänge hinzugefügt wurden, bleiben die Verbindungen bestehender Ein-/Ausgänge während der Aktualisierung erhalten.

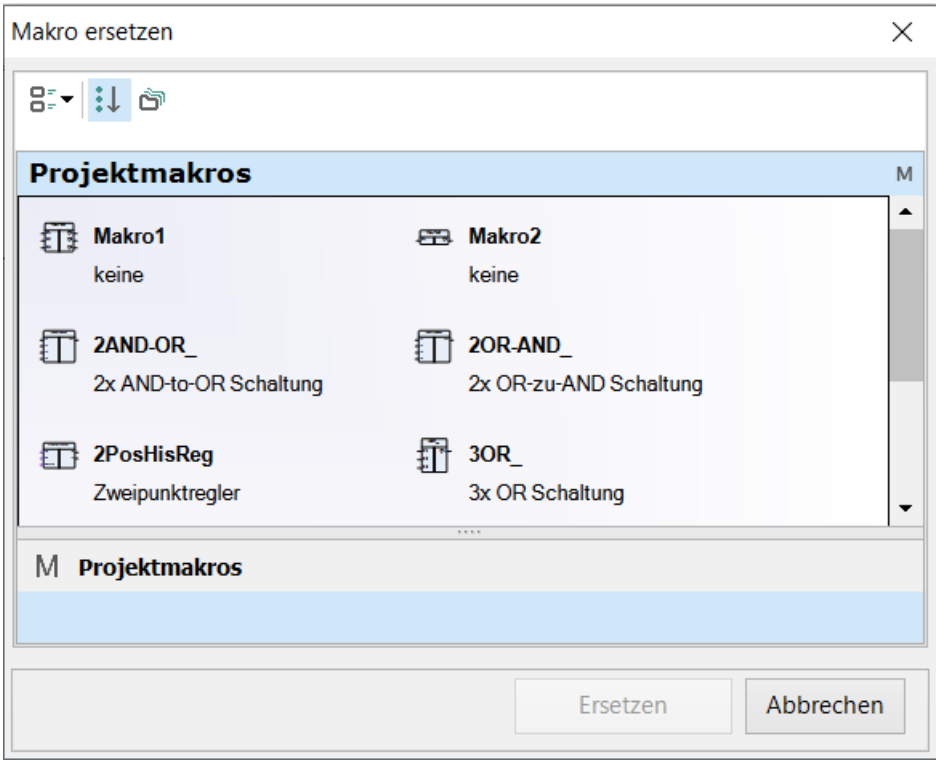
HINWEIS
ALP identifiziert Verbindungen zu Makro-Ein-/Ausgängen anhand von Name und Typ. Wenn Sie den Namen oder Typ eines Ein-/Ausgangs mit einer zugehörigen Verbindung im Makro ändern und einen neuen Ein-/Ausgang mit demselben Namen und Typ erstellen, stellt ALP die Verbindung automatisch wieder her. Wenn das Makro aktualisiert wird, wird die Verbindung zum neuen Ein-/Ausgang wiederhergestellt.

Makro ersetzen

Wenn Sie ein Makro durch ein anderes aus den Projektmakros ersetzen möchten, klicken Sie auf die Schaltfläche „Ersetzen“ im Kontextmenü des Makros.



Das Fenster „Makro ersetzen“ wird sich öffnen.



Wurde der Typ oder Name einer Ein-/Ausgabe mit angeschlossener Verbindung im Makro geändert, wird diese spezifische Verbindung beim Ersetzen getrennt, alle anderen Verbindungen bleiben jedoch erhalten.

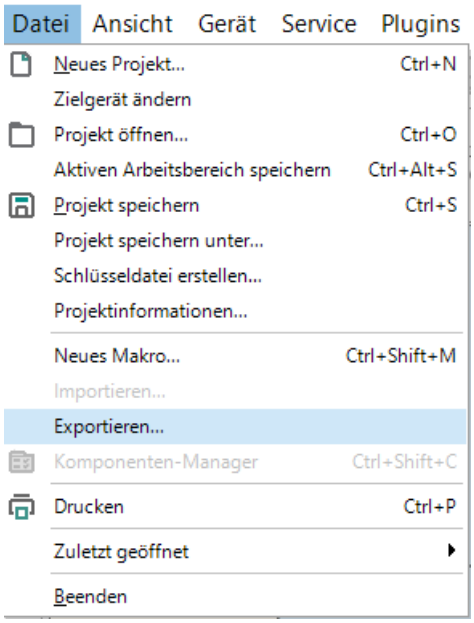
Makro exportieren

Das Exportieren eines Makros in eine Datei ist nur möglich, wenn das Makro-Editor-Fenster geöffnet ist. Um ein Makro zu exportieren, wählen Sie im Hauptmenü **Datei** → **Exportieren**.
So exportieren Sie ein Makro:

1. Öffnen Sie das Makro im Editor.

   Wenn Sie das Makro vor dem Speichern bearbeiten müssen, sollten Sie es auf die Projektfläche ziehen und im Makro-Kontextmenü „Bearbeiten und Änderungen vornehmen“ auswählen.

2. Wählen Sie den Hauptmenüpunkt **Datei** → **Exportieren**.

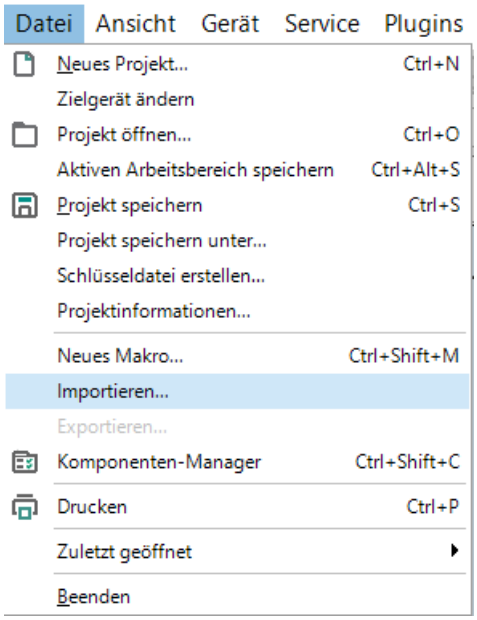


3. Wählen Sie im sich öffnenden Fenster einen Speicherort aus und speichern Sie die Makrodatei mit der Erweiterung *.tpl*. Nach dem Speichern wird eine Meldung angezeigt, dass das Makro erfolgreich exportiert wurde.

Makro importieren

Wenn Sie zum Erstellen eines Programms ein in einem anderen Projekt erstelltes Makro verwenden müssen, können Sie das erforderliche Makro in das Projekt importieren.
Um ein Makro zu importieren, verwenden Sie den Hauptmenüpunkt **Datei** → **Importieren**.

HINWEIS
Das Element **Importieren** ist nur aktiv, wenn der Fokus auf dem Arbeitsbereich liegt.



Wählen Sie im sich öffnenden Fenster die gewünschte Datei aus und klicken Sie auf die Schaltfläche **OK**. Das Makro wird in der ***Bibliothek*** im Bereich ***Projektmakros*** hinzugefügt und kann nun im Projekt verwendet werden.

HINWEIS
Das Importieren von Makros, die Variablen vom Typ Array oder Array-Elemente enthalten, ist in ALP Version 3.0 und höher möglich.

Makro kopieren

Makros können zur Wiederverwendung und Verkürzung der Entwicklungszeit von Projekt zu Projekt kopiert werden. Um ein Makro zu kopieren, wählen Sie den Makroblock im Quellprojekt aus und klicken Sie auf in der Symbolleiste oder wählen Sie den Befehl ***Kopieren*** im Kontextmenü des Blocks. Das Makro wird in ein anderes Projekt eingefügt, indem Sie auf die Schaltfläche in der Symbolleiste klicken oder im Kontextmenü des Arbeitsbereichs ***Einfügen*** wählen. Sie können zum Kopieren und Einfügen auch Tastaturkürzel verwenden, siehe *Tastaturkürzel*.
Nach dem Einfügen ist das Makro im Abschnitt ***Projektmakros*** der ***Bibliothek*** verfügbar.

HINWEIS
Das Einfügen von Makros, die Variablen vom Typ Array oder Array-Elemente enthalten, ist in ALP Version 3.0 und höher möglich.

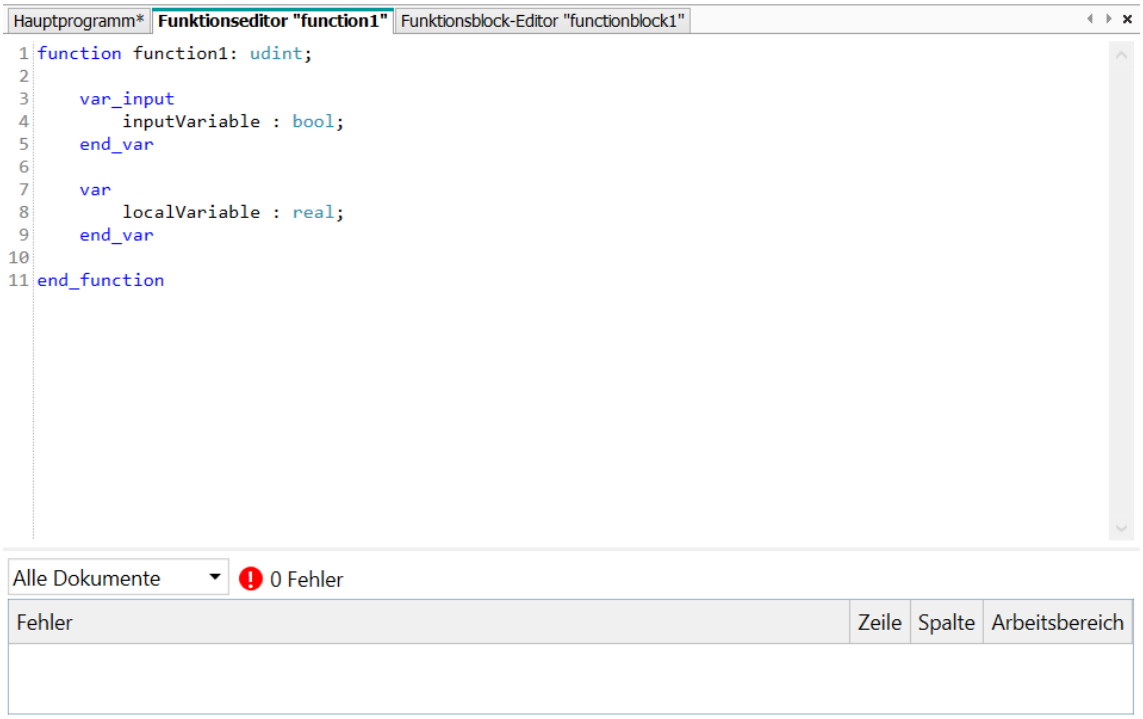
4.12 Verwendung der ST-Funktion

In ALP ist es möglich, benutzerdefinierte Funktionen in der ST-Sprache zu erstellen.

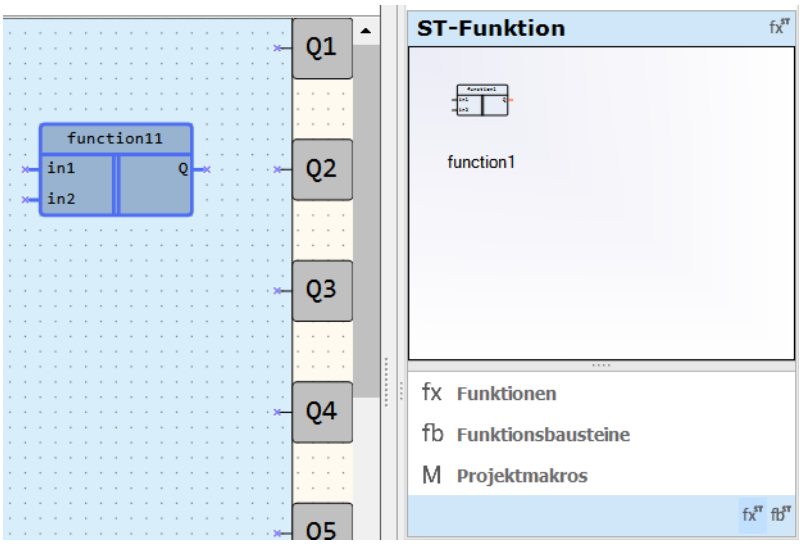
ACHTUNG
ST-Funktionen reservieren Speicherplatz im ROM-Speicher, nachdem sie der Projektbibliothek hinzugefügt wurden, unabhängig davon, ob sie im Projekt verwendet werden oder nicht.

Erstellung einer ST-Funktion

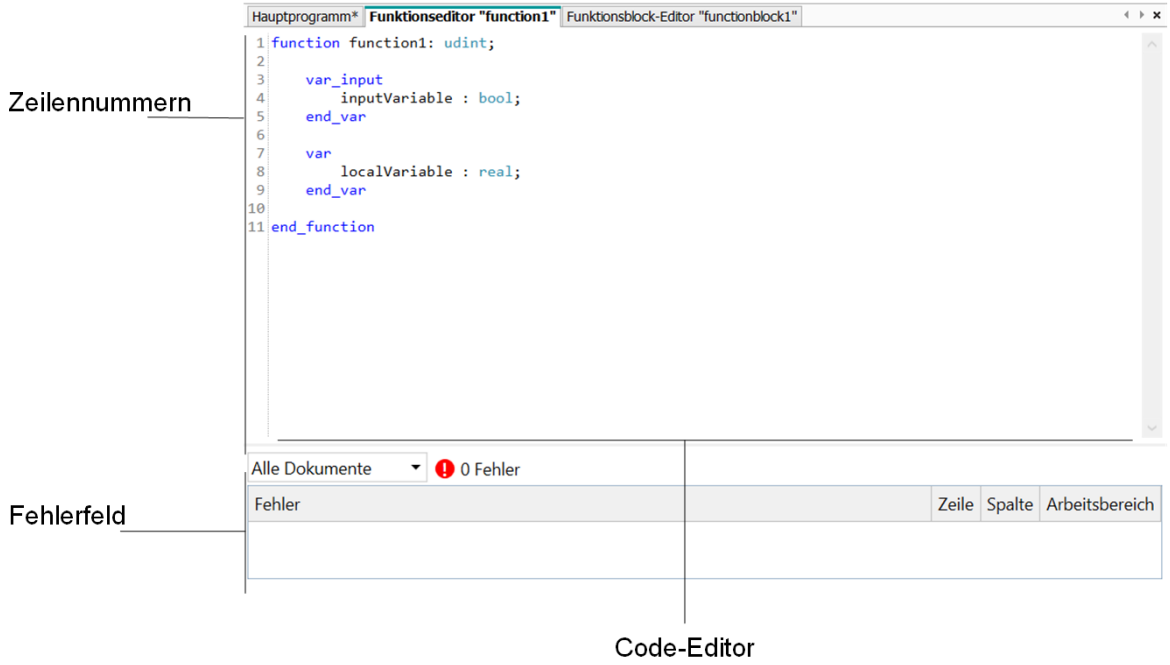
1. Klicken Sie in der Symbolleiste auf das Symbol ***Neue ST-Funktion***. Der Funktionseditor mit einer ST-Funktionsvorlage wird in einem neuen Arbeitsbereich geöffnet.



2. Geben Sie in der ersten Zeile den Funktionsnamen und den Ausgabedatentyp an.
3. Geben Sie alle erforderlichen Eingabevariablen im Eingabevariablen-Deklarationsblock ***var_input*** an.
4. Geben Sie alle erforderlichen lokalen Variablen im lokalen Variablendeklarationsblock ***var*** an.
5. Entwickeln Sie einen Funktionsalgorithmus gemäß den ST-Syntaxregeln.
6. Wechseln Sie auf die Registerkarte ***Hauptprogramm*** oder schließen Sie die Registerkarte ***Funktionseditor***. Die Funktion wird automatisch gespeichert.
7. Wählen Sie in der ***Bibliothek*** den Bereich ***ST-Funktionen*** aus und ziehen Sie die gespeicherte Funktion per Drag & Drop auf die Projektarbeitsfläche.



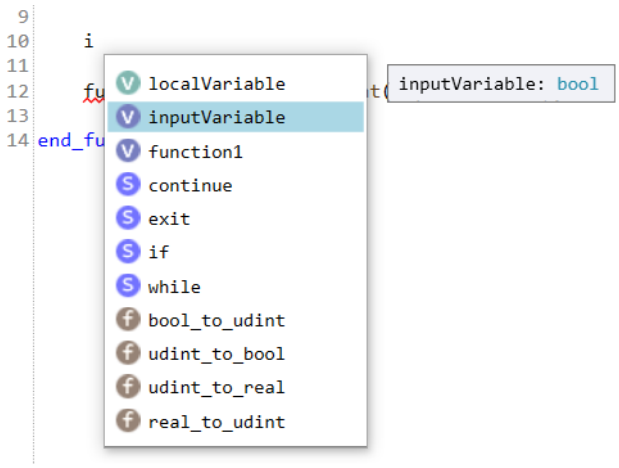
Funktionseditor-Oberfläche



1. **Zeilennummern** – Zeigt die fortlaufenden Zeilennummern im Programmcode an.
2. **Code-Editor** – Dient zur Bearbeitung des Codes mit automatischer Syntaxhervorhebung.
3. **Fehlerfeld** – Zeigt Fehler im Programmcode an.

Snippet-Verwaltung

Die Snippet-Verwaltung ist eine Texteditor-Funktion, die das einfache Einfügen von Inhalten aus einem Katalog wiederholt verwendeter Texte ermöglicht. Wenn Sie das erste Zeichen im Editor eingeben, öffnet sich ein Kontextmenü mit Fokus auf der ersten Zeile. Verwenden Sie die Cursortasten, um ein Snippet auszuwählen. Um das ausgewählte Snippet in den Code einzufügen, drücken Sie die ***Eingabetaste*** oder die ***Tabulatortaste*** oder doppelklicken Sie auf das Listenelement.



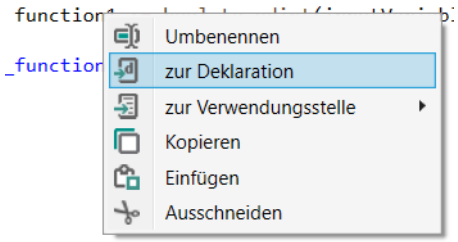
Snippet-Gruppen:
- V – Lokale Variablen
- S – Anweisungen (***while***, ***for*** usw.)
- K – Schlüsselwörter (***true***, ***false***)
- F – Integrierte Funktionen
- T – Weitere Funktionen

Innerhalb der Gruppen sind die Snippets alphabetisch sortiert.

Zur Deklaration oder zum Verwendungsort springen

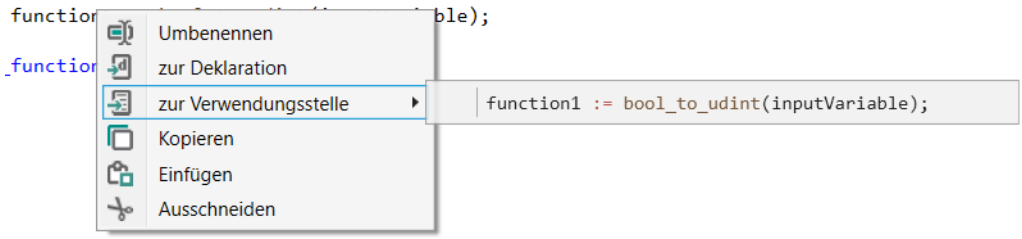
Zum komfortablen Arbeiten mit dem Code ist eine Suche implementiert, um die Stellen im Programmcode zu finden, an denen eine Funktion oder Variable deklariert oder verwendet wird.
So springen Sie zur Deklaration:

1. Platzieren Sie den Cursor auf den Namen einer Funktion oder Variable im Programmcode.
2. Klicken Sie mit der rechten Maustaste auf den Namen.
3. Wählen Sie im Kontextmenü ***Zur Deklaration*** aus.



So springen Sie zum Verwendungsort:

1. Platzieren Sie den Cursor auf den Namen einer Funktion oder Variable im Programmcode.
2. Klicken Sie mit der rechten Maustaste auf den Namen.
3. Wählen Sie im Kontextmenü ***Zur Verwendungsstelle***. Es öffnet sich eine Liste mit Stellen im Code, an denen die ausgewählte Funktion oder Variable verwendet wird.

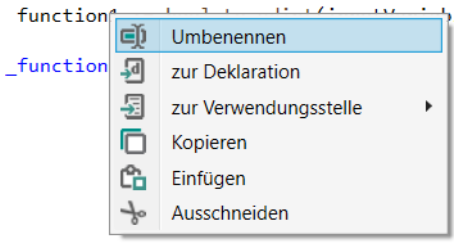


4. Klicken Sie mit der linken Maustaste auf die ausgewählte Verwendungsstelle. Der Cursor springt in die Zeile, in der die Funktion oder Variable verwendet wird.

Variable oder Funktion umbenennen

Eine zentrale Änderung des Namens einer Variablen oder Funktion im gesamten Code ist möglich.
Gehen Sie wie folgt vor:

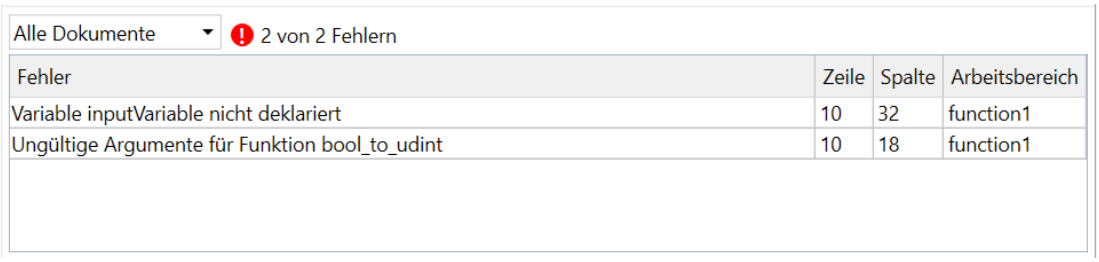
1. Platzieren Sie den Cursor auf den Namen einer Funktion oder Variable im Programmcode.
2. Klicken Sie mit der rechten Maustaste auf den Namen.
3. Wählen Sie im Kontextmenü ***Umbenennen***. Der Name wird an allen Stellen, an denen er verwendet wird, grün markiert.



4. Tragen Sie an einer der grün markierten Stellen einen neuen Namen ein und klicken Sie an eine andere Stelle im Code. Nun wird der Name im gesamten Programm geändert.

Fehlerfeld

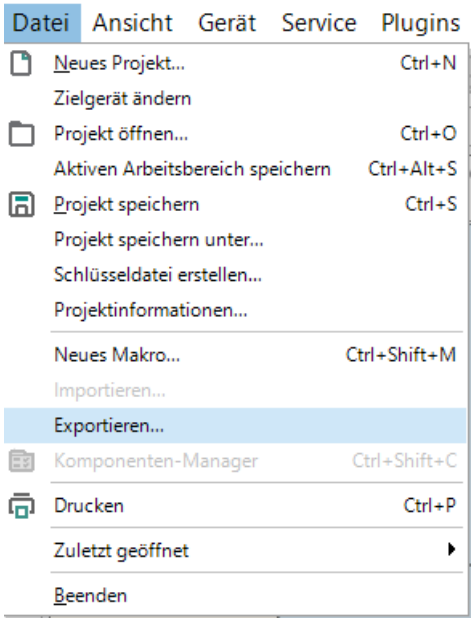
Alle beim Schreiben des Codes aufgetretenen Fehler werden im ***Fehlerfenster*** aufgelistet. Klicken Sie mit der linken Maustaste auf eine Zeile in der Liste, um zum Fehler im Code zu springen.



ST-Funktion exportieren

Das Exportieren einer Funktion in eine Datei ist nur möglich, wenn die Registerkarte Funktionseditor geöffnet ist. Um eine Funktion zu exportieren, wählen Sie im Hauptmenü **Datei** → **Exportieren**.
Um die Funktion zu exportieren, gehen Sie wie folgt vor:

1. Wählen Sie im Hauptmenü **Datei** → **Exportieren**.

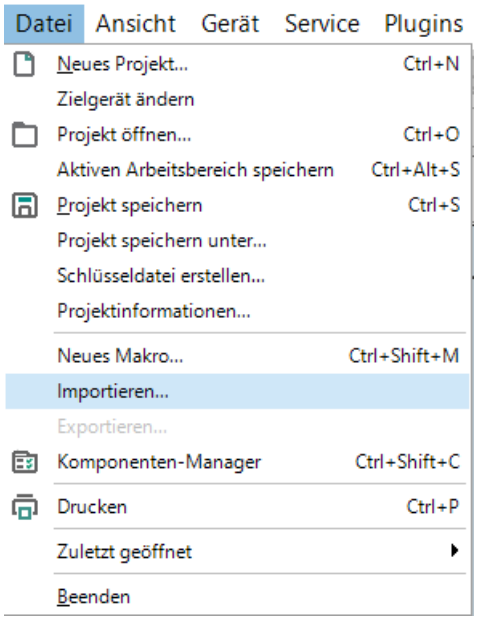


2. Wählen Sie im sich öffnenden Fenster einen Speicherort aus und speichern Sie die Funktionsdatei mit der Erweiterung **.fst*. Nach dem Speichern wird eine Meldung angezeigt, die angibt, dass die Funktion erfolgreich exportiert wurde.

ST-Funktion importieren

Wenn Sie zum Erstellen eines Programms eine in einem anderen Projekt erstellte Funktion verwenden müssen, können Sie sie in das Projekt importieren.
Um einen Funktionsblock zu importieren, wählen Sie im Hauptmenü **Datei** → **Importieren**.

HINWEIS
Das Element **Importieren** ist nur aktiv, wenn der Fokus auf dem Projektarbeitsbereich liegt.



Wählen Sie im sich öffnenden Fenster die gewünschte Datei aus und klicken Sie auf die Schaltfläche ***OK***. Die Funktion wird der ***Bibliotheksbox*** im Bereich ***ST-Funktionen*** hinzugefügt und kann nun im Projekt verwendet werden.

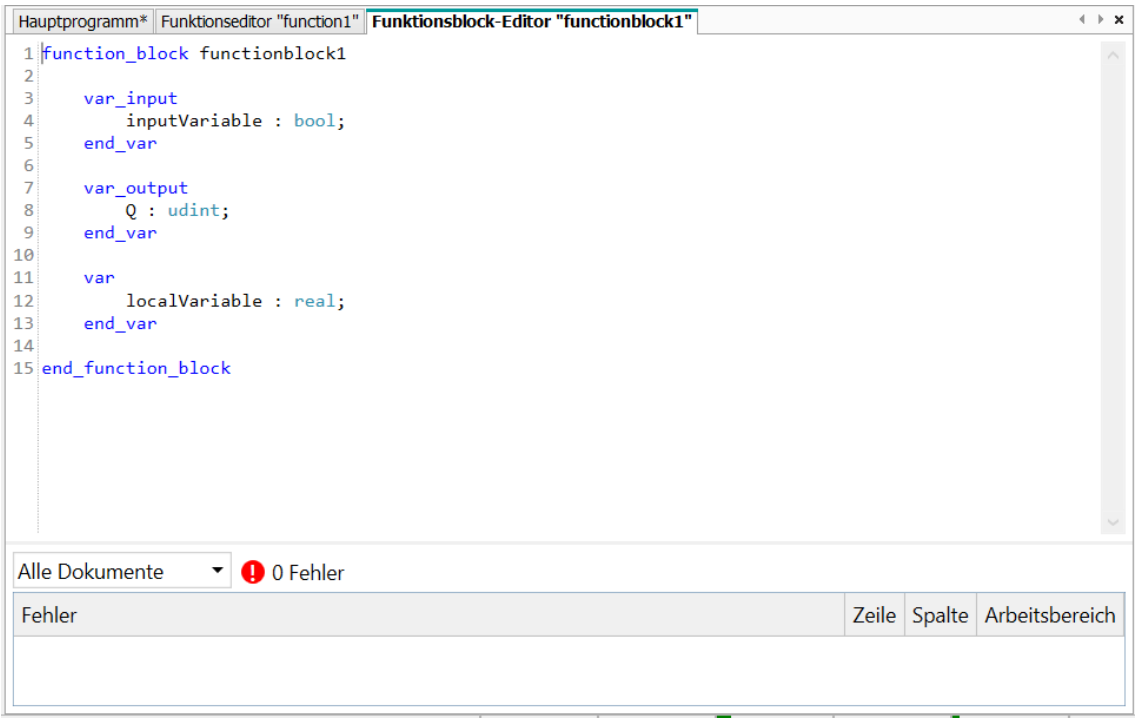
4.13 ST-Funktionsbausteine

In ALP ist es möglich, benutzerdefinierte Funktionsbausteine in der ST-Sprache zu erstellen.

Erstellung eines ST-Funktionsbausteins

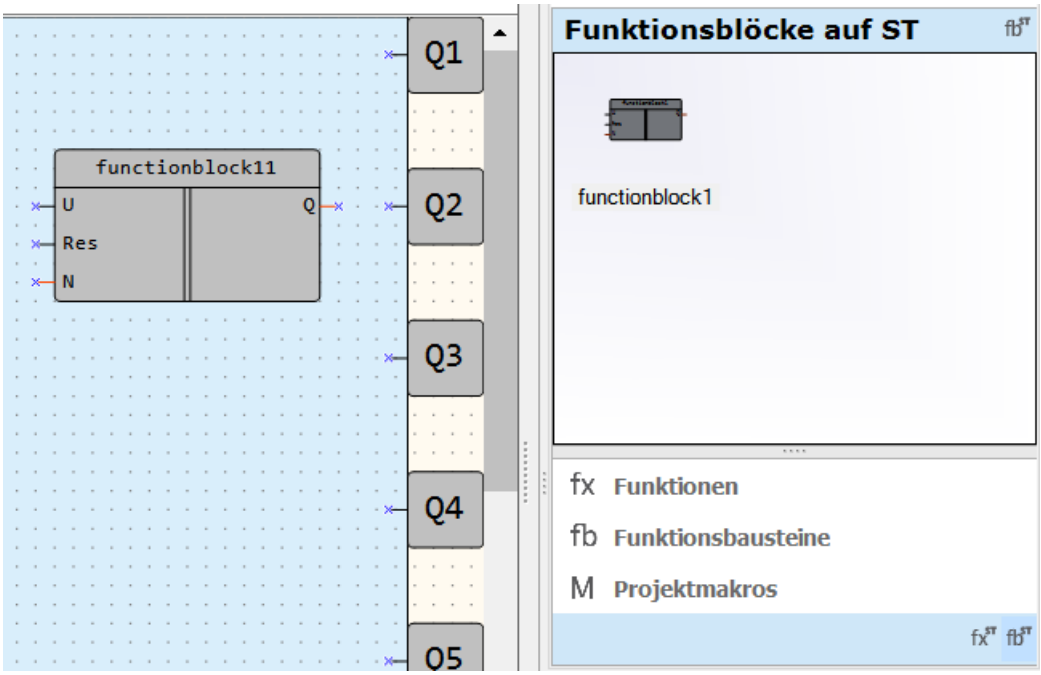
So erstellen Sie einen ST-Funktionsbaustein:

1. Klicken Sie auf das Symbol ***Neuer ST-Funktionsblock*** in der Symbolleiste. Der Funktionseditor mit einer ST-Funktionsbausteinvorlage wird in einem neuen Arbeitsbereich geöffnet.



2. Geben Sie in der ersten Zeile den Funktionsbausteinnamen und den Ausgabedatentyp an.

3. Geben Sie alle erforderlichen Eingabevariablen im Eingabevariablen-Deklarationsblock ***var_input*** an.
4. Geben Sie alle erforderlichen lokalen Variablen im lokalen Variablendeklarationsblock ***var*** an.
5. Entwickeln Sie einen Funktionsbausteinalgorithmus gemäß den ST-Syntaxregeln.
6. Wechseln Sie auf die Registerkarte ***Hauptprogramm*** oder schließen Sie die Registerkarte ***Funktionsbaustein-Editor***. Der Funktionsbaustein wird automatisch gespeichert.
7. Wählen Sie in der ***Bibliothek*** den Abschnitt ***ST-Funktionsbausteine*** aus und ziehen Sie den gespeicherten Baustein per Drag & Drop auf die Projektarbeitsfläche.



Oberfläche des Funktionsbaustein-Editors

1. **Zeilennummern** – Zeigt die fortlaufenden Zeilennummern im Programmcode an.
2. **Code-Editor** – Dient zur Bearbeitung des Codes mit automatischer Syntaxhervorhebung.

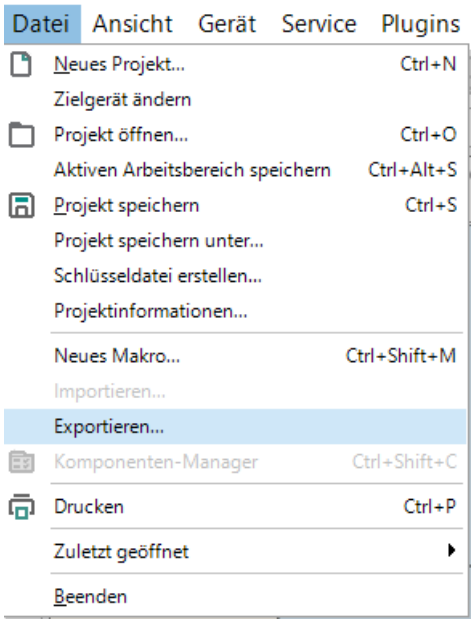
3. **Fehlerfeld** – Zeigt Fehler im Programmcode an.

HINWEIS
Ähnlich wie der Funktionseditor unterstützt der Funktionsblock-Editor *Snippets* , *Verfolgung des Deklarations- und Verwendungsorts* , *Umbenennung* und *Fehlerverfolgung* .

ST-Funktionsbaustein exportieren

Das Exportieren eines Funktionsbausteins in eine Datei ist nur möglich, wenn die Registerkarte Funktionsbaustein-Editor geöffnet ist. Um eine Funktion zu exportieren, wählen Sie im Hauptmenü **Datei** → **Exportieren**.
Um den Funktionsbaustein zu exportieren, gehen Sie wie folgt vor:

1. Öffnen Sie den Funktionsblock im Editor.
2. Wählen Sie im Hauptmenü **Datei** → **Exportieren**.

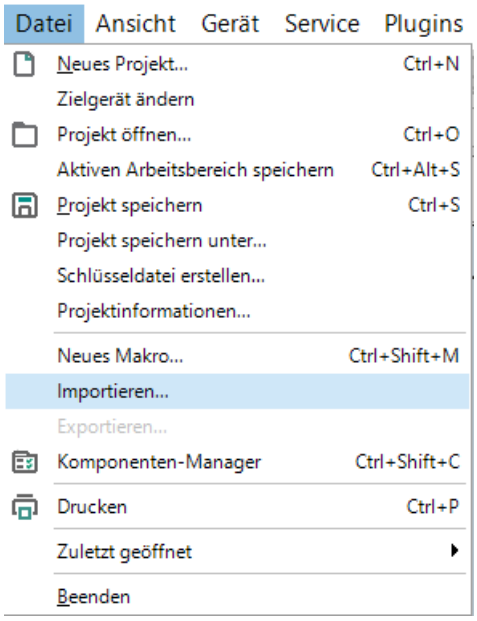


3. Wählen Sie im sich öffnenden Fenster einen Speicherort aus und speichern Sie die Funktionsdatei mit der Erweiterung **.fbst*. Nach dem Speichern wird eine Meldung angezeigt, die angibt, dass die Funktion erfolgreich exportiert wurde.

ST-Funktionsbaustein importieren

Wenn Sie zum Erstellen eines Programms einen in einem anderen Projekt erstellten Funktionsbaustein verwenden müssen, können Sie ihn in das Projekt importieren.
Um einen Funktionsbaustein zu importieren, wählen Sie im Hauptmenü **Datei** → **Importieren**.

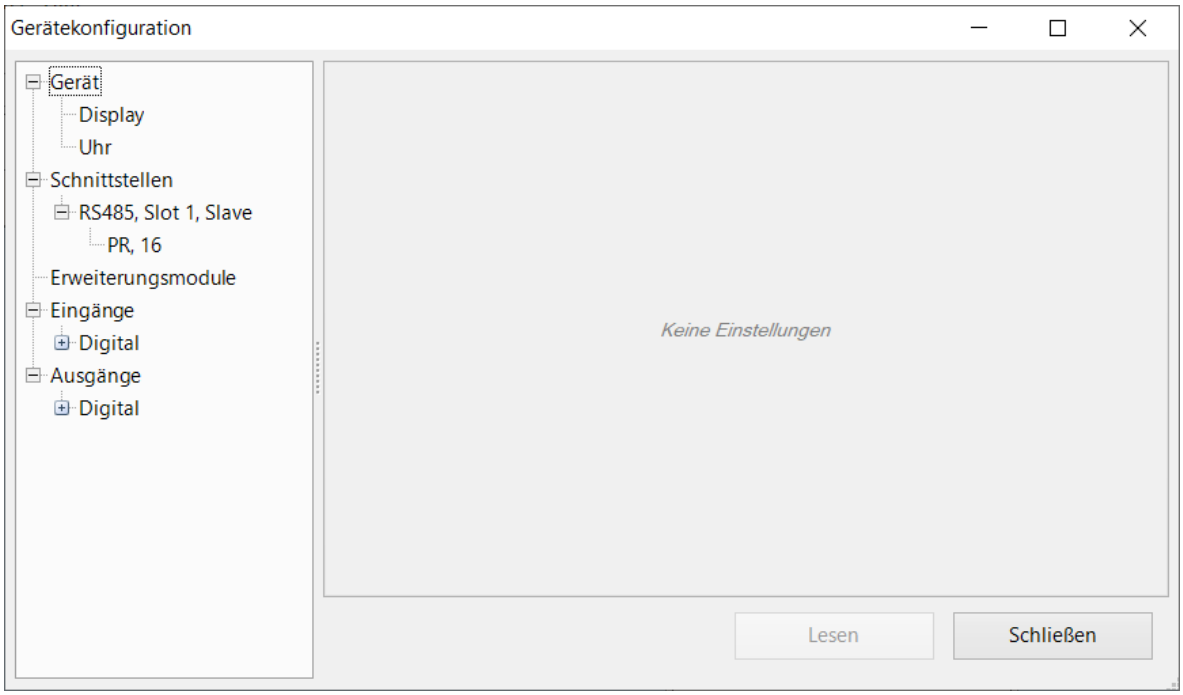
HINWEIS
Das Element **Importieren** ist nur aktiv, wenn der Fokus auf dem Projektarbeitsbereich liegt.



Wählen Sie im sich öffnenden Fenster die gewünschte Datei aus und klicken Sie auf die Schaltfläche ***OK***. Der Funktionsbaustein wird der ***Bibliothek*** im Bereich ***ST-Funktionsbausteine*** hinzugefügt und kann nun im Projekt verwendet werden.

5 Geraetekonfiguration

Die Konfiguration des Gerätes ist Bestandteil eines Projektes und kann über den Menüpunkt ***Gerät→Gerätekonfiguration*** eingestellt werden. Das Dialogfenster ***Gerätekonfiguration*** besteht aus zwei Teilen. Im linken Teil des Fensters werden die konfigurierbaren Parameter des Gerätes im Parameterbaum dargestellt. Im rechten Teil wird der Inhalt einer Gruppe dargestellt.



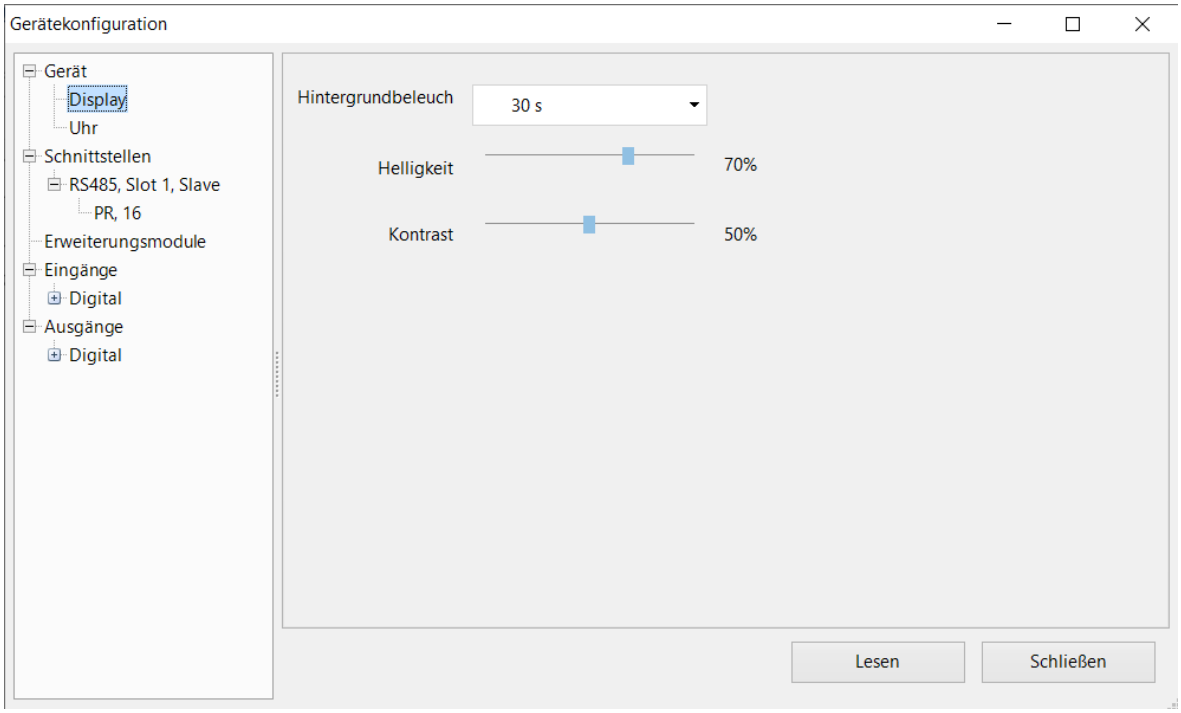
Der Inhalt des Parameterbaums hängt vom Zielgerät ab und kann folgende Gruppen enthalten:

– *Display*
– *Uhr*
– *Schnittstellen*
– *Erweiterungsmodule*
– *Eingänge und Ausgänge*

Alle Einstellungen, bis auf die Uhreinstellungen, werden im Projekt gespeichert. Die Konfiguration ist auch ohne Anschluss des Gerätes möglich.

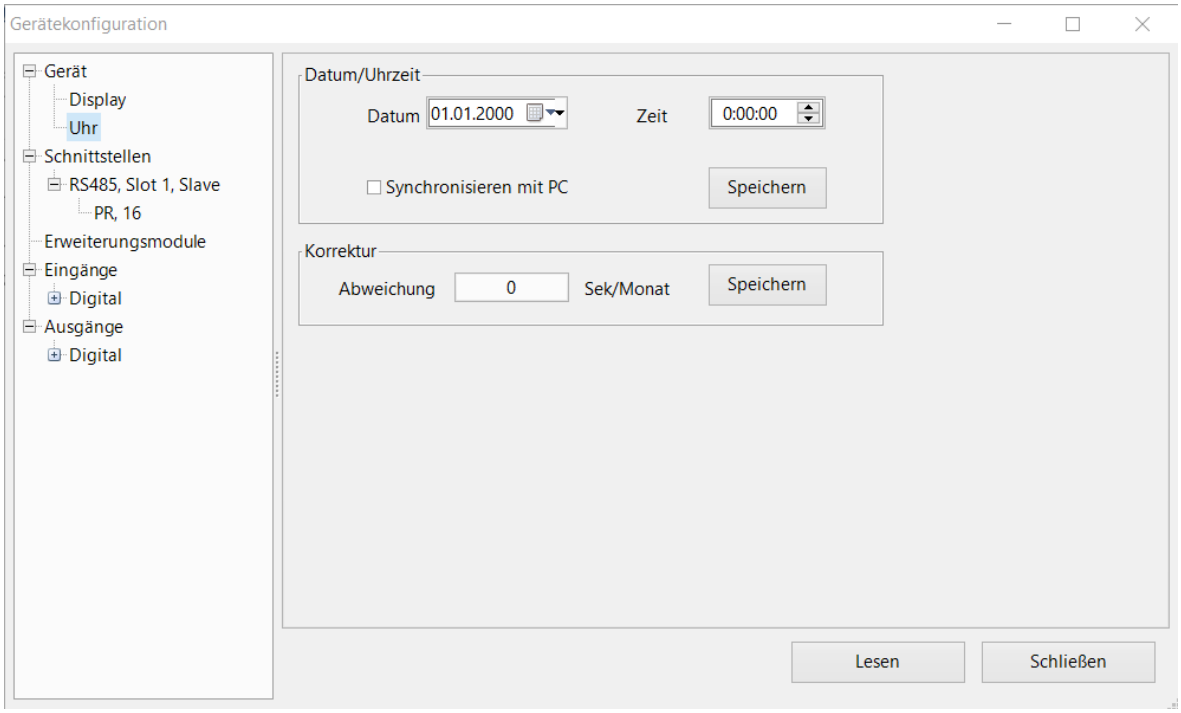
5.1 Display

Wenn das Zielgerät über ein Display verfügt, können folgende Parameter eingestellt werden:
Hintergrundbeleuchtung – Die Dauer der Hintergrundbeleuchtung seit der letzten Benutzeraktivität
Helligkeit – 0…100%
Kontrast – 0…100%
Über die Schaltfläche ***Lesen*** können die aktuellen Displayeinstellungen aus dem angeschlossenen Gerät ausgelesen werden.



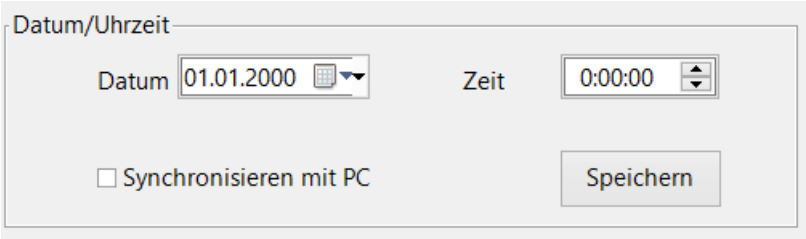
5.2 Uhr

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



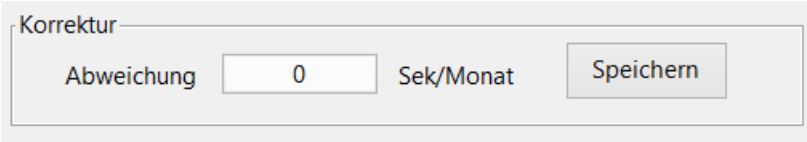
Datum und Uhrzeit

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



Korrektur

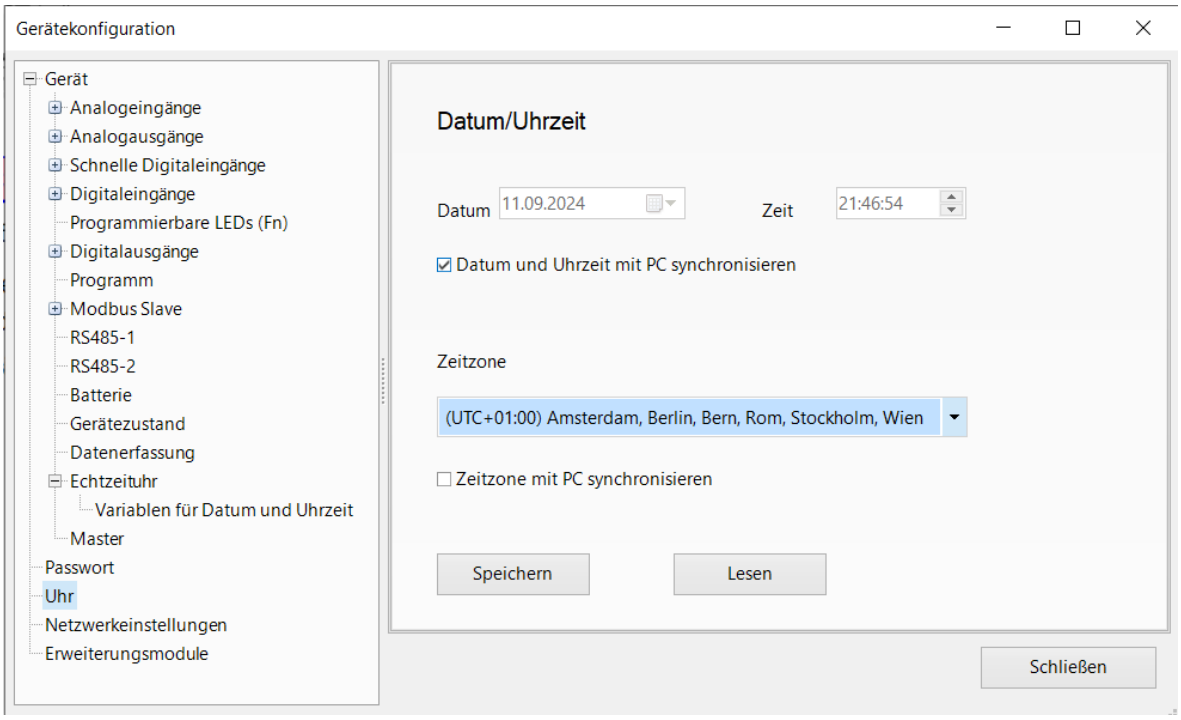
Geben Sie im Feld ***Abweichung*** den Uhrfehler in Sekunden pro Monat an, um die Uhrkorrektur einzustellen. Geben Sie einen negativen Wert ein, wenn die Geräteuhr zu schnell geht.



Um die Uhrkorrektur im Gerät zu speichern, klicken Sie im Bereich ***Korrektur*** auf die Schaltfläche ***Speichern***.
Über die Schaltfläche ***Lesen*** können die aktuellen Zeiteinstellungen vom verbundenen Gerät ausgelesen werden.

Einstellungen für Geräte auf der neuen Hardware-Plattform

Das Fenster mit den Uhreinstellungen für Geräte auf der neuen Hardwareplattform hat eine andere Benutzeroberfläche und verfügt nicht über eine Zeitkorrektur (Geräte mit Hardware, die eine höhere Genauigkeit bietet). Das Einstellen der Zeitzone ist erforderlich, um die Ortszeit korrekt anzuzeigen, da das Gerät den Zeitwert als Greenwich Mean Time (GMT) speichert. Durch Aktivieren der Option **Zeitzone mit PC synchronisieren** wird die Echtzeituhrzeit des Geräts mit der PC-Uhr synchronisiert.



5.3 Datenaustausch

– *Schnittstellen*
– *Modbus*

5.3.1 Schnittstellen

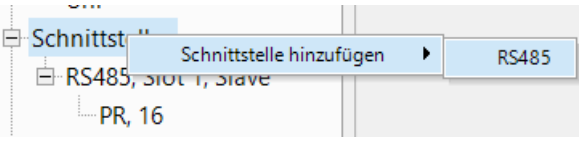
In ***Schnittstellen*** werden die Parameter von Schnittstellen für die Gerätekommunikation mit anderen Geräten im Netzwerk konfiguriert.
Folgende Schnittstellen sind verfügbar:

– *RS485*
– *Ethernet*
– *LoRa*

Kommunikationsschnittstellen RS485 und Ethernet werden mit dem *Modbus*-Protokoll verwendet.
Die Datenkommunikation über LoRa erfolgt über das LoRaWAN-Protokoll.
Das Hinzufügen, Ersetzen und Löschung einer Schnittstelle sind für Geräte der ersten Generation verfügbar. Bei Geräten der zweiten Generation befinden sich alle verfügbaren Schnittstellen im Konfigurationsbaum.

Schnittstelle hinzufügen

Wenn im Gerät die Funktion Schnittstelle hinzufügen verfügbar ist, rechtsklicken Sie im Konfigurationsbaum auf ***Schnittstellen*** und wählen Sie ***Schnittstelle hinzufügen***.

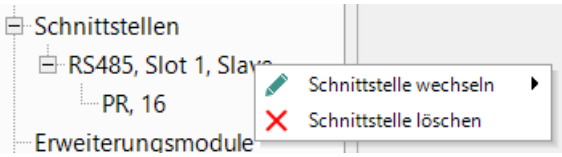


Es wird eine Schnittstelle des ausgewählten Typs mit Standardeinstellungen hinzugefügt. Der Schnittstellenname besteht aus der Nummer des im Gerät belegten Slots und dem Schnittstellenmodus (Master/Slave).
Die LoRa-Schnittstelle in PR102 und PR200 kann sich nur in Slot 2 befinden.

Die Anzahl der Schnittstellen für das Gerät darf die Anzahl der Slots im Gerät nicht überschreiten.

Schnittstelle wechseln/löschen

Je nach Gerät kann die Schnittstelle über das Kontextmenü durch eine andere Schnittstellenart ersetzt oder gelöscht werden.



Wenn Gerätevariablen der Schnittstelle im Projekt verwendet werden, wird die Schnittstelle erst ersetzt oder entfernt, wenn Sie die Variablenreferenzen löschen.

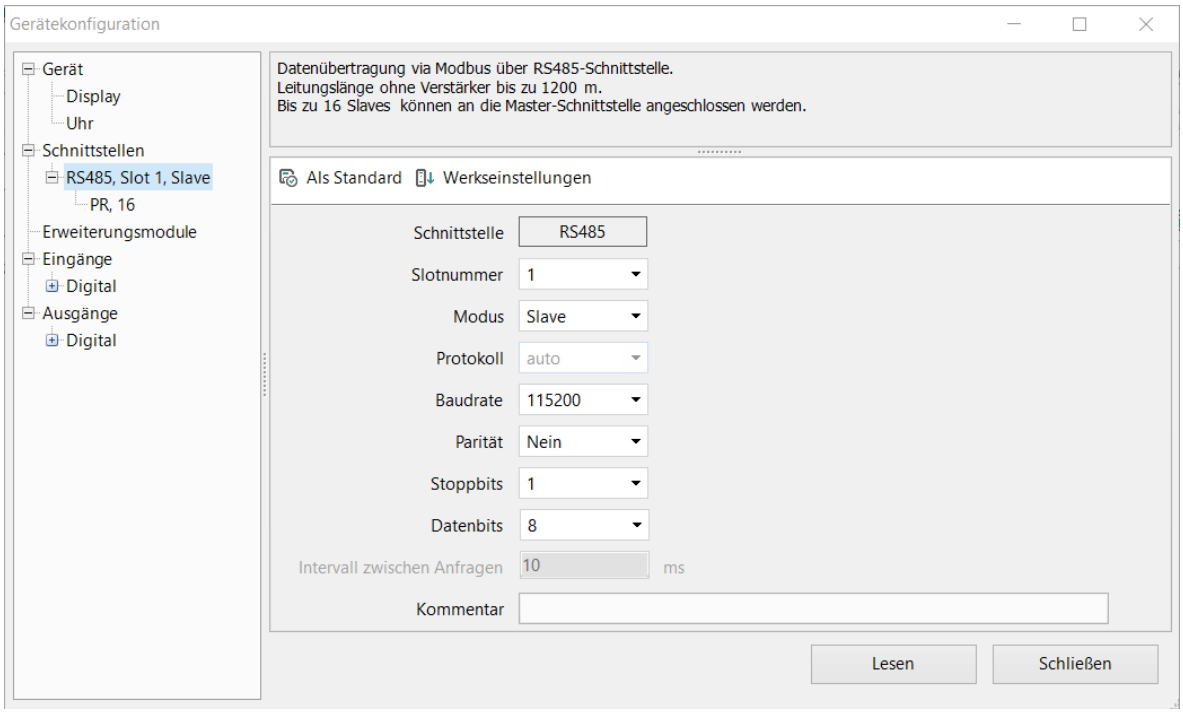
HINWEIS
Hinzufügen, Ersetzen und Entfernen von Schnittstellen sind für PR100 nicht verfügbar.

5.3.1.1 RS485-Schnittstellenkonfiguration

Um die Schnittstelleneinstellungen zu ändern, wählen Sie die Schnittstelle im Konfigurationsbaum aus.

Schnittstellenparameter für Geräte der ersten Generation

Die Schnittstelleneinstellungen werden im rechten Teil des Gerätekonfigurationsfensters angezeigt.



Im Baum werden der Schnittstellentyp (RS485), die Nummer des zugewiesenen Slots und der Modus (Master/Slave) angezeigt. Wenn im Konfigurationsbaum LoRa-Einstellungen vorhanden sind, können Sie keinen Slot für die RS485-Schnittstelle auswählen.
Modus der Schnittstelle:

– *Slave* – Das Gerät ist ein Slave, dessen Werte von einem anderen Gerät abgefragt werden.
– *Master* – Das Gerät liest und schreibt die Werte der über die Schnittstelle angeschlossenen Slave-Geräte.

Das Icon ***Als Standard*** dient dazu, die Einstellungen als Vorgabewerte für zukünftige Projekte zu speichern.

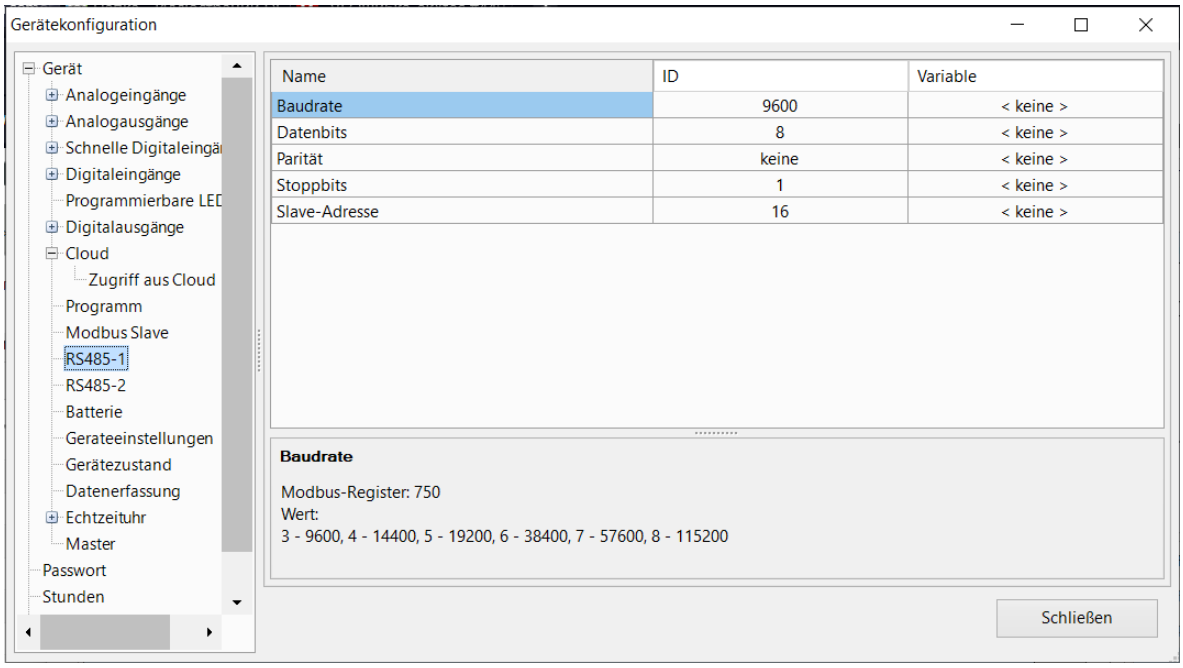
Über das Icon ***Werkseinstellungen*** werden die unveränderlichen Werkseinstellungen übernommen.
Die übrigen Parameter werden gemäß den Projektzielen und den technischen Eigenschaften anderer Geräte im Netzwerk eingegeben.

HINWEIS
Die Parameter ***Protokoll*** und ***Intervall zwischen Anfragen*** sind nur im Master-Modus verfügbar.

Mit der Schaltfläche ***Lesen*** können die aktuellen Einstellungen aus dem angeschlossenen Gerät ausgelesen werden.
Über die Schaltfläche ***Schließen*** werden die Einstellungen im Projekt gespeichert und der Dialog geschlossen.

Schnittstellenparameter für Geräte der zweiten Generation

Bei Geräten der zweiten Generation befinden sich die Schnittstellenparameter in **RS485** im Einstellungsbaum. Auf der rechten Seite des Geräteeinstellungsfensters werden die Schnittstellenparameter angezeigt. Das Einstellungsfenster sieht wie in der folgenden Abbildung aus.



Die Liste der Schnittstellenparameter hängt vom Gerät ab.
Wählen Sie Werte aus der Dropdown-Liste aus oder geben Sie sie in die Spalte **ID** ein. Es ist möglich, dem Parameter in der Spalte **Variable** eine Variable zuzuweisen.

5.3.1.2 Ethernet-Schnittstellenkonfiguration

Ethernet-Einstellungen sind nur für Geräte der zweiten Generation im Menü **Ethernet-Einstellungen** im Abschnitt **Netzwerkeinstellungen** des Einstellungsbaums verfügbar.
Das Einstellungsfenster zeigt die aktuellen Netzwerkparameter des Geräts an und legt auch neue fest.
Nach dem Speichern der Einstellungen mit der neuen IP-Adresse sollte das Gerät neu gestartet werden.

HINWEIS
Nach dem Einstellen einer neuen IP-Adresse verliert das Gerät die Verbindung zum PC. Für die neue Verbindung müssen Sie eine neue IP-Adresse angeben (siehe *Verbindung zum Gerät*).

5.3.1.3 LoRa-Schnittstellenkonfiguration

LoRa-Einstellungen sind nur für Gerätemodelle mit dem LoRa-Modul im Konfigurationsbaum verfügbar: Abschnitt **Schnittstellen** → Unterabschnitt **LoRa**. Der Datenaustausch über die LoRa-Schnittstelle erfolgt über das LoRaWAN-Protokoll. Das programmierbare Relais ist der Endknoten des LoRaWAN-Netzwerks.

Netzwerkeinstellungen

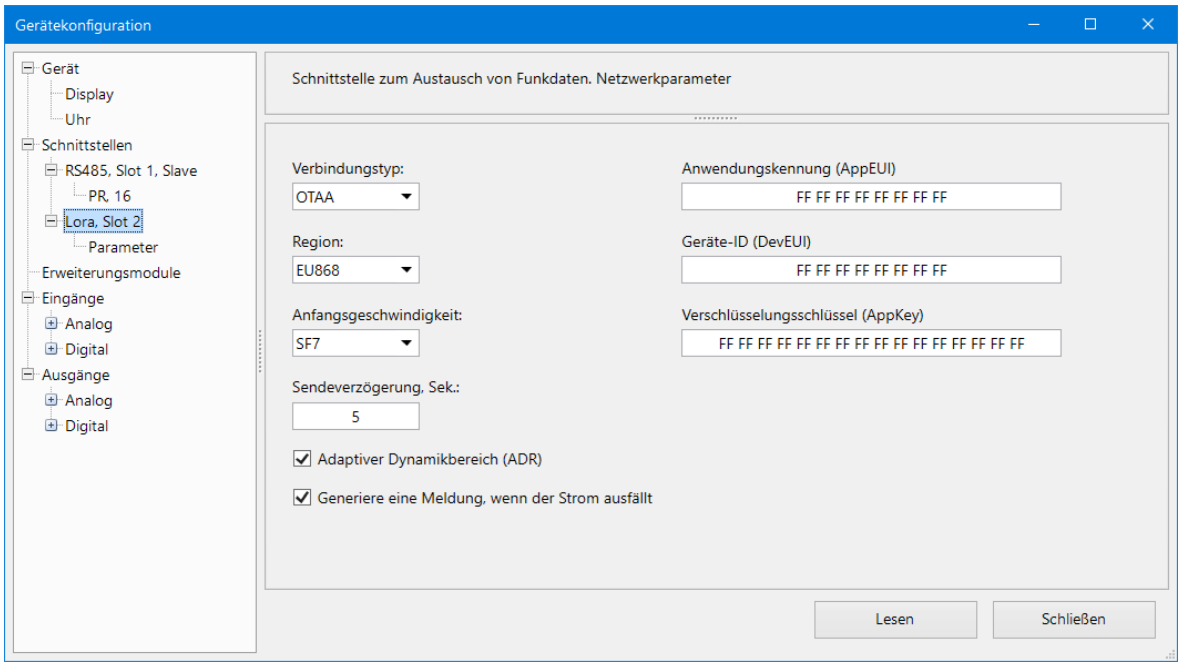
Die LoRa-Schnittstelle und die LoRaWAN-Verbindung können auf der Registerkarte **Netzwerkparameter** konfiguriert werden.
Aktivierung(Verbindungstyp) – Wählen Sie das Verfahren zur Aktivierung von LoRaWAN-Geräten im Netzwerk (OTAA/ABP):

– **OTAA** – Netzwerkverbindung mit den vom Server empfangenen Einstellungen
– **ABP** – Netzwerkverbindung mit den manuell eingestellten Einstellungen

Region – Regionsspezifisches Frequenzband von LoRaWAN
Anfangsgeschwindigkeit – Datenratenspreizungsfaktor (SF – Spreading Factor) der die Datenrate, den Übertragungsbereich und die Störfestigkeit bestimmt. SF7: maximale Rate, minimaler Bereich. SF12: minimale Rate, maximaler Bereich.
Sendeverzögerung, Sek. – Verzögerungszeit (in Sekunden) vor dem Senden von Paketen. Mit der Sendeverzögerung können Sie die Übertragung eines fertigen Pakets verschieben.

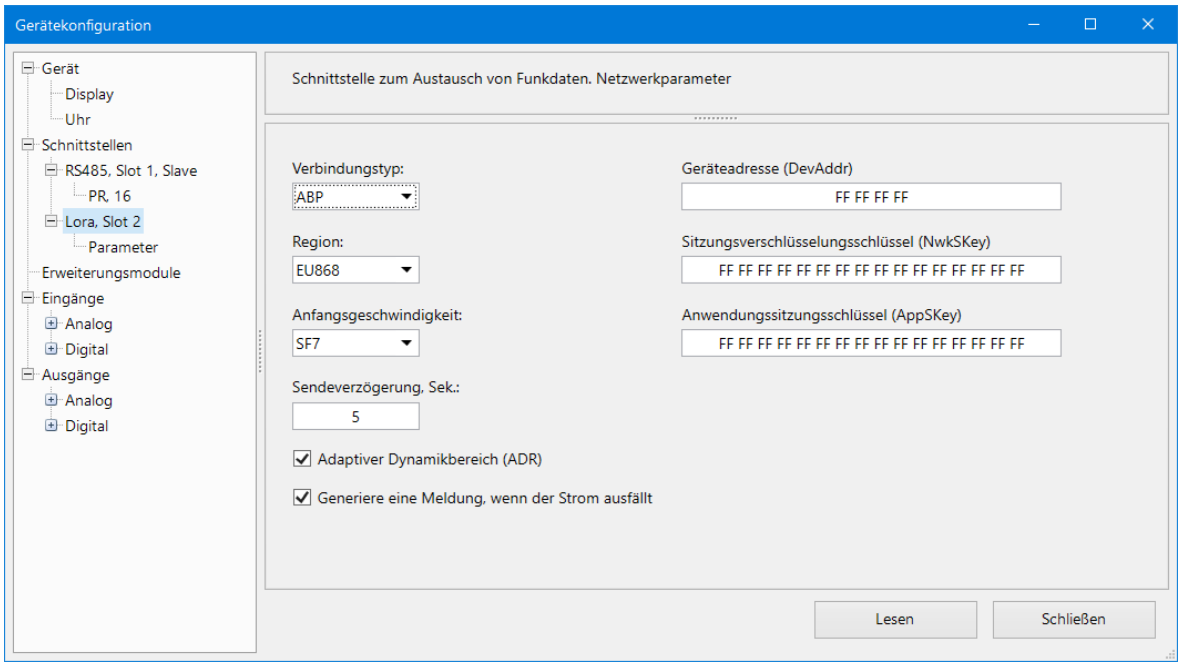
Adaptive Datenrate (ADR) – Wenn diese Option ausgewählt ist, ändert sich die anfängliche Datenrate (Anfangsgeschwindigkeit) je nach Kommunikationsqualität automatisch in die optimale Rate. Die angegebene **Anfangsgeschwindigkeit** wird als Startwert für die Optimierung verwendet.
Generiere eine Meldung, wenn der Strom ausfällt – Wenn der Parameter ausgewählt ist, sendet das Gerät bei Ausfall der Stromversorgung eine Meldung an den Server.

OTAA



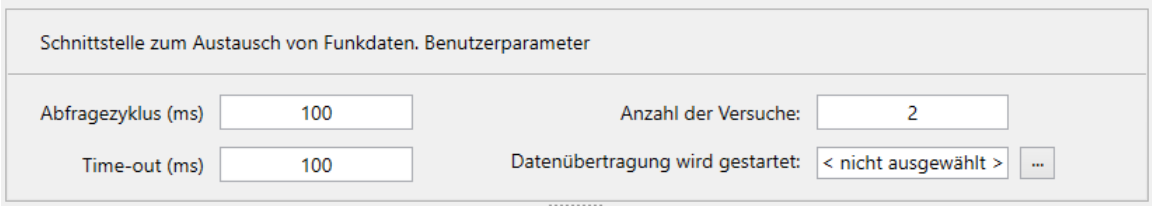
Anwendung-ID (AppEUI) – Global eindeutige 64-Bit-Kennung (16 Zeichen), die den Anwendungsanbieter des Endgeräts eindeutig identifiziert. Das Gerät verwendet den Hexadezimalcode, um die Anwendung zu identifizieren, die Daten vom Gerät empfängt und verarbeitet.
Geräte-ID (DevEUI) – Global eindeutige 64-Bit-Kennung (16 Zeichen), die ein Endgerät spezifisch identifiziert. Es wird hauptsächlich im Netzwerk verwendet, um ein angeschlossenes Endgerät sicher zu registrieren und zu aktivieren.
Anwendungsschlüssel (AppKey) – Serverschlüssel (128 Bit). Es wird verwendet, um Informationen zu verschlüsseln, die über das Netzwerk übertragen werden. Es muss vertraulich behandelt werden und auf dem Gerät und auf der Serverseite übereinstimmen.

ABP



Geräteadresse (DevAddr) – Lokale Netzwerkadresse eines Geräts. Es wird für das Nachrichtenrouting und die Geräteidentifikation während der Datenkommunikation verwendet.
Netzwerksitzungsschlüssel (NwkSKey) – Hexadezimalcode (128 Bit), der zum Verschlüsseln/ Entschlüsseln von Daten verwendet wird, die zwischen dem Gerät und dem Netzwerkserver übermittelt werden.
Anwendungssitzungsschlüssel (AppSKey) – Hexadezimalcode (128 Bit), der zum Verschlüsseln/ Entschlüsseln von Daten verwendet wird, die während einer bestimmten Kommunikationssitzung zwischen dem Gerät und dem Netzwerkserver übermittelt werden.
Die Schaltfläche **Lesen** liest alle aktuellen Parameter des angeschlossenen Geräts.
Die Schaltfläche **Schließen** zeichnet die eingegebenen Parameter im Gerät auf und schließt das Fenster **Gerätekonfiguration**. Die Parameter werden nach dem Neustart der Stromversorgung oder nach dem Laden des Programms in das Gerät wirksam.

Benutzerparameter



Abfragezyklus (ms) – Legt das Zeitintervall zwischen den Abfragezyklen.
Time-out (ms) – Zeit, die eine Abfrage dauern kann, bevor der Versuch als fehlgeschlagen gilt.
Anzahl der Versuche – Bestimmt die maximale Anzahl fehlgeschlagener Abfragen.
Datenübertragung wird gestartet – Gibt eine BOOL-Variable zur Steuerung der Abfrage an: 0 – Abfrage deaktiviert, 1 – Abfrage aktiviert. Nach der Datenübertragung wird der Wert auf 0 zurückgesetzt. Die Variable wird in der Variablentabelle ausgewählt, die durch Klicken auf die Schaltfläche "…" geöffnet wird.
Gruppenabfragen werden in LoRa-Modulen nicht unterstützt.

Benutzerdefinierte Variablen

Eine Tabelle mit Gerätevariablen befindet sich am unteren Rand des Fensters.

Um eine Variable hinzuzufügen, klicken Sie auf Schaltfläche . Die Variable wird der Tabelle hinzugefügt, wie in der folgenden Abbildung gezeigt.

Parameter in FB	Typ	Registeradresse	Kommentar
Lora1_R	BOOL	0	
Lora2_R	BOOL	1	
Lora3_R	BOOL	2	
Lora4_R	BOOL	3	

Die Variablen, die der LoRa-Schnittstelle hinzugefügt wurden, erhalten automatisch einen Postfix für den Namen:

– _R, wenn die Lesefunktion in den Variableneigenschaften ausgewählt ist
– _W, wenn die Schreibfunktion in den Variableneigenschaften ausgewählt ist

Um mehrere Variablen mit ähnlichen Einstellungen zu erstellen, verwenden Sie die Funktion Variable duplizieren. Weitere Informationen zu **Variable duplizieren** finden Sie im Abschnitt *Slave-Modus*.
Um eine Variable aus der Tabelle zu löschen, wählen Sie die gewünschte Variable aus und klicken Sie auf .

Eigenschaften der Gerätevariablen

Die Eigenschaften der Gerätevariablen werden rechts neben der Tabelle konfiguriert.

Name: Lora4_R
Typ: BOOL
Register: 3
Lesefunktion: Ja
Schreibfunktion: Nein
Kommentar:

– **Name** – Gibt den Namen der Gerätevariablen an (vom Benutzer festgelegt).
– **Type** – Definiert den Typ der Gerätevariable: Integer (Ganzzahl) oder Floating Point (Gleitkommazahl).
– **Register** – Gibt die Registeradresse der Gerätevariable an (vom Benutzer festgelegt). Der zulässige Adressbereich ist in der Bedienungsanleitung des Geräts definiert.
– **Lesefunktion/Schreibfunktion**– Aktiviert oder deaktiviert die Auswahl die Lese-/Schreibfunktion.
– **Anzahl der Register** (nur für Integer-Variablen) – Anzahl der Register, die von einer Variablen belegt werden: 1 oder 2.

– **Kommentar**– Definiert eine textliche Beschreibung des Variablenwerts zur Anzeige in der Variablentabelle.

5.3.2 Betrieb ueber Modbus-Protokoll

Allgemein

Für Geräte im Master-Modus müssen Sie ein Protokoll in den *Einstellungen* auswählen.

Modbus Master	**PR200, PR100, PR102, SMI200 Erste Generation**	**PR225, PR205, PR103 Zweite Generation**
Protokolle	Modbus RTU, Modbus ASCII (vom Benutzer auswählbar)	Modbus RTU, Modbus ASCII (vom Benutzer auswählbar) Modbus TCP
Maximale Anzahl von Slave-Geräten	16 pro Schnittstelle	32 pro Schnittstelle
Maximale Anzahl an Anfragen	255 Variablen pro Gerät Je nach Konfiguration kann eine einzelne Variable einer Leseanforderung, einer Schreibanforderung oder sowohl einer Lese- als auch einer Schreibanforderung gleichzeitig entsprechen.	192, beginnend mit ALP Version 2.7.349 und Firmware-Versionen: PR205: 1.5.0, PR103: 1.7.0 64 (in früheren Softwareversionen). Je nach Konfiguration kann eine einzelne Variable entweder einer Leseanforderung oder einer Schreibanforderung entsprechen.
Gruppenabfragen	Unterstützt ab ALP Version 2.11 und Firmware-Versionen: PR100: 3.08, PR100: 2.70, PR102: 2.70, PR200: 2.73, IPP120: 2.73. Bis zu 48 Register pro Anfrage (konfigurierbar).	Unterstützt ab ALP Version 2.10 und Firmware-Versionen: PR225: 1.9.0, PR205: 1.9.0, PR103: 1.11.0. Bis zu 16 Register pro Anfrage (konfigurierbar).
Modbus-Funktionen	0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x0F, 0x10	0x01, 0x02, 0x03, 0x04, 0x06, 0x0F, 0x10
Datentypen	BOOL, UINT16, UINT32, REAL	BOOL, UINT16, UINT32, REAL
Byte-/Registerreihenfolge vertauschen	Unterstützt, betrifft nur REAL- und UINT32-Variablen.	Unterstützt, betrifft nur REAL- und UINT32-Variablen.
Broadcast-Anfragen	Nicht unterstützt	Nicht unterstützt
Abstimmungsvorlagen	Unterstützt	Unterstützt ab ALP-Version 2.10
Wahlkontrolle	Auf Geräteebene (Abfrage aktivieren/deaktivieren) – siehe den Parameter **Abfrage**. Auf Anforderungsebene – Auslösung am Rande einer booleschen Variable (Parameter **Lese-Trigger Schreib-Trigger**) + das Kontrollkästchen **Schreiben bei Änderung**.	Auf Geräteebene (Abfrage aktivieren/deaktivieren) – siehe den Parameter **Abfrage** Auf Anforderungsebene – Auslösung am Rande einer booleschen Variable (Parameter **Lese-Trigger Schreib-Trigger**) + das Kontrollkästchen **Schreiben bei Änderung**.
Verwaltung des gewünschten Wahlzeitraums	Auf Geräteebene	Auf Anfrageebene

Modbus Master	**PR200, PR100, PR102, SMI200** **Erste Generation**	**PR225, PR205, PR103** **Zweite Generation**
Diagnostik	Auf Geräteebene (Kommunikation hergestellt/ unterbrochen) – siehe den Parameter **Status** Auf Anforderungsebene – Modbus-Fehlercode – siehe den Parameter **Status**.	Auf Geräteebene (Kommunikation hergestellt/ unterbrochen) – siehe den Parameter **Status**
Modbus TCP	Nicht unterstützt	Unterstützt Die Abfrage aller Slave-Geräte erfolgt sequenziell (es ist jeweils nur eine Client-Verbindung aktiv). Anfragen werden innerhalb einer Verbindung nacheinander gesendet. Gültiger Bereich für die Slave-Geräteadresse (Geräte-ID): 1…247.

Geräte im Slave-Modus ermitteln das Protokoll automatisch.

Modbus Slave	**PR200, PR100, PR102, SMI200** **Erste Generation**	**PR225, PR205, PR103** **Zweite Generation**
Protokolle	Modbus RTU, Modbus ASCII (Automatische Erkennung – die Antwort wird im Format der Anfrage gesendet)	Modbus RTU, Modbus ASCII (Automatische Erkennung – die Antwort wird im Format der Anfrage gesendet) Modbus TCP
Speicherbereichsmodell	Gemeinsam, mit Eingangs-/ Halteregister-Overlay	Gemeinsam, mit Eingangs-/ Halteregister-Overlay
Speichergröße	64 Register (128 Bytes)	1020 Register (2040 Bytes)
Gruppenabfragen	Unterstützt ohne künstliche Einschränkungen	
Modbus-Funktionen	0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x0F, 0x10	0x03, 0x04, 0x06, 0x0F, 0x10, 0x14, 0x15
Arbeiten mit Bitmasken	Die Bitmaskenparameter können entweder mit der Funktion 0x03 oder 0x01 ausgelesen werden. Bei Verwendung von 0x01 muss die Registernummer mit 16 multipliziert und die Bitnummer addiert werden.	Bitmaskenparameter können nur mit der Funktion 0x03 ausgelesen werden.
Datentypen	UINT16, REAL	UINT16, REAL
Byte-/Register-Reihenfolge (Byte-Swapping)	Unterstützt, betrifft aber nur die Systemvariablen REAL und UINT32.	Nicht unterstützt. Beim Abfragen von Variablen, die zwei oder mehr Register belegen, sollten die folgenden Einstellungen verwendet werden: – Byte-Reihenfolge – Big-Endian (höchstwertiges Byte zuerst)

Modbus Slave	PR200, PR100, PR102, SMI200 Erste Generation	PR225, PR205, PR103 Zweite Generation
		– Registerreihenfolge – Little-Endian (Register mit der geringsten Signifikanz zuerst)
Broadcast-Anfragen	Unterstützt	Nicht unterstützt
Modbus TCP	Nicht unterstützt	Unterstützt werden bis zu 4 gleichzeitige Client-Verbindungen (eine fünfte Verbindung ist unabhängig und für akYtec Cloud reserviert). Die Geräte-ID in der Anfrage wurde nicht validiert.
Persistente Speicherung	Alle Netzwerkvariablen sind stets nichtflüchtig (bleiben erhalten).	Ab ALP Version 2.9 ist die Nichtvolatilität jeder Netzwerkvariablen vom Benutzer konfigurierbar. In früheren Versionen waren alle Netzwerkvariablen immer nichtflüchtig (blieben erhalten).
Export der Registerzuordnung	Der Export in die akYtec Cloud wird unterstützt, ebenso wie der Zugriff auf eine *.csv*-Datei.	Der Export in die akYtec Cloud wird unterstützt, ebenso wie der Zugriff auf eine *.csv*-Datei.

Um den Datenaustausch in einem Netzwerk über eine Kommunikationsschnittstelle zu organisieren, wird ein Netzwerk-Master benötigt – ein Gerät, das im Master-Modus arbeitet. Die Hauptfunktion des Netzwerk-Masters besteht darin, den Datenaustausch zu initiieren.
An der Kommunikationsleitung ist nur ein Gerät im Master-Modus zulässig.

Anzahl der Kommunikationsschnittstellen

Die Anzahl der Kommunikationsschnittstellen eines Geräts hängt von der Anzahl der verfügbaren Anschlussplätze ab. Unterstützt das Gerät bis zu zwei Kommunikationsschnittstellen, können diese für den Betrieb in zwei unabhängigen Netzwerken konfiguriert werden.

Standard-Modbus-Ausnahmecodes

- **01** – Der in der Anfrage empfangene Funktionscode ist für den Server nicht zulässig (Ungültige Funktion).
- **02** – Die in der Anfrage empfangene Datenadresse ist für den Server nicht zulässig (Ungültige Datenadresse).
- **03** – Der im Abfragedatenfeld enthaltene Wert ist für den Server nicht zulässig (Ungültiger Datenwert).

Funktionen und Datenbereiche

Bei einer Anfrage greift der Master mithilfe einer Funktion auf einen der Speicherbereiche des Slaves zu. Ein Speicherbereich wird durch den Datentyp (Bits/Register) und den Zugriffstyp (nur lesend / lesend-schreibend) charakterisiert.

Tabelle 5.1 Modbus-Protokoll – Datenbereiche

Datenbereich	Bezeichnung	Datentyp	Zugriffstyp
Spulen	0x	BOOL	Lesen/Schreiben
Digitale Eingänge	1x	BOOL	Nur lesen

Datenbereich	Bezeichnung	Datentyp	Zugriffstyp
Eingangsregister	3x	INT	Nur lesen
Halteregister	4x	INT	LesenSchreiben

Jeder Datenbereich besteht aus einer bestimmten Anzahl von Zellen, die vom jeweiligen Gerät abhängt. Jede Zelle hat eine eindeutige Adresse. Bei konfigurierbaren Geräten stellt der Hersteller eine **Registerkarte** bereit, die Informationen über die Zuordnung zwischen Geräteparametern und ihren Adressen enthält. Bei programmierbaren Geräten erstellt der Benutzer eine solche Karte selbstständig mithilfe einer Programmierumgebung. Einige Geräte kombinieren beide beschriebenen Fälle – ihre Registerkarte hat einen festen Teil und einen Teil, den der Benutzer je nach Bedarf ergänzen kann.
Bei einigen Geräten sind die Speicherbereiche übereinander angeordnet (z. B. **0x** und **4x**), was bedeutet, dass der Benutzer mit verschiedenen Funktionen auf dieselben Register zugreifen kann.
Eine **Funktion** definiert die Operation (Lesen/Schreiben) und den Speicherbereich, in dem diese Operation ausgeführt wird.

Tabelle 5.2 Hauptfunktionen des Modbus-Protokolls

Funktionscode	Funktionsname	Befehl ausgeführt
1 (0x01)	Spulenstatus lesen	Liest Werte von mehreren Spulen.
2 (0x02)	Diskrete Eingänge lesen	Liest Werte von mehreren diskreten Eingängen.
3 (0x03)	Lesen Sie die Haltelisten	Liest Werte aus mehreren Halteregistern.
4 (0x04)	Eingangsregister lesen	Liest Werte aus mehreren Eingangsregistern.
5 (0x05)	Einzelspule erzwingen	Schreibt einen Wert in eine einzelne Spule.
6 (0x06)	Voreingestelltes Einzelregister	Schreibt einen Wert in ein einzelnes Halteregister.
15 (0x0F)	Mehrere Windungen erzwingen	Schreibt Werte an mehrere Spulen.
16 (0x10)	Mehrere voreingestellte Register	Schreibt Werte in mehrere Halteregister.

In verschiedenen Dokumenten können identische Bezeichnungen je nach Kontext unterschiedliche Bedeutungen haben. Beispielsweise wird das Präfix **0x** häufig verwendet, um ein Hexadezimalsystem anzugeben. Daher könnte **0x30** in einem Fall „Bit 30 des Spulenspeicherbereichs“ bezeichnen und in einem anderen Fall „Adresse 30 im Hexadezimalsystem (HEX)“ bedeuten (wobei diese Adresse sich auf einen beliebigen Speicherbereich beziehen kann).
Die Abfrage eines Slaves kann einzeln oder in Gruppen erfolgen. Bei der Einzelabfrage liest der Master jeden Parameter des Slaves mit einem separaten Befehl aus.
Bei **Gruppenabfragen** liest der Master mehrere Parameter gleichzeitig mit einem einzigen Befehl. Dies ist nur möglich, wenn die Adressen dieser Parameter in der Registerzuordnung strikt aufeinanderfolgend und lückenlos sind. Gruppenabfragen tragen zur Reduzierung des Netzwerkverkehrs und der Abfragezeit bei. In einigen Fällen können Gruppenabfragen jedoch aufgrund der individuellen Eigenschaften des Geräts nicht oder nur eingeschränkt verwendet werden.

Abfragezyklus und Antwort-Timeout

Das Gerät sendet Abfragen in Intervallen, die durch den Parameter **Abfragezyklus** des abgefragten Geräts festgelegt sind. Die Wartezeit auf eine Antwort wird durch den Parameter **Antwort -Timeout** bestimmt. Wird innerhalb dieser Zeit keine Antwort empfangen, wiederholt das Gerät die Anfrage N-mal, wobei N der Parameter **Wiederholungen** ist. Erfolgt weiterhin keine Antwort, wird die **Status-Variable** auf 0 (Falsch) gesetzt.

Netzwerkvariablentypen

Datentyp		Größe im Slave-Modus	Größe im Master-Modus
Integer (Ganzzahl)	UINT	16 Bit (1 Register)	16 Bit (1 Register)
	UDINT	-	32 Bit (2 Register)
Floating point (Gleitkommazahl)	REAL	32 Bit (2 Register)	32 Bit (2 Register)
Boolean	BOOL	-	1 bit

Register- und Byte-Reihenfolge

Wenn das Gerät im Slave-Modus arbeitet, beeinflussen die Register-/Byte-Reihenfolgeeinstellungen nur, wie der Master Gleitkomma-Systemnetzwerkvariablen (analoge Ein-/Ausgänge) liest.
Bei benutzerdefinierten Netzwerkvariablen im Slave-Modus werden die Daten immer im folgenden Format übertragen:

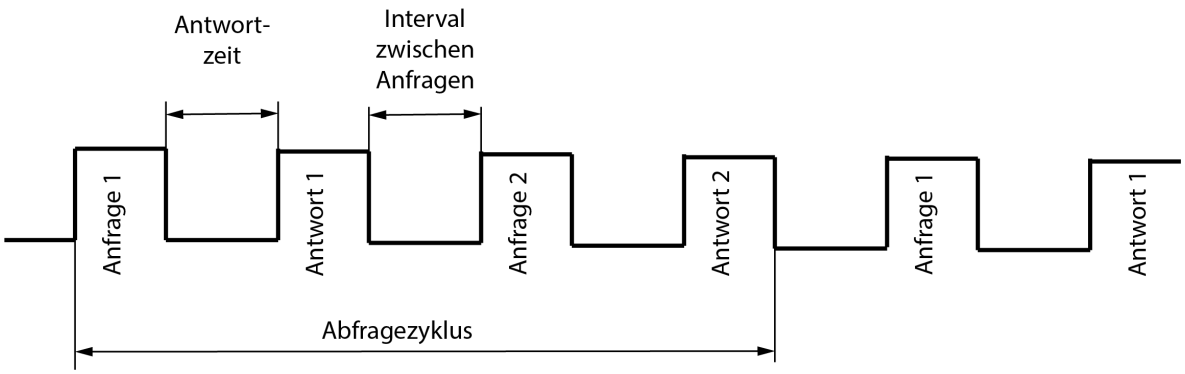
– Wichtigstes Register zuerst: Nein
– Höchstwertigstes Byte zuerst: Ja

Automatische Programmzykluszeitanpassung

Die Geräte können die Programmzykluszeit automatisch an die Komplexität des Algorithmus anpassen. Diese automatische Anpassung der Programmzykluszeit beeinflusst die Funktion der Kommunikationsschnittstelle, da Abfragen in der verbleibenden Zykluszeit nach Abschluss der Programmausführung verarbeitet werden.
Gemäß dem Algorithmus zur Zykluszeitanpassung beträgt die Mindestanzahl an Master-Aufrufen bis zu 50 pro Sekunde. Kann der Master innerhalb dieses Zeitraums nicht alle Geräte abfragen, sollte der Algorithmus angepasst werden, um die Anzahl der Abfragen zu optimieren.

Abfragezyklus und Zuteilungsregeln

Das Gerät reiht alle Anfragen in eine Warteschlange ein. Ist die Warteschlange kurz, führt das Gerät alle Abfrage-Antwort-Zyklen durch und wartet anschließend die festgelegte Zeitspanne ab. Ist die Warteschlange lang und kann nicht innerhalb der festgelegten Zeitspanne verarbeitet werden, fragt das Gerät alle erforderlichen Parameter in maximal mögliche Periode ab, die länger sind als die in der Konfiguration festgelegte Zeitspanne.
Zeitdiagramm für die Abfrage:



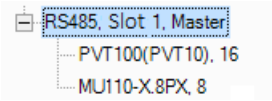
Für eine maximale Geschwindigkeit von 115200 bit/s beim Abfragen zwei REAL-Variablen von einem einzelnen Slave-Gerät unter "idealen" Bedingungen (kurze Kommunikationsleitung, keine Störungen) sollte Folgendes gelten:

– Antwortzeit ab Beginn der Anfrage: 2,64 ms
– Die nächste Anfrage beginnt frühestens nach 12 ms
– Gesamtzeit für eine REAL-Registeranforderung: 3,4 ms

Unter den oben genannten Bedingungen kann das Gerät 83 Anfragen pro Sekunde senden. Dieser Wert gilt auch für andere Geräte mit ähnlichen Zeitcharakteristika. Mit zunehmender Komplexität des Algorithmus während der Entwicklung verlängert sich die Zykluszeit, und die Anzahl der Anfragen pro Sekunde sinkt.
Die Abfrageperiode hängt vom Algorithmus und der Häufigkeit der Abfrage bestimmter Parameter ab. Es wird empfohlen, die Abfrageperiode auf 1 Sekunde einzustellen. In diesem Fall kann das Gerät bis zu 50 Variablen abfragen.

Abfrage mehrerer Geräte im Netzwerk

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



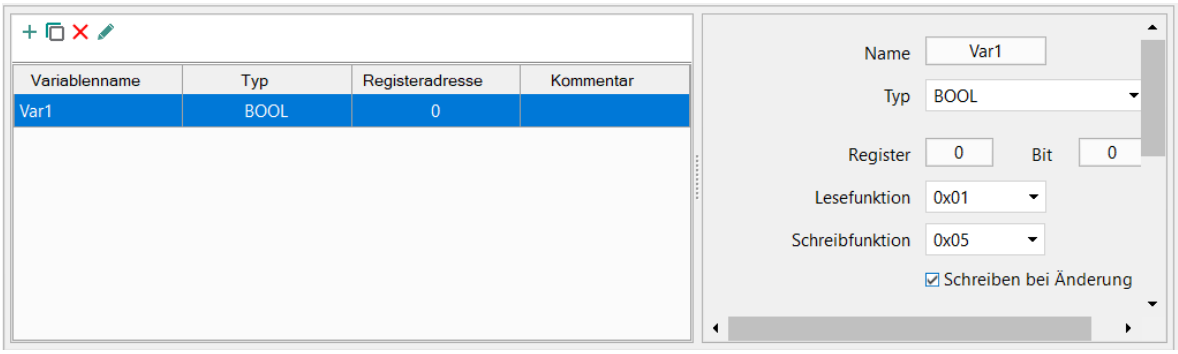
Der ***Abfragezyklus*** kann für jeden Slave einzeln festgelegt werden.

Berechnung der Adresse und des Bits zum Lesen einer booleschen Variable im Master-Modus

In manchen Fällen ist es erforderlich, die Registeradresse und die Bitnummer eines Slave-Geräts zu berechnen. Um die Registeradresse und die Bitnummer aus der Dokumentation des abgefragten Geräts zu ermitteln, ermitteln Sie die Position der Bitvariablen, z. B. *Bit 1400*. Gehen Sie dann wie folgt vor:

1. Berechne die Registeradresse: 1400 ÷ 16 = 87
2. Berechne die Bitanzahl: 1400 mod 16 = 8

Die erhaltene Adresse und Nummer werden beim Konfigurieren der Geräteabfrage in die Felder **Register** und **Bit** eingegeben.



5.3.2.1 Master-Modus

Im Master-Modus fragt das Gerät andere, über die Schnittstelle verbundene Slave-Geräte ab. Die **Gruppenabfrage** ermöglicht das Abfragen ähnlicher Parameter der angeschlossenen Slave-Geräte.
Zum Abfragen müssen Geräte hinzugefügt und konfiguriert werden. Um ein Gerät hinzuzufügen, klicken Sie im Gerätekonfigurationsfenster mit der rechten Maustaste auf den Namen der Master-Schnittstelle – für *Geräte der ersten Generation* – oder auf den Eintrag Modbus Master für *Geräte der zweiten Generation* – und wählen Sie im Kontextmenü Gerät hinzufügen.

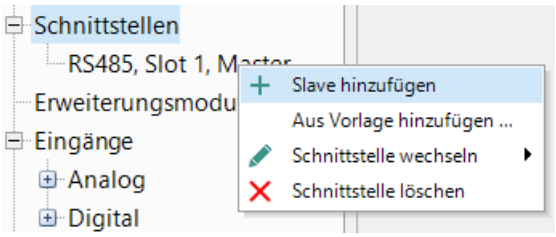


Abb. 5.1 Hinzufügen eines Slave-Geräts (Gerät der ersten Generation)

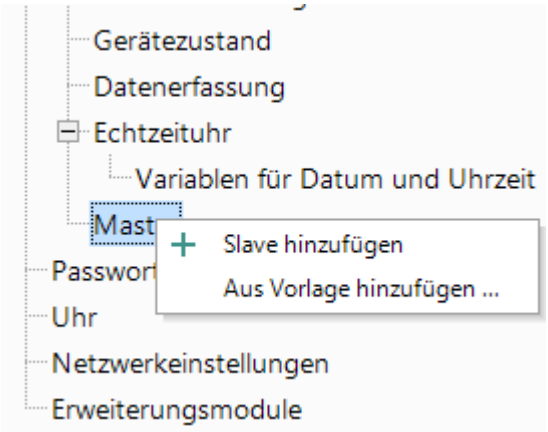


Abb. 5.2 Hinzufügen eines Slave-Geräts (Gerät der zweiten Generation)

Das hinzugefügte Gerät erscheint in der Baumstruktur als neuer Zweig für die entsprechende Schnittstelle. Der Gerätename setzt sich aus dem Namen, der Geräteadresse und der Kommunikationsschnittstelle (bei Geräten der zweiten Generation) zusammen.

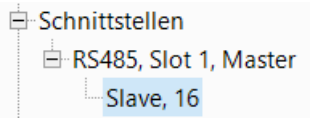


Abb. 5.3 Slave-Gerät im Parameterbaum (Geräte der ersten Generation)

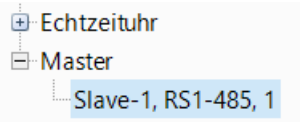


Abb. 5.4 Slave-Gerät im Parameterbaum (Geräte der zweiten Generation)

Gruppenabfragen

Gruppenabfragen ermöglichen das Auslesen mehrerer Register mit einem einzigen Befehl. Die maximale Anzahl der Register in einer Gruppenabfrage hängt von der Puffergröße des Geräts, dem Modbus-Protokolltyp und der verwendeten Lese-/Schreibfunktion ab. Bei *Geräten der ersten Generation* beträgt die maximale Größe einer Gruppenabfrage 48 Register. Bei *Geräten der zweiten Generation* sind es 16 Register. Die Bildung einer Gruppenabfrage erfolgt nach folgenden Regeln:

1. Variablen müssen in der Registerzuordnung strikt sequenziell angeordnet sein – ohne Lücken oder ungenutzte Register/Bits dazwischen.
2. Den Variablen in der Abfrage müssen identische Lese-/Schreibfunktionscodes zugewiesen sein.

   Ausnahme: Die Funktionen **0x05** und **0x06** sind nicht zulässig – Gruppenabfragen werden dafür nicht unterstützt.
3. Variablen werden nur dann zu einer Gruppenabrage zusammengefasst, wenn sie den gleichen Abfragetyp aufweisen (periodisch, auf Befehl oder bei Änderung).
4. Nur Variablen kompatibler Größe können in einer Gruppenabfrage zusammengefasst werden:
   – **Boolesche Variablen**, können nur mit anderen booleschen Variablen gruppiert werden.
   – **Integer (1 Register)** , kann nur mit anderen Ganzzahlvariablen (1 Register) gruppiert werden.
   – **Integer (2 Register)**, können mit Ganzzahlvariablen (2 Register) und Gleitkommavariablen gruppiert werden.
   – **Floating-Point-Variablen**, können mit Ganzzahlvariablen (2 Register) und Gleitkommavariablen gruppiert werden.

Um den Zeitaufwand für die Datenerfassung von allen Geräten zu minimieren, wird Folgendes empfohlen:

– Wenn ein oder mehrere Slave-Geräte getrennt oder nicht verfügbar sind, wird empfohlen, im Programm eine Logik zu implementieren, die das Polling für diese Geräte deaktiviert, oder den

Parameter „Time-out“ für sie minimiert. Andernfalls erhöht sich die Polling-Zeit proportional zur Anzahl der getrennten Geräte und dem für ihren ***Time-out***-Parameter festgelegten Wert.

– Bei der Festlegung des Parameters „Abfragezyklus“ sind die Anzahl der Slave-Geräte und die Gesamtzahl der Abfragen zu berücksichtigen. Wird das im Parameter „Abfragezyklus“ festgelegte Zeitlimit für die Verarbeitung aller Abfragen überschritten, wird dieser Parameter ignoriert.

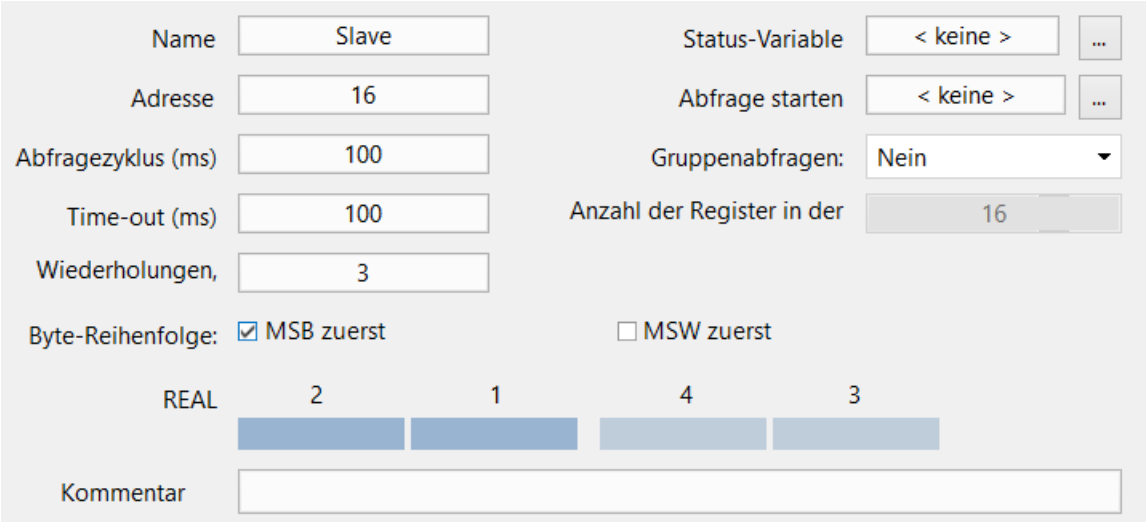
Die Verwendung identischer Variablenadressen und -namen für jedes Gerät ist zulässig.

Master-Modus für Geräte der ersten Generation

HINWEIS
Eine Liste der Geräte der ersten und zweiten Generation finden Sie im Abschnitt *Einführung*.

Slave-Geräteparameter

Um Geräteparameter zu ändern, klicken Sie im Konfigurationsbaum auf den Gerätenamen – die Parameter werden im rechten Teil des Gerätekonfigurationsfensters angezeigt.



– ***Name***: Gibt den Namen des Slaves an, der im Baum angezeigt wird.
– ***Adresse***: Gibt die Netzwerkadresse des Slaves an.
– ***Abfragezyklus (ms)***: Definiert das Zeitintervall zwischen den Abfragen. Eine Abfrage umfasst alle für den Slave definierten Variablen. Der gültige Bereich liegt bei 0–65535 ms.
– ***Time-out (ms)***: Legt die maximale Dauer einer Anfrage fest, bevor sie als fehlgeschlagen gilt. Der gültige Bereich liegt bei 0 und 65535 ms. Es wird empfohlen, den Parameter innerhalb von 300-500 ms einzustellen.
– ***Wiederholungen, max.***: Bestimmt die maximale Anzahl fehlgeschlagener Anfragen, bevor die Abfrage abgebrochen und der Gerätestatus geändert wird. Der gültige Bereich liegt bei 0–255.
– ***Status-Variable***: Ermöglicht die Auswahl einer BOOL-Variable über das Icon “…” zur Aufzeichnung des Gerätestatus:
  – 1 – Signalisiert, dass das Gerät einwandfrei funktioniert.
  – 0 – Signalisiert, dass das Gerät nicht verbunden ist.
– ***Abfrage starten***: Wählen Sie über das Icon “…” eine BOOL-Variable aus, um die Abfrage zu steuern:
  – 0 – Deaktiviert die Abfrage.
  – 1 – Aktiviert die Abfrage.

HINWEIS
Variablen, die mit den Parametern **Status-Variable** und **Abfrage starten** verknüpft sind, können in der Programmlogik verwendet werden.

- ***Gruppenabfragen***: Ermöglicht eine Sammelabfrage aufeinanderfolgender Register zur Erhöhung des Datendurchsatzes.
- ***Anzahl der Register in der Abfrage***: Von 2 bis 48.
- ***Byte-Reihenfolge***: Bestimmt die Reihenfolge der Bytes im Paket.
- ***MSW zuerst***: Legt die Registerreihenfolge bei Zweiregistervariablen fest.
- ***MSB zuerst***: Legt die Byte-Reihenfolge im Register fest.
- ***Kommentar***: Definiert einen Beschreibungstext.

Beispiel:
Ziel: Lesen des Werts „-250.123“.
Abhängig von den Einstellungen für Register- und Byte-Reihenfolge wird der Wert unterschiedlich übertragen.

Register-/Byte-Reihenfolge-Einstellung	Übertragene Nummer
Byte-Reihenfolge: ☐ MSB zuerst ☐ MSW zuerst 2 1 4 3	Niederstwertiges Byte 01111101 Höchstwertiges Byte 00011111 Niederstwertiges Byte 01111010 Höchstwertiges Byte 11000011 Niederstwertiges Wort Höchstwertiges Wort
Byte-Reihenfolge: ☑ MSB zuerst ☐ MSW zuerst 2 1 4 3	Höchstwertiges Byte 00011111 Niederstwertiges Byte 01111101 Höchstwertiges Byte 11000011 Niederstwertiges Byte 01111010 Niederstwertiges Wort Höchstwertiges Wort
Byte-Reihenfolge: ☐ MSB zuerst ☑ MSW zuerst 2 1 4 3	Niederstwertiges Byte 01111010 Höchstwertiges Byte 11000011 Niederstwertiges Byte 01111101 Höchstwertiges Byte 00011111 Höchstwertiges Wort Niederstwertiges Wort
Byte-Reihenfolge: ☑ MSB zuerst ☑ MSW zuerst 2 1 4 3	Höchstwertiges Byte 11000011 Niederstwertiges Byte 01111010 Höchstwertiges Byte 00011111 Niederstwertiges Byte 01111101 Niederstwertiges Wort Höchstwertiges Wort

Variablen des Slave-Geräts

Die Tabelle mit den abgefragten Gerätevariablen befindet sich am unteren Rand des Fensters.

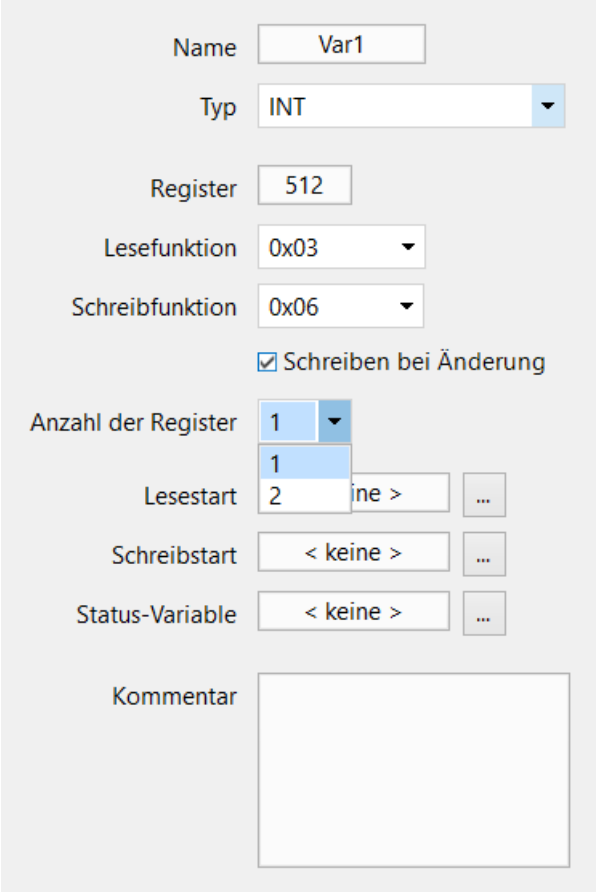
Fügen Sie eine Variable hinzu, indem Sie auf das Symbol + klicken. Die Variable wird in die Tabelle eingefügt, wie in der folgenden Abbildung gezeigt.

Variablenname	Typ	Registeradresse	Kommentar
Var1	BOOL	0	
Var2	BOOL	0	

Um eine Variable aus der Tabelle zu löschen, wählen Sie sie aus und klicken Sie auf die Schaltfläche ✕ .
Um mehrere Variablen mit denselben Einstellungen zu erstellen, wählen Sie eine Variable in der Tabelle aus und klicken Sie auf das Symbol ***Duplizieren***.

HINWEIS
Weitere Informationen zu **Variable duplizieren** finden Sie im Abschnitt *Slave-Modus*.

Eigenschaften der Netzwerkvariablen für das Slave-Gerät



Die Eigenschaften einer abgefragten Gerätevariablen werden rechts neben der Tabelle konfiguriert:

– **Name** – Gibt den Namen an, der in der *Variablentabelle* angezeigt wird.

> **HINWEIS**
> Vermeiden Sie die Verwendung identischer Variablennamen.

– **Typ** – Definiert den *Variablentyp* (BOOL, INT, oder REAL).
– **Bit (nur für boolesche Variablen)** – Gibt die zu lesende Bitnummer an.
– **Register/Registeradresse** – Gibt den vom Gerät gelesenen Registerwert an, der in der Tabelle angezeigt wird.
– **Lesefunktion/Schreibfunktion** – Aktiviert oder deaktiviert die Lese-/Schreibfunktion.

> **HINWEIS**
> **Schreiben bei Änderung** – Ist diese Funktion aktiviert, sendet der Master bei jeder Änderung des Variablenwerts automatisch eine Schreibanforderung an das Slave-Gerät.

– **Anzahl der Register** – Gibt die Anzahl der von der Variable belegten Register an (1 oder 2).
– **Lesestart** – Weist einer booleschen Befehlsvariable einen Wert zu, um das Lesen der konfigurierten Variable zu erzwingen. Wechselt die Befehlsvariable von 0 auf 1, wird eine Leseanforderung ausgelöst. Nach erfolgreichem Abschluss wird der Wert automatisch von 1 auf 0 zurückgesetzt.
– **Schreibstart** – Weist einer booleschen Befehlsvariable einen Wert zu, um einen Schreibvorgang der konfigurierten Variable zu erzwingen. Wechselt die Befehlsvariable von 0 auf 1, wird eine Schreibanforderung ausgelöst. Nach erfolgreichem Abschluss wird der Wert automatisch von 1 auf 0 zurückgesetzt.
– **Status-Variable** – Weist einer Integer-Variable einen Wert zu, der den *Modbus*-Fehlercode speichert, sofern ein Fehler auftritt.
– **Kommentar** – Definiert einen Beschreibungstext der Variable zur Anzeige in der *Variablentabelle*.

HINWEIS
Befehlsbasiertes Lesen und Schreiben ermöglicht das erzwungene Lesen oder Schreiben des Wertes einer Variablen und bietet so eine flexiblere Kontrolle über diese Operationen. Verschiedene Lese- und Schreibmodi können kombiniert werden, zum Beispiel:
- Zeitbasiertes Lesen
- Ereignisbasiertes Lesen/Schreiben
- Schreiben bei Änderung (Standardeinstellung)

Master-Modus für die Geräte der zweiten Generation

Wählen Sie im Parameterbaum des Geräts den Knoten **Master**, um die Abfrageparameter für verbundene Geräte zu konfigurieren.

HINWEIS
Die maximale Anzahl von Geräten über alle Schnittstellen hinweg beträgt 32.

Klicken Sie im Parameterbaum auf den Namen eines abgefragten Geräts, um dessen Parameter zu ändern. Die verfügbaren Parameter werden im rechten Fensterbereich angezeigt: oben die Geräteparameter, unten die Netzwerkvariablen.

HINWEIS
Ändern Sie die Netzwerkeinstellungen von Slave-Geräten optional über externe Software (*akYtec ToolPro*).

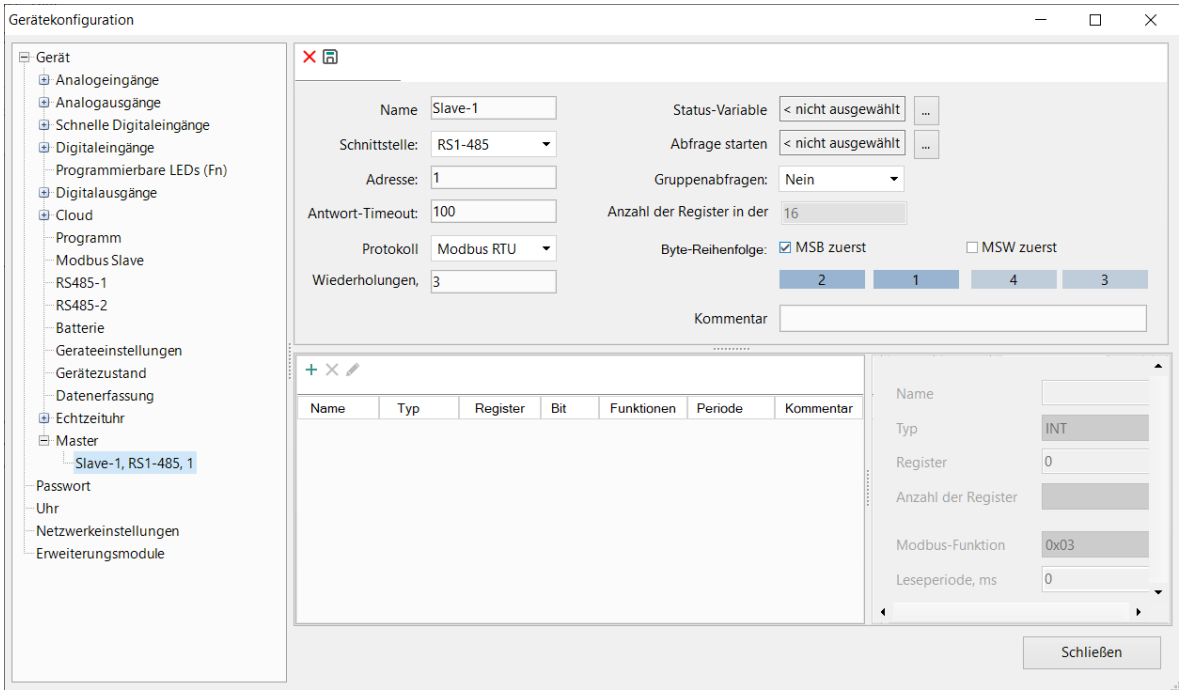


Abb. 5.5 Slave-Geräteparameter

- ***Name*** – Gibt den Gerätenamen an, der im Einstellungsbaum angezeigt wird.
- ***Schnittstelle*** – Legt die Schnittstelle fest, über die das Gerät verbunden ist. Die verfügbaren Parameter hängen von der gewählten Schnittstelle ab.
- ***Adresse*** – Gibt die Netzwerkadresse des Geräts an.
- ***Antwort-Timeout, ms*** – Definiert die Zeit, nach der eine Abfrage als fehlgeschlagen gilt. Gültiger Bereich: 10–10 000 ms.
- **Protokoll**
- ***Wiederholungen, max.*** – Bestimmt die maximale Anzahl fehlgeschlagener Anfragen. Der gültige Bereich liegt bei 0–3.
- ***Status-Variable*** – Ermöglicht die Auswahl einer BOOL-Variable in der *Variablentabelle* über das Icon "…" zur Aufzeichnung des Gerätestatus:
  - 1 – Signalisiert, dass das Gerät einwandfrei funktioniert.

  - 0 – Signalisiert, dass das Gerät nicht verbunden ist.
- ***Abfrage starten*** – Wählen Sie über das Icon "…" eine BOOL-Variable aus, um die Abfrage zu steuern:
  - 0 – Deaktiviert die Abfrage.
  - 1 – Aktiviert die Abfrage.
- ***Gruppenabfragen*** – Aktiviert die Abfrage von mehr als einer Variablen.
- ***Anzahl der Register in einer Anfrage*** – Von 2 bis 16.
- ***Byte-Reihenfolge*** – Legt die Byte-Reihenfolge im Datenpaket fest.
- ***Kommentar*** – Definiert einen Beschreibungstext des Geräts.

Spezifische Parameter für über Ethernet verbundene Geräte:

- ***IP-Adresse*** – Gibt die eindeutige Netzwerkadresse des Geräts an. Gültiger Bereich: 0.0.0.0–255.255.255.255.
- ***Port*** – Gibt die Portnummer an. Gültiger Bereich: 0–65535.

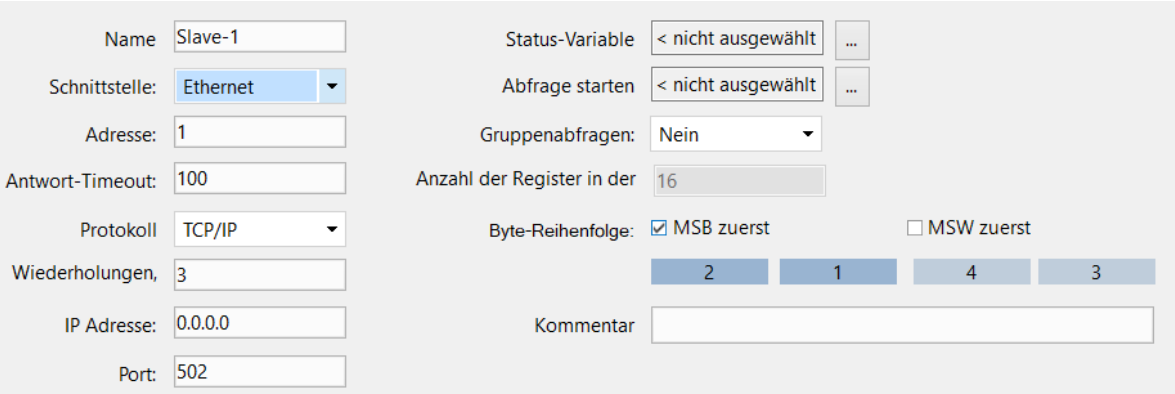


Abb. 5.6 Spezifische Parameter für ein über Ethernet angeschlossenes Gerät

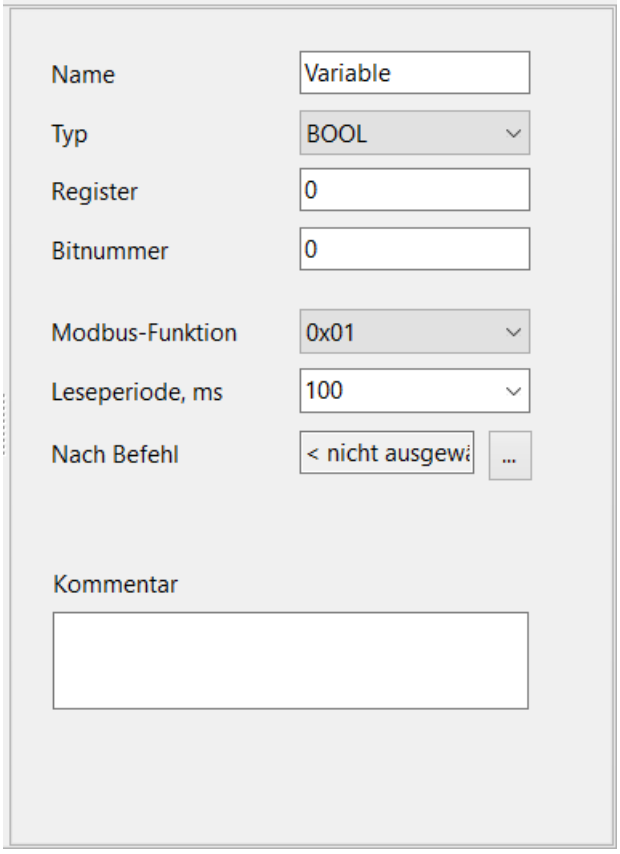


Abb. 5.7 Eigenschaften der Netzwerkvariablen für das Slave-Gerät

- ***Name*** – Gibt den Namen an, der in der *Variablentabelle* angezeigt wird.
- ***Typ*** – Definiert den *Typ* der Variable (BOOL, INT oder REAL).
- **Register** – Gibt den vom Gerät gelesenen Registerwert an, der in der Tabelle angezeigt wird.
- ***Bit (nur boolesche Variablen)*** – Gibt die zu lesende Bitnummer an.
- ***Anzahl der Register (nur für Integer-Variablen)*** – Legt die Anzahl der von der Variable belegten Register fest (1 oder 2).
- ***Modbus-Funktion*** – Aktiviert oder deaktiviert die Lese-/Schreibfunktion.

Die Liste der zu konfigurierenden Parameter hängt von der Wahl der Schreib-/Lesefunktion ab.
Parameter der Lesefunktion:

- ***Leseperiod*** – Definiert das Zeitintervall zwischen den Abfragen.
- ***Lesebefehl*** – Weist eine BOOL-Variable zu, deren Änderung das Lesen auslöst.

Parameter der Schreibfunktion:

- ***Schreibperiode*** – Definiert das Zeitintervall zwischen Schreibvorgängen.
- ***Schreibbefehl*** – Weist eine BOOL-Variable zu, deren Änderung das Schreiben auslöst.
- ***Schreiben bei Änderung*** – Löst bei Änderung des Variablenwerts automatisch einen Schreibvorgang im Slave-Gerät aus.
- ***Kommentar*** – Definiert einen Beschreibungstext, der in der *Variablentabelle* angezeigt wird.

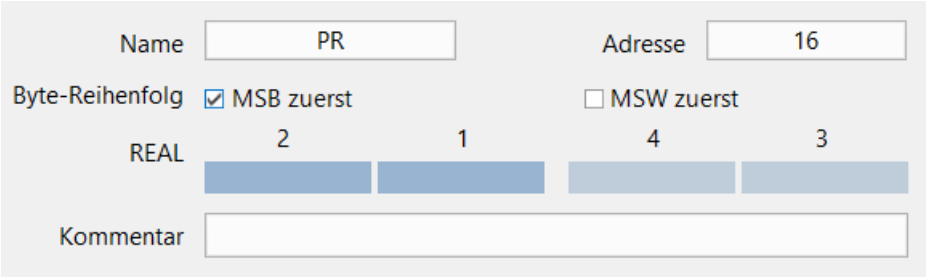
5.3.2.2 Slave-Modus

Im Slave-Modus stellt das Gerät Daten bereit, die von einem anderen Gerät im Netzwerk gelesen werden können; es initiiert keine eigene Abfrage.

Slave-Modus für Geräte der ersten Generation

Um die Datenübertragungsparameter zu konfigurieren, klicken Sie im Baum auf den Gerätenamen (PR, standardmäßig 16).

Geräteparameter



Im oberen Fensterbereich werden die gemeinsamen Parameter für den Datenaustausch eingestellt.

– ***Name*** – Gibt den Namen des Masters an, der im Baum angezeigt wird.
– ***Adresse*** – Gibt die Netzwerkadresse des Masters an.
– ***Kommentar*** – Definiert einen Beschreibungstext.

Im Slave-Modus beeinflussen die Einstellungen **MSW zuerst** und **MSB zuerst** nur die Register- bzw. Byte-Reihenfolge der Systemnetzwerkvariablen.
Überträgt benutzerdefinierte Projektnetzwerkvariablen im Slave-Modus immer im folgenden Format:

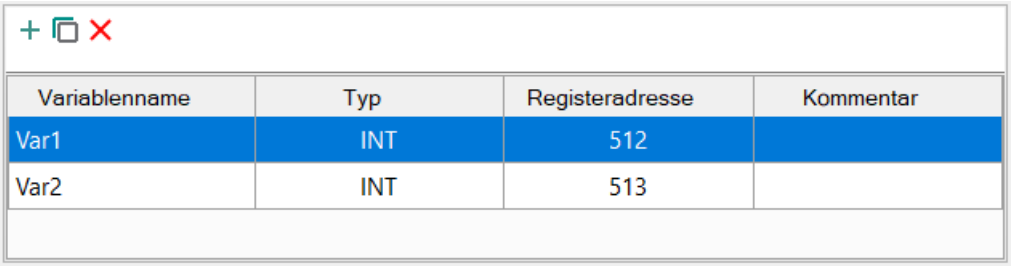
– **Wichtigstes Wort zuerst**: Nein
– **Höchstwertigstes Byte zuerst**: Ja

Erlaubt im Slave-Modus während einer Gruppenabfrage das Lesen von bis zu 64 Registern.

Gerätevariablen

Die Gerätevariablentabelle, die sich am unteren Rand des Fensters befindet, enthält die Variablen, deren Werte von einem anderen Gerät gelesen werden.

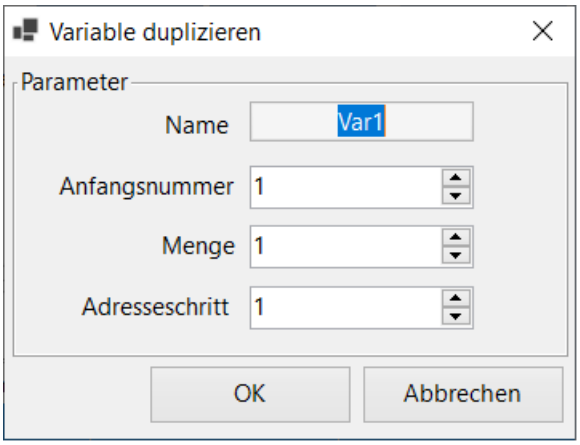
Fügen Sie eine Variable hinzu, indem Sie auf das Symbol + klicken. Die Variable wird in die Tabelle eingefügt, wie in der folgenden Abbildung gezeigt.



Variablenname	Typ	Registeradresse	Kommentar
Var1	INT	512	
Var2	INT	513	

Um eine Variable aus der Tabelle zu löschen, wählen Sie sie aus und klicken Sie auf die Schaltfläche ✕ .
Um mehrere Variablen mit denselben Einstellungen zu erstellen, wählen Sie eine Variable aus und klicken Sie auf das Symbol ***Duplizieren***.

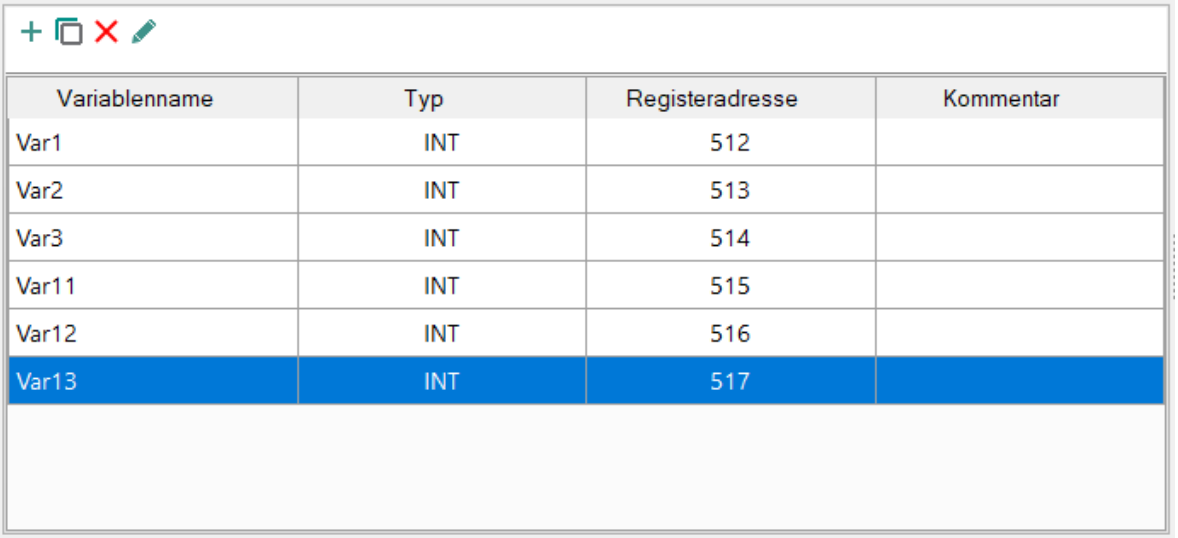
HINWEIS
Das Duplizieren von Variablen ist nur für Geräte der ersten Generation möglich.



Einstellungen für **Variable duplizieren**:

– ***Name*** – Gibt den Namen der duplizierten Variable an.
– ***Anfangsnummer*** – Legt die erste Nummer fest, die dem Variablennamen hinzugefügt wird.
– ***Menge*** – Definiert die Anzahl der zu duplizierenden Variablen.
– ***Adresseschritt*** – Legt das Inkrement der Registeradresse fest.

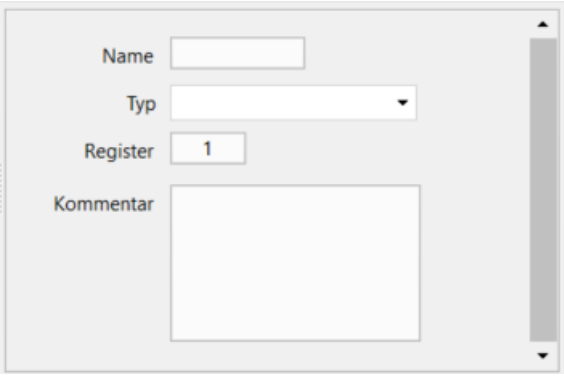
Die Variablen werden der Tabelle hinzugefügt.



Variablenname	Typ	Registeradresse	Kommentar
Var1	INT	512	
Var2	INT	513	
Var3	INT	514	
Var11	INT	515	
Var12	INT	516	
Var13	INT	517	

Die duplizierten Variablen, die mit Postfix _R oder _W in die LoRa-Schnittstelle eingefügt wurden, haben eine Startnummer vor dem Postfix.

Eigenschaften der Gerätevariablen



Die Eigenschaften der Gerätevariablen werden rechts neben der Tabelle konfiguriert:

– **Name** – Gibt den Namen der Gerätevariablen an (vom Benutzer festgelegt).
– **Typ** – Definiert den *Typ* der Gerätevariable (Integer oder Floating Point).

HINWEIS
Boolesche Variablen können in eine Integer-Variable eingelesen werden. Der Zustand eines diskreten Eingangs kann mit dem *EXTRACT*-Block oder einem geeigneten Makro aus dem *Komponenten-Manager* ausgewertet werden.

– **Register** – Gibt die Registeradresse der Gerätevariable an. Der zulässige Adressebereich ist in der Bedienungsanleitung des Geräts definiert.
– **Kommentar** – Definiert eine textliche Beschreibung des Variablenwerts zur Anzeige in der *Variablentabelle*.

Slave-Modus für Geräte der zweiten Generation

Geräte der zweiten Generation arbeiten standardmäßig im Slave-Modus. Die Gerätevariablen werden über die *Variablentabelle* auf der Registerkarte **Netzwerk, Slave** konfiguriert.

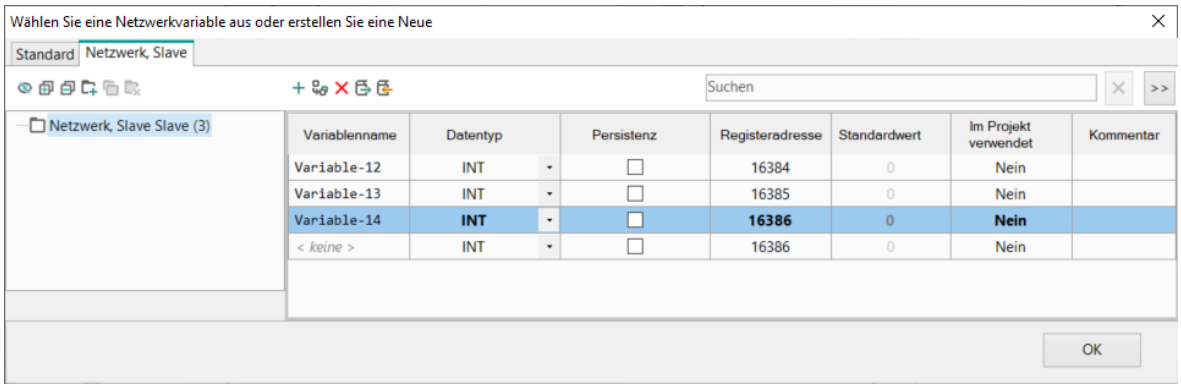


Abb. 5.8 Variablentabelle

– **Variablenname** – Gibt den Namen der Gerätevariablen an (vom Benutzer festgelegt).
– **Datentyp** – Definiert den Datentyp (Integer oder Floating Point).

HINWEIS
Boolesche Variablen können in eine Integer-Variable eingelesen werden. Der Zustand eines diskreten Eingangs kann mit dem *EXTRACT*-Block oder einem geeigneten Makro aus dem *Komponenten-Manager* ausgewertet werden.

– **Persistenz** – Speichert die Variable bei aktivierter Checkbox im nichtflüchtigen EEPROM. Speichert Persistenzinformationen im Projekt.
– **Registeradresse** – Gibt die Adresse der Netzwerkvariablen an (Bereich: 16384–20479).
– **Standardwert** – Legt den Initialwert beim Programmstart fest.
– **Im Projekt verwendet** – Zeigt an, ob die Variable im Programm verwendet wird (Ja/Nein).
– **Kommentar** – Definiert eine textliche Beschreibung zur Anzeige in der Variablentabelle.

Nach dem Verbinden des Geräts mit akYtec ToolPro oder akYtec Cloud werden die Netzwerkvariablen in der Parameterstruktur angezeigt.
Um die Anzeige zu konfigurieren, wählen Sie eine Variable in der Tabelle aus und klicken Sie auf die Schaltfläche >> **Parameter-Einstellungen** in der oberen rechten Ecke des Fensters. Das Panel **Parameter-Konfiguration** wird eingeblendet.

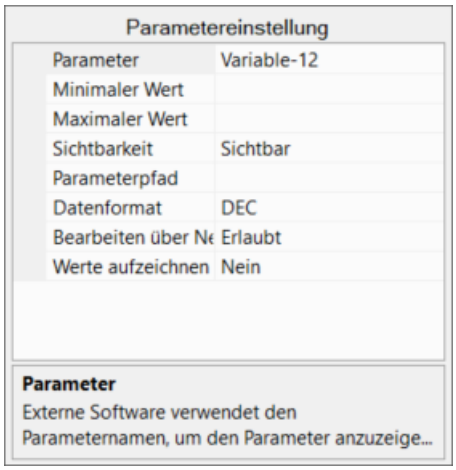


Abb. 5.9 Parametereinstellung

- **Parameter** – Gibt den Parameternamen an, der in der Parameterstruktur in akYtec ToolPro und akYtec Cloud angezeigt wird.
- **Minimaler Wert** – Definiert den minimalen Anzeigewert in akYtec ToolPro und akYtec Cloud. Hat keine Auswirkung auf die Programmlogik.
- **Maximaler Wert** – Definiert den maximalen Anzeigewert in akYtec ToolPro und akYtec Cloud. Hat keine Auswirkung auf die Programmlogik.
- **Sichtbarkeit** – Legt fest, ob der Parameter angezeigt wird (**Sichtbar**) oder nicht (**Versteckt**).
- **Parameterpfad** – Legt den Anzeigeort des Parameters in externer Software fest. Trennt Verzeichnisse durch „\“.
- **Datenformat** – Definiert das Anzeigeformat des Werts (DEC, BIN, HEX, ENUM). Relevant für Integer-Variablen.
- **Bearbeiten über Netzwerk** – Legt fest, ob der Parameter über akYtec ToolPro und akYtec Cloud bearbeitet werden kann.
- **Wert aufzeichnen** – Legt fest, ob der Parameterwert im Gerät archiviert wird (Ja/Nein).

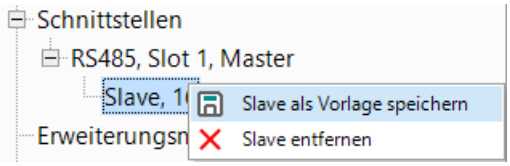
5.3.2.3 Vorlagen

Vorlagen für Netzwerkgeräte

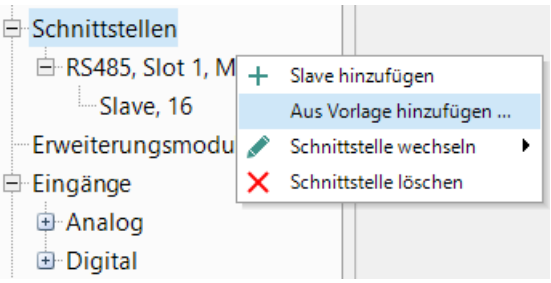
Für Schnittstellen, die im Master-Modus arbeiten, steht die Erstellung und Verwendung von Netzwerkgerätevorlagen zur Verfügung. Die Parameter und Variablen eines konfigurierten Geräts können als Vorlagendatei gespeichert werden.

Erstellen einer Vorlage

Um eine Vorlage zu erstellen, wählen Sie **Slave als Vorlage speichern** im Kontextmenü des Geräts oder im oberen Bereich des Gerätekonfigurationsfensters.



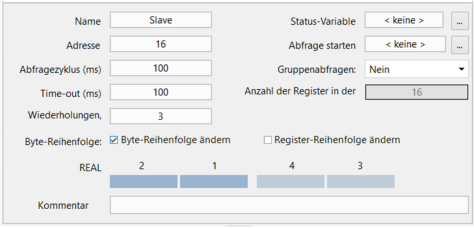
Wählen Sie im geöffneten Datei-Explorer-Fenster den Speicherort aus und geben Sie den Dateinamen ein. Die Datei wird mit der Endung **.dvtpe* gespeichert. Die gespeicherte Vorlage kann für andere Schnittstellen und Projekte verwendet werden.
Um eine gespeicherte Vorlage zu verwenden, klicken Sie mit der rechten Maustaste auf den Schnittstellennamen im Konfigurationsbaum und wählen Sie **Aus Vorlagen hinzufügen…** Wählen Sie im geöffneten Datei-Explorer-Fenster die Datei aus, die die Vorlage enthält.



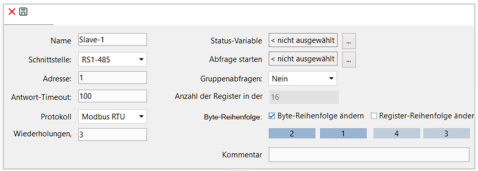
Das Gerät aus der Vorlage wird zusammen mit allen seinen Parametern zur Schnittstelle hinzugefügt.
Für eine Reihe von akYtec-Geräten stehen fertige Vorlagen zur Verfügung, die über den *Komponenten-Manager* aus der Online-Datenbank heruntergeladen werden können.
Bei Verwendung von Netzwerkgerätevorlagen, die in Projekten für *Geräte der ersten Generation* erstellt wurden, in Projekten mit *Geräten der zweiten Generation*:

– Wenn **Gruppenabfrage** ausgewählt ist, behält die Anzahl der Register in der Anforderung entweder ihren Wert (wenn der eingegebene Wert zwischen 2 und 16 liegt), oder wird auf 16 gesetzt, wenn der in der Vorlage gespeicherte Wert größer als 16 ist:

Slave-Parameter für Geräte der ersten Generation

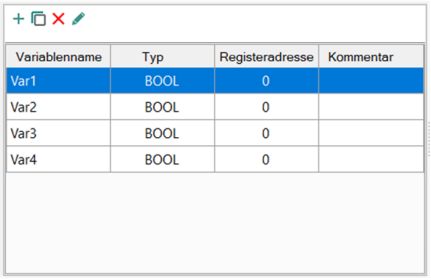


Slave-Parameter für Geräte der zweiten Generation



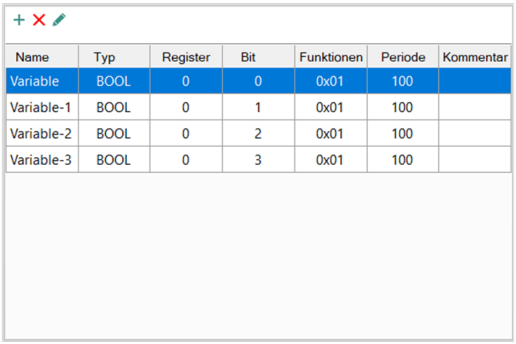
– Eine Variable mit Lese- und Schreibfunktion wird in zwei Variablen aufgeteilt: eine Lese-Funktion _read und eine Schreib-Funktion _write:

Variablentabelle von Slave für Geräte der ersten Generation



Variablenname	Typ	Registeradresse	Kommentar
Var1	BOOL	0	
Var2	BOOL	0	
Var3	BOOL	0	
Var4	BOOL	0	

Variablentabelle von Slave für Geräte der zweiten Generation

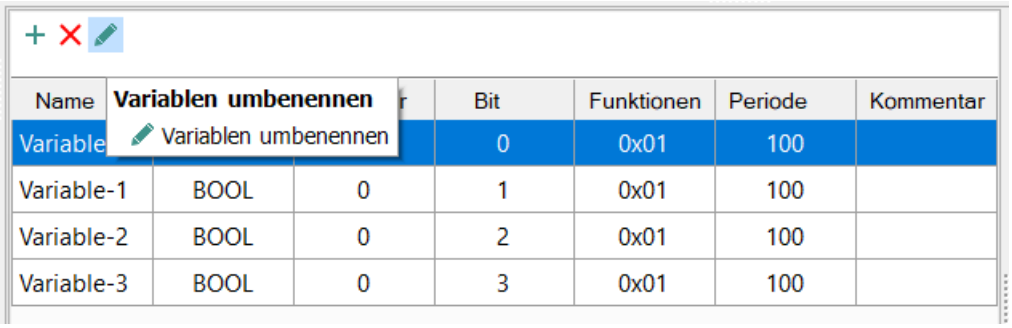


Name	Typ	Register	Bit	Funktionen	Periode	Kommentar
Variable	BOOL	0	0	0x01	100	
Variable-1	BOOL	0	1	0x01	100	
Variable-2	BOOL	0	2	0x01	100	
Variable-3	BOOL	0	3	0x01	100	

Bei Verwendung von Netzwerkgerätevorlagen, die in Projekten für Geräte der zweiten Generation erstellt wurden, in Projekten mit Geräten der ersten Generation, wird der Parameter Abfrageintervall auf seinen Standardwert (100 ms) zurückgesetzt.

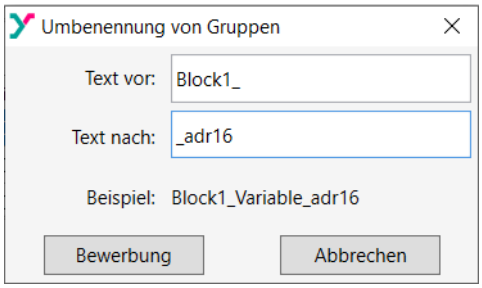
Gruppenweise Umbenennung von Variablen

Um Variablen in der Variablentabelle des Slave-Geräts im Gerätekonfigurationsfenster gruppenweise umzubenennen, klicken Sie auf die Schaltfläche .

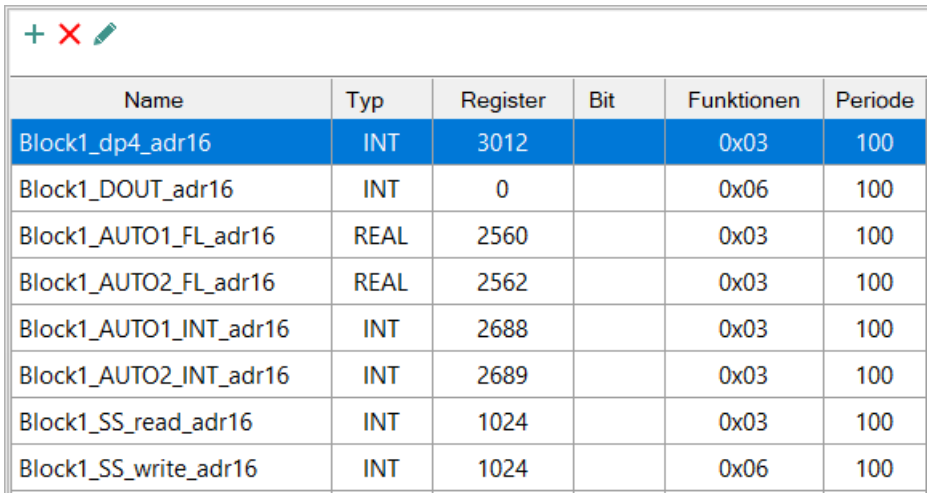


Name			Bit	Funktionen	Periode	Kommentar
Variable			0	0x01	100	
Variable-1	BOOL	0	1	0x01	100	
Variable-2	BOOL	0	2	0x01	100	
Variable-3	BOOL	0	3	0x01	100	

Geben Sie im daraufhin erscheinenden Fenster den Text ein, der vor und nach dem Variablennamen angezeigt werden soll, und klicken Sie auf die Schaltfläche **Übernehmen**.



Daraufhin werden alle Variablen gemäß den eingegebenen Daten umbenannt:

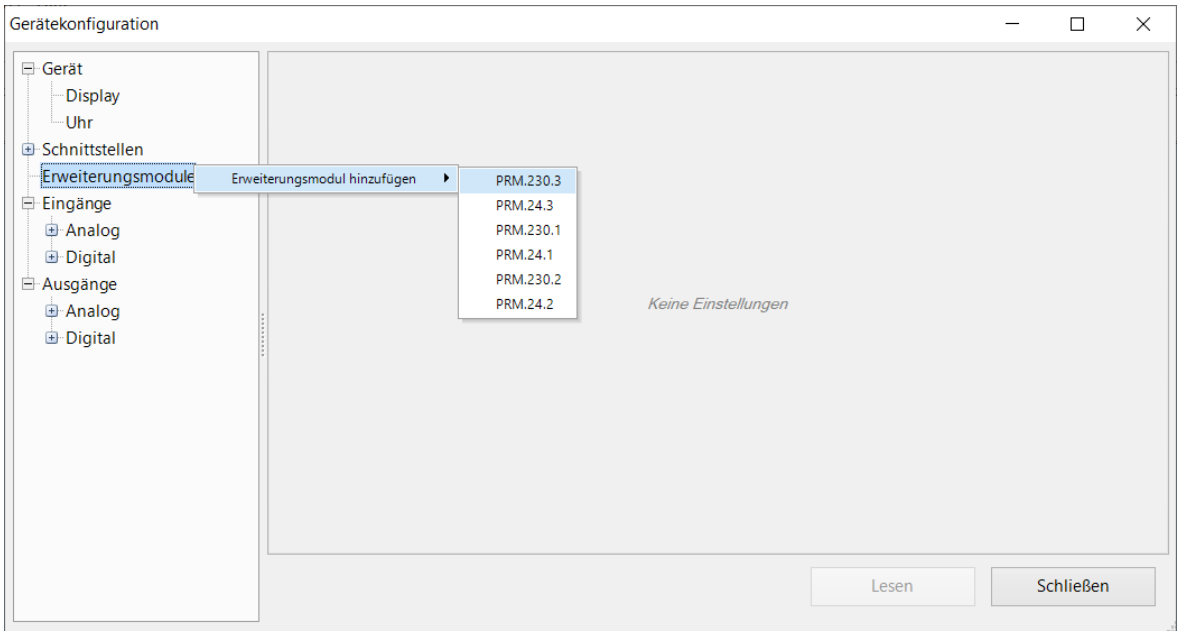


Name	Typ	Register	Bit	Funktionen	Periode
Block1_dp4_adr16	INT	3012		0x03	100
Block1_DOUT_adr16	INT	0		0x06	100
Block1_AUTO1_FL_adr16	REAL	2560		0x03	100
Block1_AUTO2_FL_adr16	REAL	2562		0x03	100
Block1_AUTO1_INT_adr16	INT	2688		0x03	100
Block1_AUTO2_INT_adr16	INT	2689		0x03	100
Block1_SS_read_adr16	INT	1024		0x03	100
Block1_SS_write_adr16	INT	1024		0x06	100

Beim Replizieren einer Variable, die mithilfe der gruppenweise Umbenennung umbenannt wurde, ändert sich der "Text nachher" nicht, an den Namen der replizierten Variable werden die Anfangs- und Folgenummern angehängt.
Die Einstellungen für die gruppenweise Umbenennung werden in der Vorlage gespeichert und stehen bei der späteren Verwendung der Vorlage zur Bearbeitung zur Verfügung.
Die individuelle Umbenennung jeder einzelnen Variable ist in der *Variablentabelle* und im *Programmcode* möglich.

5.4 Erweiterungsmodule

An das Basisgerät können bis zu zwei E/A-Erweiterungsmodule vom Typ PRM angeschlossen werden. Weitere Informationen zu Erweiterungsmodulen finden Sie im PRM-Benutzerhandbuch.
Um die E/A-Punkte eines Moduls im Schaltprogramm zu verwenden, fügen Sie das Modul über dessen Kontextmenü der Gruppe ***Erweiterungsmodule*** hinzu.

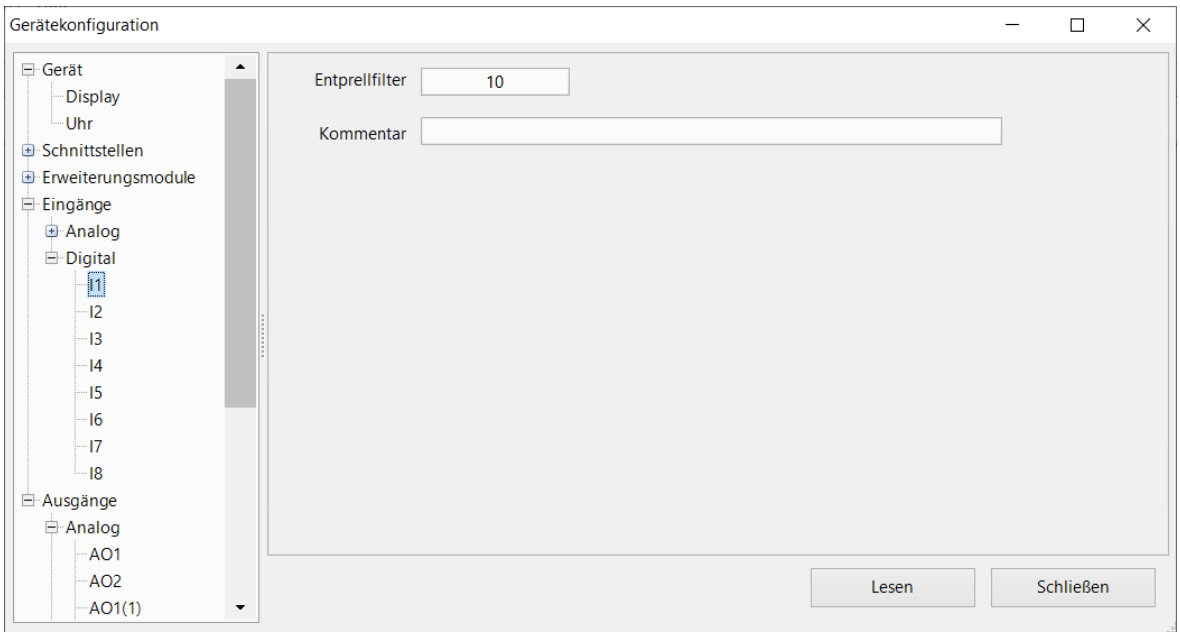


Die zusätzlichen E/A-Punkte der hinzugefügten Module können in den Zweigen ***Eingänge*** bzw. ***Ausgänge*** konfiguriert werden. Sie werden im Baum als ***Ix(y)*** bzw. ***Qx(y)*** angezeigt, wobei ***x*** die Ordnungsnummer des E/A-Punkts auf dem Modul und ***y*** die Ordnungsnummer des Moduls ist, gezählt vom Basisgerät aus.
Vor dem Hochladen des Projekts auf das Basisgerät müssen alle Module über den internen Bus mit dem Basisgerät verbunden und eingeschaltet sein. Beim Hochladen eines Projekts wird die Modul-Firmware mit der aktuellen Version von ALP synchronisiert.

5.5 Eingaenge und Ausgaenge

Eingänge

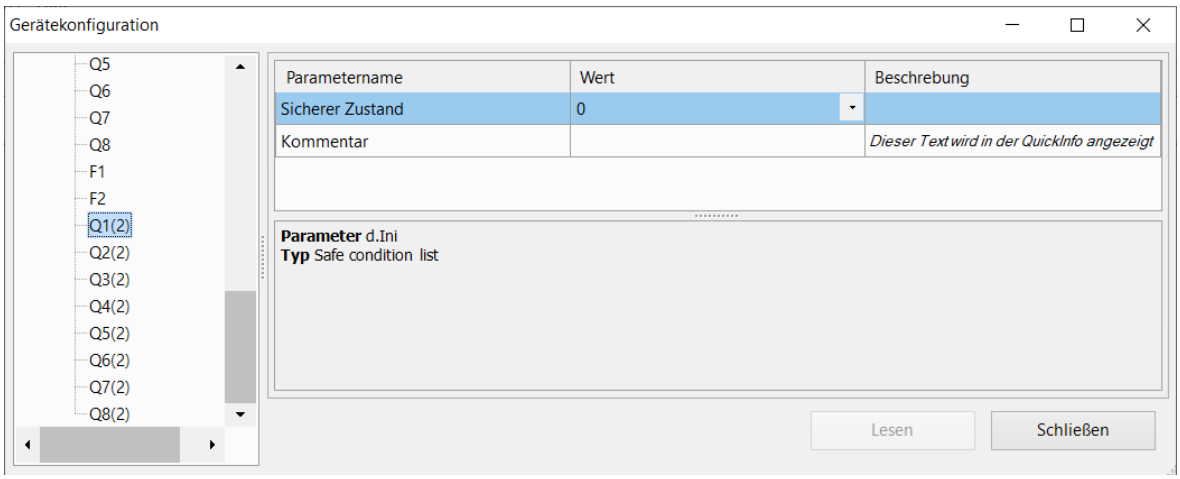
Der Inhalt des Zweiges ***Eingänge*** hängt von den Ressourcen des Zielgerätes ab. Es können analoge und/oder digitale Eingänge sein.
Der Parameter ***Kommentar*** ist für alle Eingangstypen gleich. Der Text in diesem Feld wird in einem Tooltip angezeigt, wenn sich der Mauszeiger im Arbeitsbereich über dem Eingang befindet. Der Text kann auch in das Eigenschaftenfeld eingegeben werden.
Weitere Einzelheiten zur Konfiguration der Eingänge finden Sie im Benutzerhandbuch des Geräts.



Andere Eingangsparameter hängen von der Art des Eingangs und des Geräts ab.

Ausgänge

Der Inhalt des Zweiges ***Ausgänge*** hängt von den Ressourcen des Zielgerätes ab. Es können analoge und/oder digitale Ausgänge sein.
Der Parameter ***Kommentar*** ist für alle Ausgangstypen gleich. Der Text in diesem Feld wird in einem Tooltip angezeigt, wenn sich der Mauszeiger im Arbeitsbereich über dem Ausgang befindet. Er kann auch in das Eigenschaftenfeld eingegeben werden.
Weitere Einzelheiten zur Konfiguration der Ausgänge finden Sie im Benutzerhandbuch des Geräts.
Die digitalen Ausgänge des Erweiterungsmoduls verfügen über einen zusätzlichen Parameter ***Sicherer Zustand***. Der Parameter gibt den Ausgangszustand an, falls die Verbindung zwischen Modul und Basisgerät verloren geht.



Einstellungen für Geräte auf der neuen Hardwareplattform

Das Fenster zum Einrichten der Ein- und Ausgänge für Geräte auf der neuen Plattform hat eine andere Oberfläche und die Parameter auf der rechten Seite des Fensters werden in einer Tabelle dargestellt.
Bei Geräten der neuen Plattform befindet sich das Menü zum Einstellen des sicheren Zustands der Ausgänge im Zweig des hinzugefügten Erweiterungsmoduls.

5.6 Passwort

Für Geräte auf der neuen Plattform können Sie ein Passwort festlegen, um das Gerät zu schützen. Das Passwort wird im Bereich **Passwort** im Einstellungsbaum nur für das mit dem PC verbundene Gerät festgelegt.

Passwort erstellen

Wenn im Gerät kein Passwort festgelegt ist, ist die Passworterstellung im Einstellungsfenster aktiv. Zur Installation müssen Sie Ihr Passwort eingeben und bestätigen.

Passwort ändern und zurücksetzen

Wenn das Gerät bereits ein Passwort festgelegt hat, können Sie es ändern oder zurücksetzen.
Um das Passwort zu ändern, geben Sie das aktuelle Passwort und das neue Passwort in die Spalten **Passwort ändern** ein.
Um Ihr Passwort zurückzusetzen, geben Sie Ihr aktuelles Passwort in die Spalte **Passwort zurücksetzen** ein.
Wenn Sie Ihr Passwort vergessen haben, lesen Sie im Benutzerhandbuch des Geräts nach, wie Sie es zurücksetzen können.
Bei einem passwortgeschützten Gerät ist bei der Aufnahme einer Sendung die Eingabe eines Passworts erforderlich, siehe *Projekt auf Gerät hochladen*.

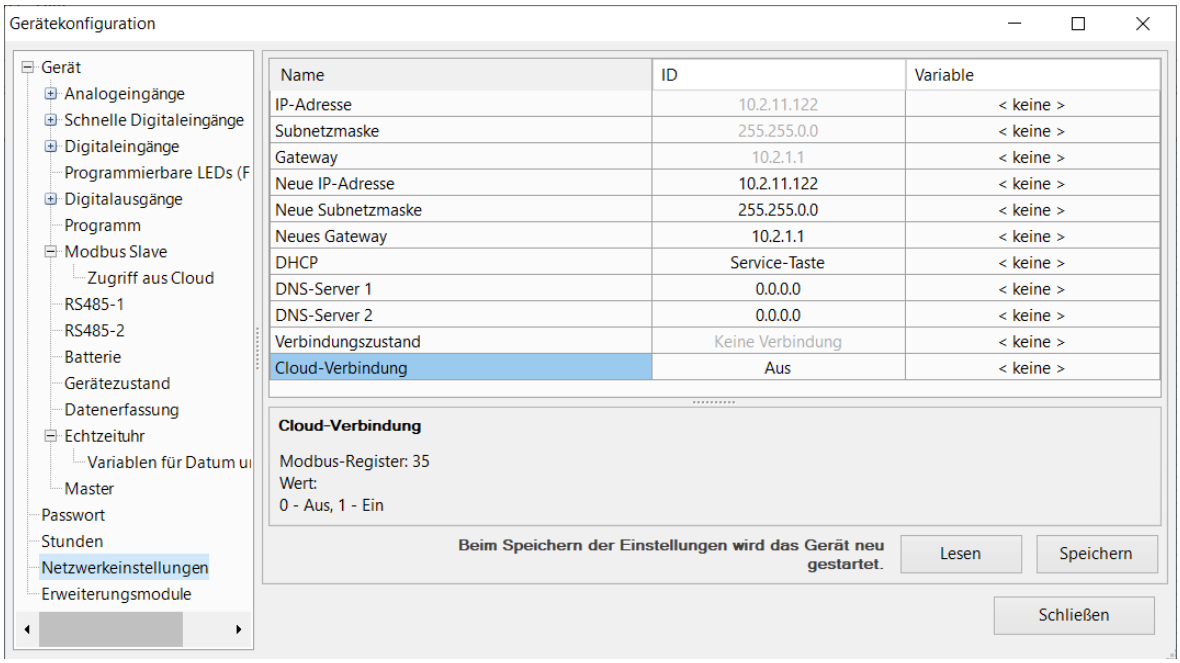
5.7 Verbindung zur akYtec Cloud

Für Geräte auf der neuen Plattform ist eine akYtec Cloud-Konnektivität verfügbar (z. B. PR103).

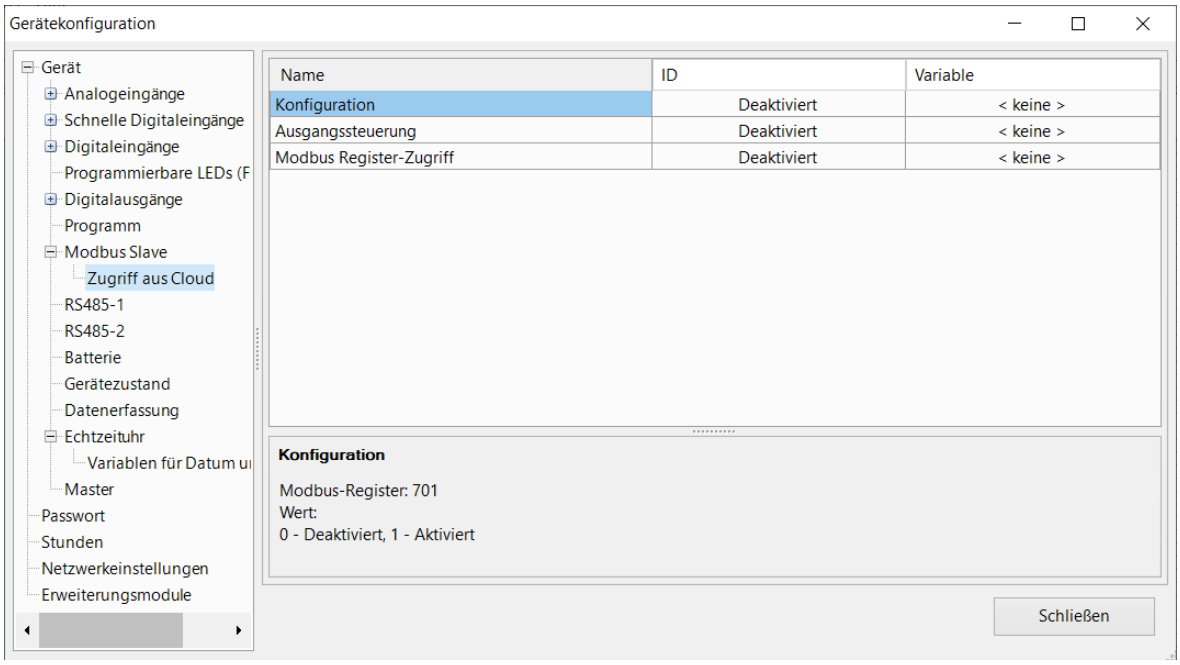
HINWEIS
Eine Liste der Geräte auf der neuen Plattform finden Sie unter *Einführung*.

HINWEIS
Um eine Verbindung zur akYtec Cloud herzustellen, muss das Gerät passwortgeschützt sein. Ohne Passwort kann keine Verbindung zur akYtec Cloud hergestellt werden. Informationen zum Festlegen des Passworts finden Sie unter *Passwort*.

Bevor Sie die Verbindung zur akYtec Cloud herstellen, sollten Sie die Autorisierung des Geräts in der akYtec Cloud einrichten. Weitere Informationen finden Sie in der *akYtec Cloud-Bedinungsanleitung* und in der *Bedinungsanleitung* des Geräts. Während des Betriebs muss das Gerät für die Datenübertragung mit dem Internet verbunden sein.
Im Abschnitt **Netzwerkeinstellungen** des Einstellungsbaums können Sie die Verbindung und den Verbindungsstatus in zur akYtec Cloud einstellen.



Im Abschnitt **Zugriff aus Cloud** des Konfigurationsbaums können Sie die akYtec Cloud-Zugriffsebenen für das Gerät festlegen.



Ein Beispiel für die Arbeit mit der akYtec Cloud finden Sie im *Abschnitt 11.3*.

6 Variablen

Variablen werden verwendet, um Werte im Projektdiagramm zu schreiben und zu lesen, und bei der *Display-Programmierung*.

In Verwendung befindliche Variablen

Eine Variable gilt als **in Verwendung**, wenn sie **mindestens eine** der folgenden Bedingungen erfüllt:

- Allgemeine Bedingungen für alle Variablentypen:
  - Es ist mit einem Visualisierungselement verknüpft.
  - Es ist mit einem Geräteparameter verknüpft.
  - Es ist mit physischen Tasten verbunden.
  - Es wird für logische Übergänge zwischen Bildschirmen verwendet.
  - Für die Variable im Diagramm existiert ein Ausgabeblock für eine nichtflüchtige Variable (Nichtflüchtigkeit = Ein).
- Zusätzliche Bedingungen für Variablen vom Typ Array:
  - Für ein Array mit **Booleschen Elementen** und auf „Aus“ gesetzter Nichtflüchtigkeit – existiert ein Ausgabeblock für das Array oder seine Elemente im Diagramm.
  - Für ein Array mit **Integer- oder Real-Elementen** und Nichtflüchtigkeit auf "Aus" gesetzt – existiert ein Eingabeblock für das Array oder seine Elemente im Diagramm.

Wenn keine der Bedingungen erfüllt ist, wird die Variable als **unbenutzt** betrachtet.
Nicht verwendete Variablen:

- Nehmen nicht an der Programmlogik teil.
- Sind nicht an der Programmsimulation beteiligt.
- Belegen den Gerätespeicher nicht.

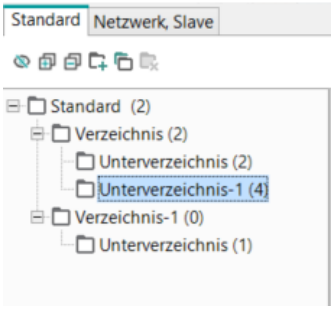
Variablentabelle

Um eine Variable in einem Projekt zu verwenden, müssen Sie sie zunächst in der Variablentabelle erstellen.
Um alle Projektvariablen anzuzeigen, klicken Sie in der Symbolleiste auf das Symbol oder verwenden Sie den Menüpunkt ***Gerät*** → ***Variablentabelle***.
Die Variablen sind in drei Gruppen unterteilt, denen in der Tabelle jeweils ein eigener Reiter zugeordnet ist:

- **Standard**
- *Service*
- *Netzwerk*

Verzeichnisbaum der Variablen

Die **Variablenverzeichnisstruktur** ermöglicht das Erstellen von Katalogabschnitten zur Organisation von Variablen. Für jede Variablenkategorie wird eine separate Katalogstruktur angezeigt. Für Standard- und Netzwerkvariablen können Sie zusätzliche Katalogabschnitte erstellen (maximale Verschachtelungstiefe 3).
Die Kataloge für Servicevariablen sind fest und können nicht geändert werden. Neben jedem Katalognamen wird die Anzahl der Variablen in diesem Katalog in Klammern angezeigt. Diese Anzahl schließt Variablen in verschachtelten Katalogen ein oder aus, abhängig von der Einstellung „Verschachtelte Variablen anzeigen“.



	Anzeige ein-/ausblenden für geschachtelte Variablen
	Unterverzeichnisse aufklappen
	Unterverzeichnisse reduzieren
	Neues Unterverzeichnis erstellen. Das Erstellen eines Bereichs ist auch über das Kontextmenü des Katalogs verfügbar.
	Verzeichnis duplizieren – der ausgewählte Ordner wird im übergeordneten Katalog dupliziert, einschließlich aller eingebetteten Bereiche und hinzugefügten Variablen. Das Kopieren eines Bereichs ist auch über das Kontextmenü des Katalogs verfügbar. Wählen Sie **Kopieren**, dann wählen Sie den Katalog aus, in dem Sie den kopierten Bereich einfügen möchten, klicken Sie mit der rechten Maustaste und wählen Sie **Einfügen** Der Zielkatalog zum Hinzufügen der Informationen muss sich eine Ebene über dem kopierten Bereich befinden.
	Katalog löschen. Das Löschen eines Katalogbereichs ist auch über das Kontextmenü des Katalogs verfügbar. **HINWEIS** Ein Katalogbereich, der Variablen enthält, kann nicht gelöscht werden.

Um einen Abschnitt des Variablenverzeichnisses zu kopieren, klicken Sie mit der rechten Maustaste auf den zu kopierenden Verzeichnisabschnitt und wählen Sie **Kopieren**. Wählen Sie anschließend das Verzeichnis aus, dem Sie den kopierten Abschnitt hinzufügen möchten, klicken Sie mit der rechten Maustaste und wählen Sie **Einfügen**.

HINWEIS
Das Verzeichnis, zu dem die Informationen hinzugefügt werden sollen, muss eine Ebene über dem Verzeichnis liegen, das kopiert wird.

Um einen Abschnitt des Variablenverzeichnisses zu löschen, klicken Sie mit der rechten Maustaste auf den zu löschenden Abschnitt und wählen Sie **Löschen**.

HINWEIS
Der Verzeichnisabschnitt mit den Variablen kann nicht gelöscht werden.

Um Verzeichnisabschnitte zu verschieben, halten Sie die Umschalttaste gedrückt und verschieben Sie den Abschnitt mit der Maus an eine neue Position.

HINWEIS
Das Verschieben von Abschnitten ist nur innerhalb des übergeordneten Verzeichnisses möglich.

Um einen Verzeichnisabschnitt umzubenennen, geben Sie in der Zeile **Name** unten im **Variablenverzeichnisbaum** einen neuen Namen ein.

HINWEIS
Das Feld für den Verzeichnisabschnittsnamen darf nicht leer sein oder den Namen eines anderen Abschnitts im übergeordneten Verzeichnis duplizieren. Bei Eingabe eines falschen Namens wird eine Fehlermeldung angezeigt.

Um den Verzeichnisabschnitt einer Variable zu ermitteln, markieren Sie die Variable in der Tabelle und drücken Sie ***Strg + H***. Der Abschnitt, in dem sich die Variable befindet, wird im Verzeichnisbaum blau hervorgehoben, der Name des Abschnitts wird in der Zeile darunter angezeigt und der Abschnitt, in dem sich die Variable befindet, wird in der Variablentabelle geöffnet. Sie können auch

mit der Maus über eine Variable fahren, um ein Popup-Fenster mit dem Namen und dem Speicherort der Variable zu öffnen.

Variablen

Die Projektvariablen werden in der Variablentabelle angezeigt.

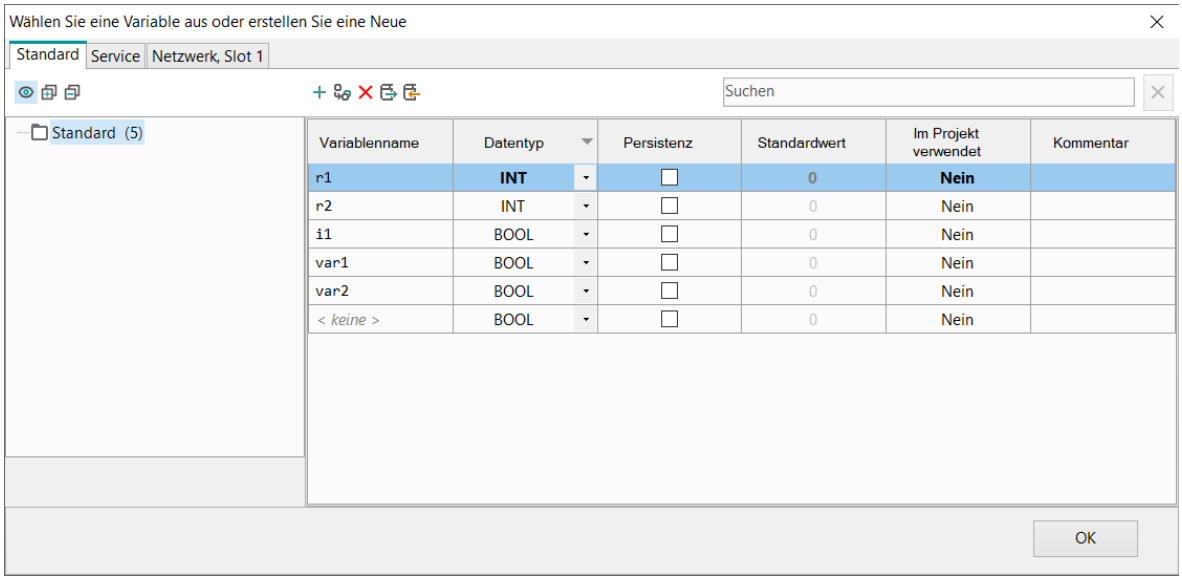


Abb. 6.1 Fenster zur Variablenauswahl und -erstellung

+	Erstellt eine Variable.
	Dupliziert eine Variable.
✕	Löscht eine Variable.
	Exportiert Standardvariablen in eine CSV-Datei.
	Importiert Standardvariablen aus einer CSV-Datei.
	Erweitert alle Arrays.
	Blendet alle Arrays aus.

Um eine Variable zu erstellen, geben Sie einen Namen ein und legen Sie den *Typ 6.1* fest oder klicken Sie oben links in der Tabelle auf die Schaltfläche +. Die weiteren Parameter werden entsprechend den betrieblichen Anforderungen ausgefüllt:

– **Variablenname** – Gibt den Namen an, der im Variablenbereich und im Variablenblock im Projekt angezeigt wird.
– **Datentyp** – Definiert den *Variablentypen* (BOOL, INT oder REAL).

  Wenn Sie eine Variable vom Typ Array auswählen, geben Sie Folgendes im daraufhin angezeigten Fenster **Array-Einstellungen** ein:

  – **Größe des Arrays** – Legt die Anzahl der Elemente im Array fest (max. 256). Gibt Werte im Format [x..y] an, wobei x der Index des ersten und y der Index des letzten Elements ist.

  – **Elementtyp** – Definiert den Datentyp der Array-Elemente (BOOL, INT oder REAL).

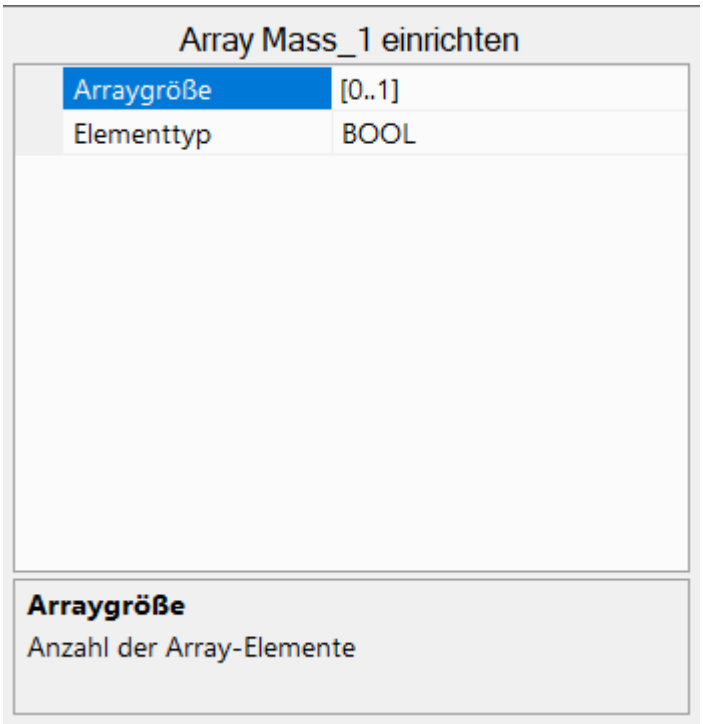


Abb. 6.2 Array-Einstellungen

– **Persistenz** – Speichert die Variable im nichtflüchtigen Gerätespeicher und macht sie zur Retain-Variable (nur für Standardvariablen verfügbar).

> **HINWEIS**
> Wenn bei Geräten auf der neuen Plattform das Kontrollkästchen **Persistenz** in der Variablentabelle aktiviert ist, wird der Variablenwert wie folgt platziert:
>
> – Zum System-EEPROM für eine Standardvariable, die an einen Geräteparameter oder ein Visualisierungselement gebunden ist. Für die Netzwerk-Slave-Variable
>
> – Zum EEPROM für eine Standardvariable, die nicht an den Geräteparameter oder das Visualisierungselement gebunden ist
>
> Wenn Sie ein Projekt speichern, werden die Informationen zu den nichtflüchtigen Variablen gespeichert.

– **Standardwert** – Wert, den die Variable behält, bis ein neuer Wert geschrieben wird.
– **Im Projekt verwendet** – Angabe der Bindung an Blöcke im Programm. Wenn die Variable gebunden ist, lautet der Wert **Ja**.
– **Kommentar** – Textbeschreibung, die im Tooltip des Projektdiagramms angezeigt wird, wenn der Cursor über den Variablenblock bewegt wird.

Um eine Variable zu duplizieren, klicken Sie mit der rechten Maustaste auf die Variablenzeile und wählen Sie im Kontextmenü die Option **Variable duplizieren** oder klicken Sie oben in der Tabelle auf die Schaltfläche .

> **HINWEIS**
> Das Duplizieren eines Array-Elements ist nicht möglich.

Um eine Variable aus der Tabelle zu entfernen, klicken Sie mit der rechten Maustaste auf die Variablenzeile und wählen Sie im Kontextmenü **Variable löschen** oder klicken Sie oben in der Tabelle auf die Schaltfläche .

> **HINWEIS**
> Das Löschen eines Array-Elements ist nicht möglich.

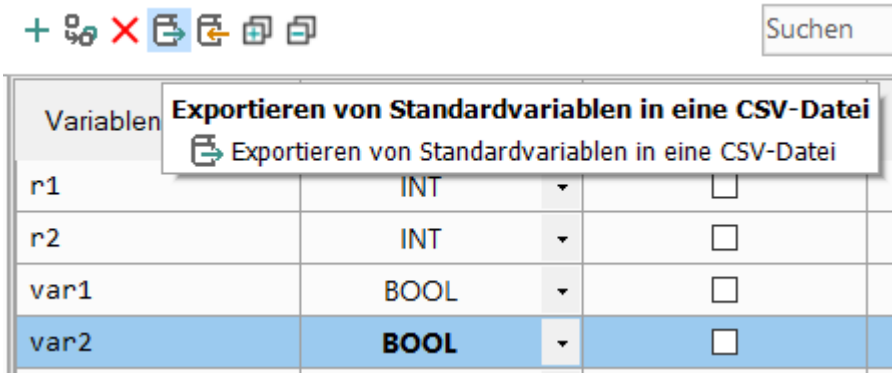
Variablen in eine Datei exportieren

Exportieren Sie Variablen als Tabelle im CSV-Format:

1. Klicken Sie oben in der Tabelle auf **Variablen in CSV-Datei exportieren**.
2. Wählen Sie im geöffneten Fenster den Speicherort der Datei.
3. Klicken Sie auf die Schaltfläche **Speichern**.

HINWEIS
Der Dateiname wird anhand des exportierten Tabs nach folgendem Muster generiert: ***Projektname_Tab_Variablen***.

HINWEIS
Bei Geräten der zweiten Generation wird die Registerkarte „Slave-Netzwerkvariablen“ zusammen mit den Einstellungen für die Variablenparameter exportiert.



Variablen aus einer Datei importieren

Um eine Variablentabelle aus einer ***.csv***-Datei zu importieren, klicken Sie auf die Schaltfläche **Variablen importieren aus CSV-Datei** oben in der Tabelle. Wählen Sie die Datei aus, aus der Sie Daten importieren möchten, und klicken Sie auf **Öffnen**. Alle in der Datei vorhandenen Kataloge werden ebenfalls importiert, wobei die Verschachtelung der Variablen erhalten bleibt. Wenn die Datei fehlerhafte Daten enthält, z. B. eine fehlende Spalte für den Variablennamen, den Typ oder das Register, wird eine Fehlermeldung angezeigt. Nachdem Sie den Fehler behoben haben, führen Sie den Import erneut durch. Wenn es Übereinstimmungen in den Variablennamen und/oder Registern zwischen dem Projekt und der importierten Datei gibt, wird ein Informationsfenster angezeigt, in dem alle Namens- und/oder Registerkonflikte aufgelistet sind. Sie werden dann aufgefordert, den Import fortzusetzen oder abzubrechen.

Arrays erweitern/reduzieren

Wenn Ihr Projekt eine Variable vom Typ Array enthält, können Sie die Anzeige ihrer Elemente in der Variablentabelle konfigurieren. Ein Array wird in der Variablentabelle als Arraybaum dargestellt. Um den Arraybaum für alle Array-Variablen in der Variablentabelle zu erweitern, klicken Sie auf die Schaltfläche **Alle Arrays erweitern** oben in der Tabelle.
Diese Einstellung bleibt erhalten, wenn Sie die Variablentabelle in derselben Sitzung das nächste Mal öffnen. Um die Elemente einer einzelnen Array-Variable anzuzeigen, klicken Sie auf die Schaltfläche „+“ vor dem Namen des Arrays. Um den Arraybaum für alle Array-Variablen in der Variablentabelle zu reduzieren, klicken Sie auf die Schaltfläche **Alle Arrays reduzieren** oben in der Tabelle. Diese Einstellung bleibt beim nächsten Öffnen der Variablentabelle in derselben Sitzung erhalten. Um die Elemente einer einzelnen Array-Variablen auszublenden, klicken Sie auf die Schaltfläche „-“ vor dem Namen des Arrays.

Variablen suchen

Die Variablentabelle unterstützt die Suche und Filterung nach Namen. Das Leerzeichen fungiert als logisches ODER und ermöglicht das Filtern von Variablen nach mehreren Kriterien.

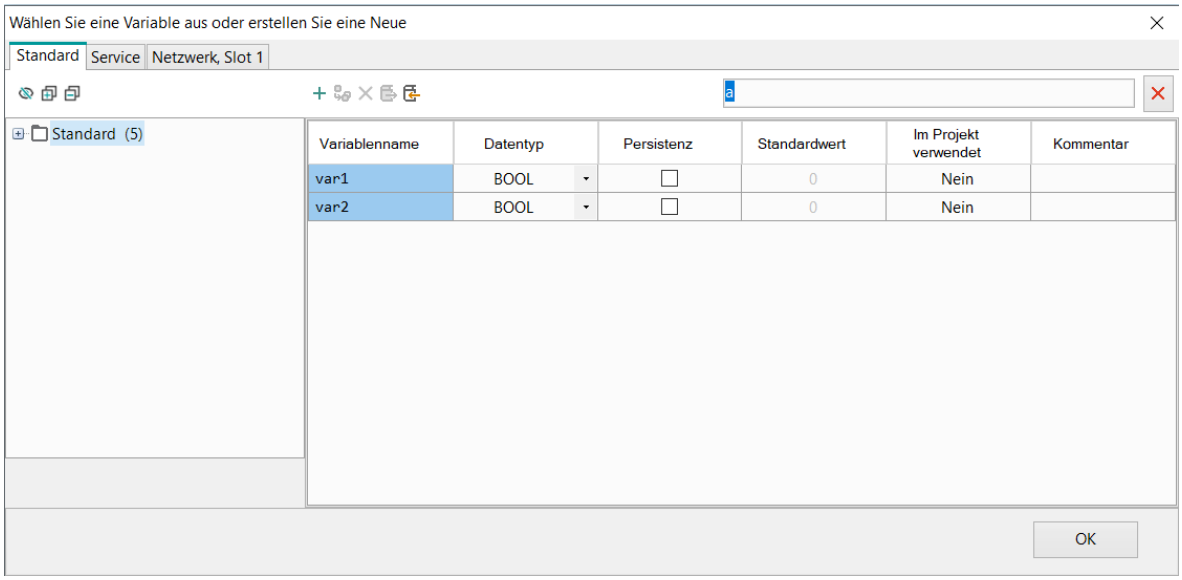


Abb. 6.3 Suche nach Variablen

Sollte die in der Suchzeile eingegebene Variable nicht vorhanden sein, können Sie sie durch Klicken auf die Schaltfläche + oben in der Tabelle erstellen. Der Variable wird automatisch der im Suchfeld eingegebene Name zugewiesen.

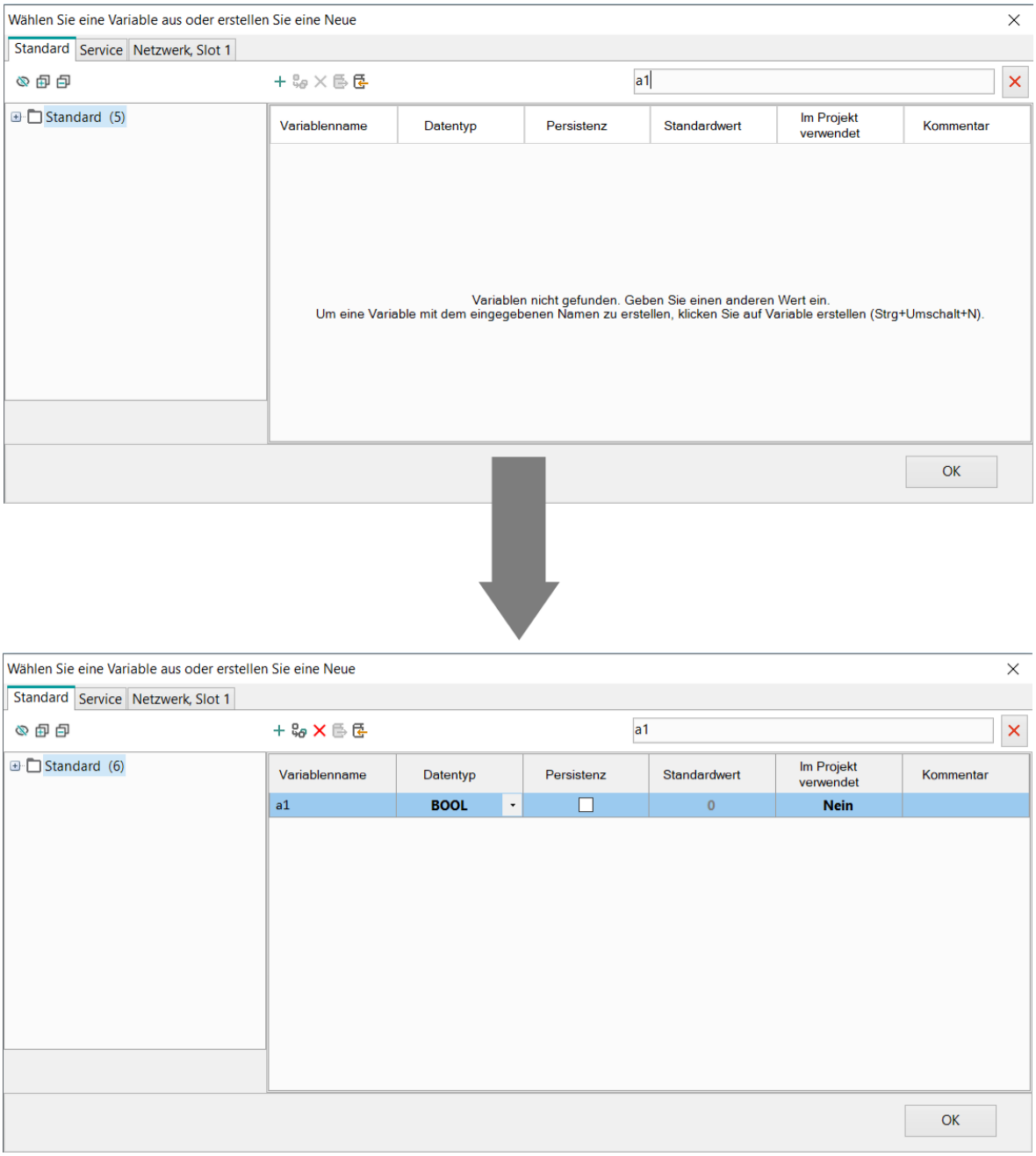


Abb. 6.4 Erstellen einer Variable

Kontextmenü der Variablentabelle

Die Variablentabelle ermöglicht die Sortierung nach Wert. Klicken Sie dazu mit der rechten Maustaste auf den Spaltennamen und wählen Sie im Kontextmenü die gewünschte Sortierreihenfolge (aufsteigend oder absteigend).

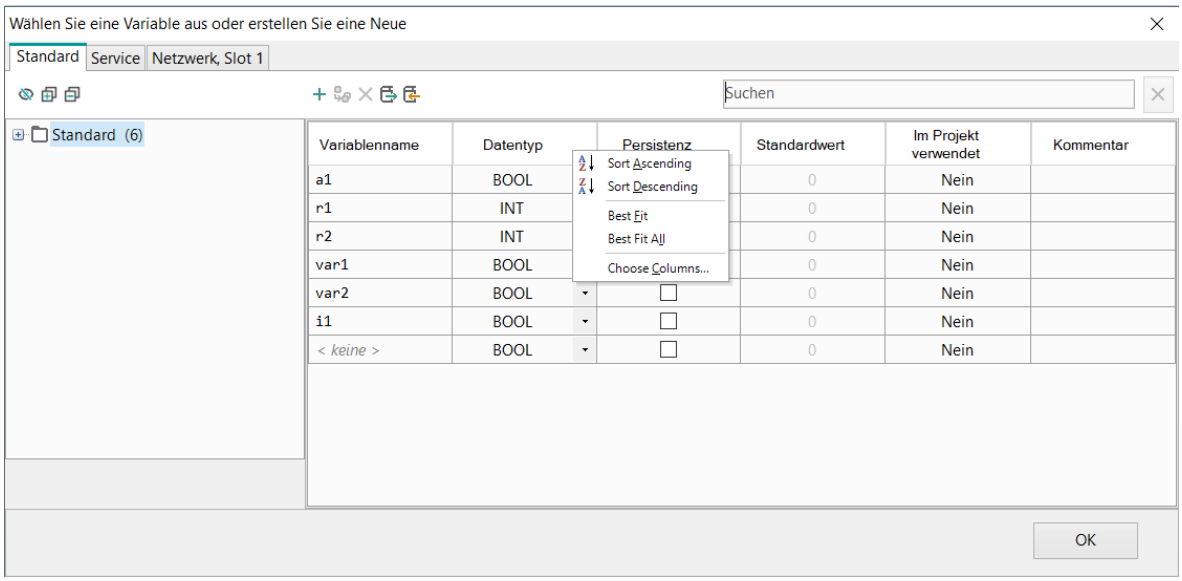
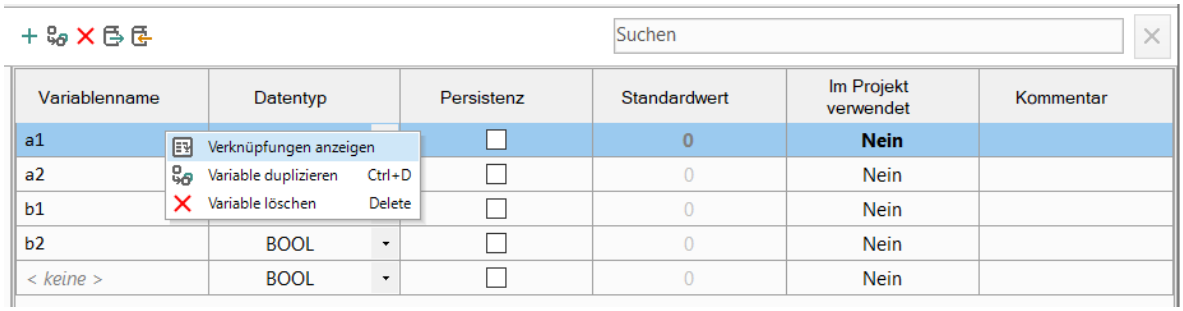
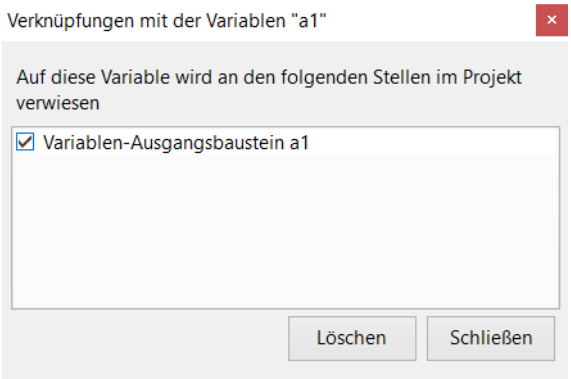


Abb. 6.5 Sortieren von Variablen

Um die Bereiche zu finden, in denen die Variable im Projekt verwendet wird, klicken Sie mit der rechten Maustaste auf die Variablenzeile in der Tabelle und wählen Sie **Verküpfungen anzeigen**.



Das sich öffnende Fenster zeigt Informationen zur ausgewählten Variable an.



Bei Variablen vom Typ Array werden Referenzen sowohl für das Array selbst als auch für seine einzelnen Elemente angezeigt. Um die Bindung eines Variablenblocks im Projekt zu entfernen, markieren Sie die Kontrollkästchen neben den gewünschten Variablen und klicken Sie auf die Schaltfläche **Löschen**. Die Variable bleibt in der Tabelle erhalten, wird aber im Projekt nicht mehr verwendet.

6.1 Datentypen

In einem Programm können Variablen der folgenden Typen verwendet werden:

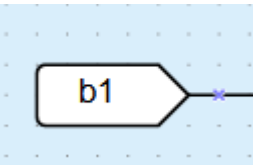
– Boolescher Wert (***BOOL***)

– Ganzzahl (***INT***)
– Gleitkommazahl (***REAL***)
– Feld (***Array***)

HINWEIS
Bei verschiedenen Geräten kann es Einschränkungen hinsichtlich der Unterstützung bestimmter Variablentypen geben.

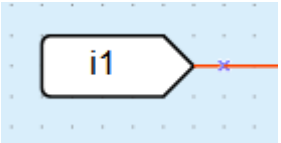
BOOL

Eine Variable dieses Typs (Boolescher Wert (BOOL)) hat nur zwei mögliche Werte, 1 (Wahr) oder 0 (Falsch).
Die Verbindungslinien zwischen den BOOL-Variablen im Schaltprogramm werden schwarz dargestellt.



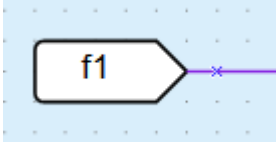
INT

Eine Variable dieses Typs (Ganzzahl (INT)) ist eine vorzeichenlose Ganzzahl im Bereich von 0 bis 4.294.967.295 (4 Byte).
Die Verbindungslinien zwischen den INT-Variablen im Schaltprogramm werden rot dargestellt.



REAL

Eine Variable dieses Typs (Gleitkommazahl (REAL)) hat einen Wert im Bereich von −3,402823e +38–3,402823e+38 und wird durch eine Gleitkommazahl mit einfacher Genauigkeit (4 Byte) dargestellt.
Die Verbindungslinien zwischen den REAL-Variablen im Schaltprogramm werden violett dargestellt.

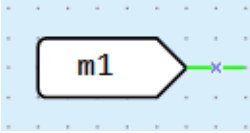


Array

Eine Variable vom Typ Array speichert eine geordnete Menge von Elementen desselben Typs: BOOL, INT oder REAL. Die maximale Anzahl von Elementen in einem einzelnen Array beträgt 256.

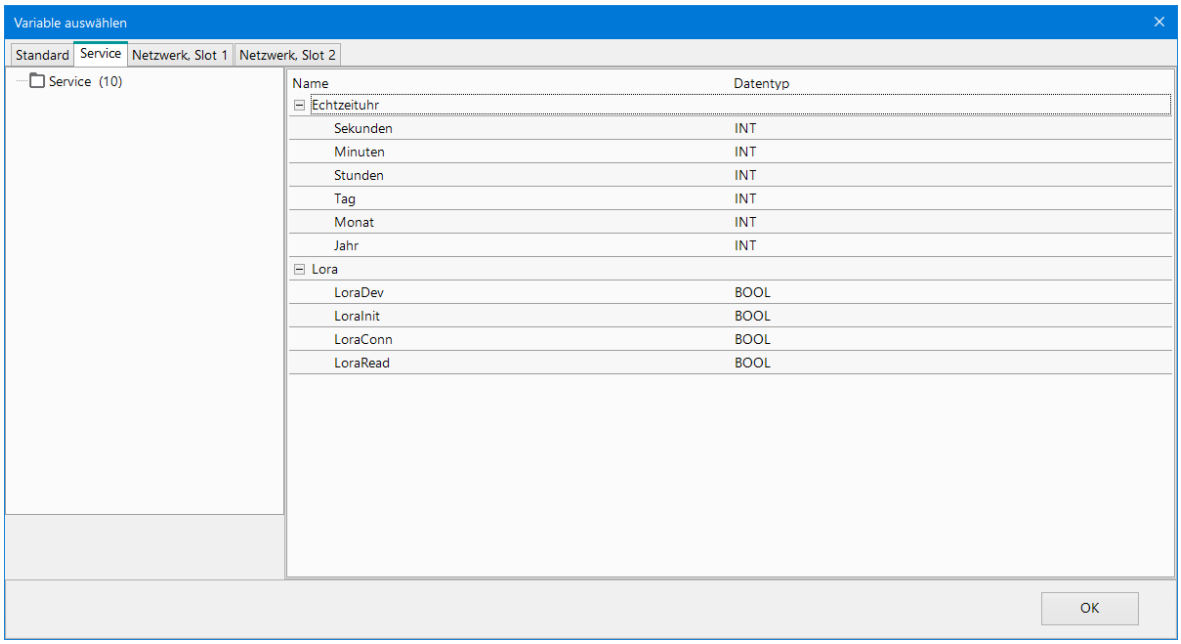
HINWEIS
Die Arbeit mit Arrays ist in ALP ab Version 3.0 möglich. In der aktuellen Implementierung werden jedoch nur eindimensionale Arrays unterstützt.

Die Verbindungslinien zwischen den Array-Variablen im Schaltungsprogramm sind grün.

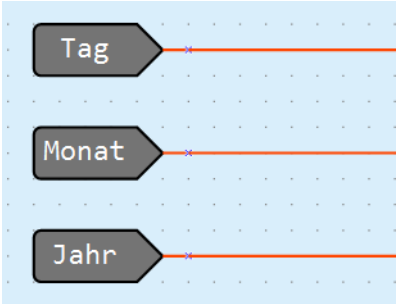


6.2 Servicevariablen

Servicevariablen sind mit den Geräteeinstellungen verknüpft und können nur gelesen werden. Die Menge der Dienstvariablen kann je nach Gerät unterschiedlich sein. Wenn das Gerät über eine LoRa-Kommunikationsschnittstelle verfügt, werden die Servicevariablen der Schnittstelle auch in Registerkarte ***Service*** angezeigt.



Die Blöcke der Servicevariablen werden im Schaltprogramm grau hinterlegt dargestellt.

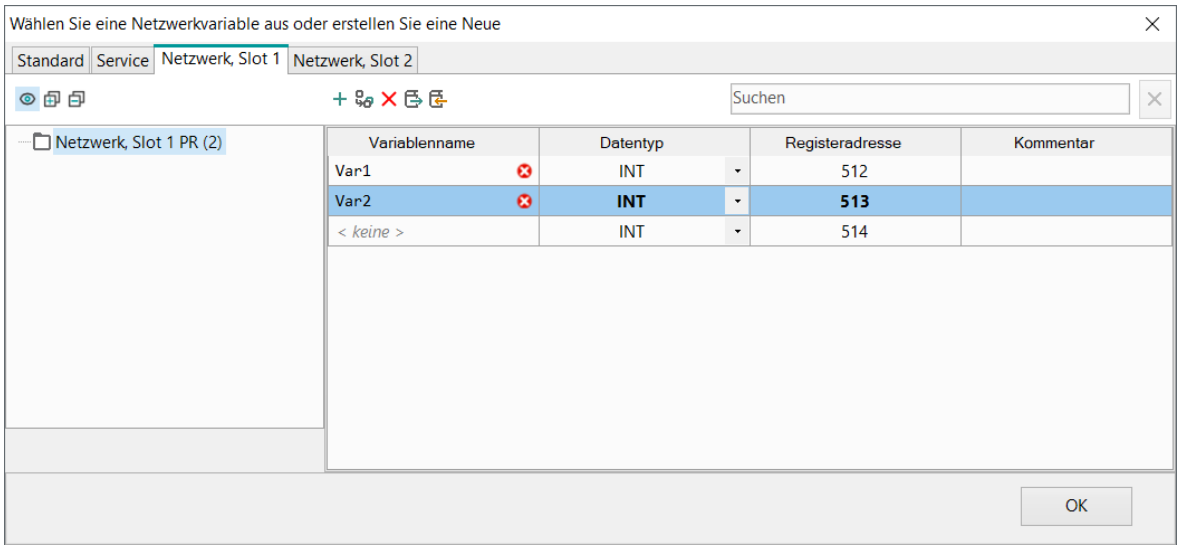


Servicevariablen für die Geräte der zweiten Generation

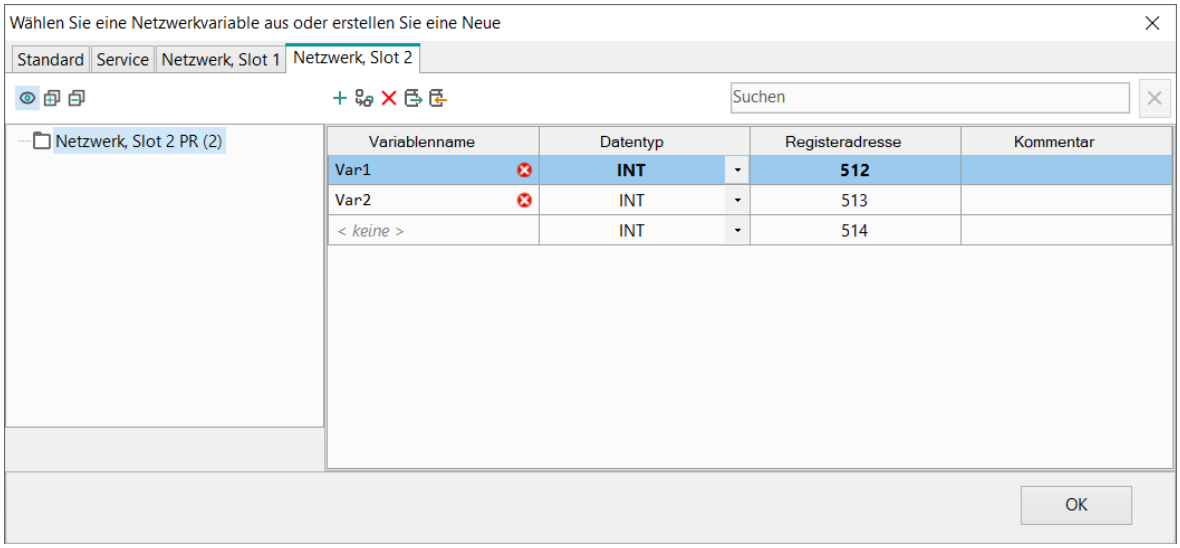
In den Geräten *der zweiten Generation* gibt es keine Servicevariablen. Anstelle von Servicevariablen werden Variablen an Geräteparameter gebunden (siehe Abschnitt *Variablen mit Parametern verknüpfen*).

6.3 Netzwerkvariablen

Jeder Schnittstellensteckplatz verfügt in der Tabelle über eine eigene Registerkarte.
Wenn die Schnittstelle als Master konfiguriert ist, gibt es innerhalb des Slot-Tabs für jedes Slave-Gerät einen eigenen Tab. Der Slave-Tab enthält die abzufragenden Variablen für dieses Slave-Gerät.



Das Löschen von Netzwerkvariablen und deren Referenzen erfolgt auf die gleiche Weise wie das Löschen von Standardvariablen.
Weitere Einzelheiten zu Netzwerkvariablen für die Masterschnittstelle finden Sie im Abschnitt *Master-Modus*.
Ist die Schnittstelle als Slave konfiguriert, werden alle vom Master abzufragenden Netzwerkvariablen in einer Liste angezeigt.



6.4 Variablen mit Parametern verknuepfen

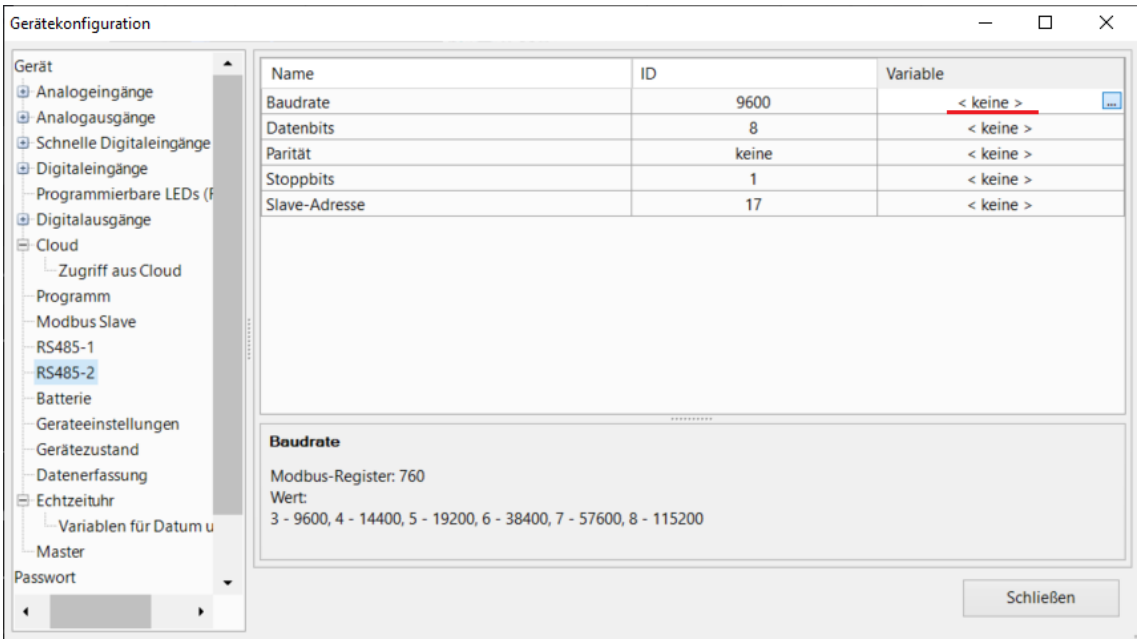
HINWEIS
Die Verknüpfung von Variablen mit Parametern ist nur für **Geräte der zweiten Generation** verfügbar. Eine Liste der Geräte der zweiten Generation finden Sie im Abschnitt *Einfürung*.

In Geräten der zweiten Generation können Sie Variablen mit Geräteparametern verknüpfen und im Programm verwenden. Variablen, die mit Parametern verknüpft sind, können ausgelesen und geändert werden.
So verknüpfen Sie eine Variable mit einem Geräteparameter:

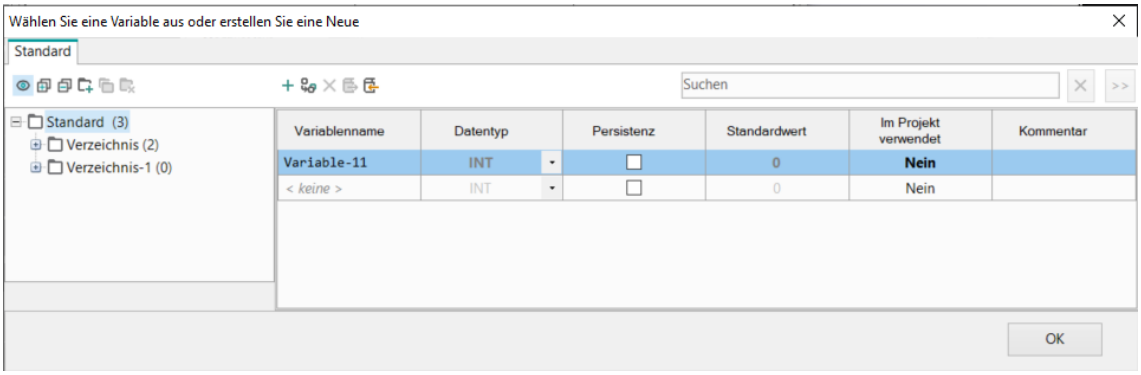
1. Öffnen Sie die **Geräteeinstellungen**.

   HINWEIS
   Die Variablenverknüpfung ist nicht für alle Parameter verfügbar.

2. Wählen Sie den Parameter für die Verknüpfung aus und klicken Sie auf die Schaltfläche "…" in der Spalte **Variable** des Einstellungsfensters.



3. Wählen Sie im sich öffnenden Variablenfenster eine Variable für die Verknüpfung aus oder erstellen Sie eine. Grau hinterlegte Variablen sind schreibgeschützt.



Der Name der verknüpften Variable wird in der Spalte "Variable" neben dem Parameter angezeigt.

Nach der Verknüpfung mit Parametern können die Variablen im Programm verwendet werden. Dies ermöglicht das Auslesen und Ändern von Parameterwerten innerhalb des Programms.

HINWEIS
Wenn eine mit einem Parameter verknüpfte Variable zum Schreiben über einen Block verwendet wird und sich der Parameterwert nicht geändert hat, wird der über den Block geschriebene Wert für einen Zyklus in der Variable gespeichert. Im nächsten Zyklus wird der Variablenwert wieder aus dem Parameter ausgelesen.

6.5 Variablenblock kopieren und einfuegen

Variablen können von Projekt zu Projekt kopiert werden, um sie wiederzuverwenden und die Entwicklungszeit zu verkürzen. Um eine Variable zu kopieren, wählen Sie den Variablenblock im Quellprojekt aus und klicken Sie auf die Schaltfläche in der Symbolleiste oder wählen Sie den Befehl **Kopieren** aus dem Kontextmenü des Blocks. Die Variable wird in ein anderes Projekt eingefügt, indem Sie auf die Schaltfläche in der Symbolleiste klicken oder den Befehl **Einfügen** aus dem Kontextmenü der Arbeitsfläche auswählen.

Variablen

Wird ein kopierter Variablenblock in ein Projekt eingefügt, werden alle seine Eigenschaften geprüft. Existiert bereits eine Variable mit exakt übereinstimmenden Eigenschaften im Projekt, wird diese mit dem Block verknüpft. Variablen mit abweichenden Eigenschaften werden automatisch im Projekt erstellt. Das Kopieren eines Array-Elements von einem Projekt in ein anderes folgt diesen Regeln:

– Wenn das Projekt, in das das Element eingefügt wird, kein Array mit Eigenschaften (Typ, Größe) enthält, die mit denen des Arrays, zu dem das kopierte Element gehört, identisch sind, wird ein neues Array erstellt, das zusammen mit dem Element die Eigenschaften des ursprünglichen Arrays vollständig repliziert.
– Falls das Zielprojekt bereits ein Array mit identischen Eigenschaften enthält, wird kein neues Array erstellt.

Die Regeln zum Kopieren von Array-Variablen sind dieselben wie für Standardvariablen. Wenn Variablennamen übereinstimmen, sich aber andere Eigenschaften unterscheiden, wird eine neue Variable erstellt. Der Namenskonflikt wird jedoch in der Variablentabelle angezeigt, und eine der Variablen muss umbenannt werden.

HINWEIS
Variablen vom Fließkommatyp können nicht in ein Projekt für ein Gerät eingefügt werden, das den Fließkomma-Datentyp nicht unterstützt.

HINWEIS
Nichtflüchtige Variablen können nicht in ein Projekt für ein Gerät eingefügt werden, das diese nicht unterstützt.

Servicevariablen

Servicevariablen können nicht in ein Projekt für ein Gerät kopiert werden, das keine Echtzeituhren unterstützt.

Netzwerkvariablen

Nur Netzwerkvariablen im Slave-Modus mit identischer Slot-Nummer können in ein anderes Projekt kopiert werden. Netzwerkvariablen im Master-Modus müssen manuell erstellt werden. Beim Einfügen eines kopierten Netzwerkvariablenblocks in ein Projekt werden alle Variableneigenschaften überprüft. Wenn das Projekt eine Variable mit exakt übereinstimmenden Eigenschaften enthält, wird diese im Block dupliziert. Variablen mit abweichenden Eigenschaften werden automatisch im Projekt erstellt. Wenn Variablennamen oder Register übereinstimmen, aber andere Eigenschaften abweichen, wird eine neue Variable erstellt, jedoch tritt ein Register- oder Namenskonflikt auf. Der Konflikt kann in der Variablentabelle behoben werden, indem ein anderer Name oder andere Register zugewiesen werden.

7 Bibliothek

Wenn ein Projekt geöffnet ist, enthält das Panel ***Bibliothek*** die folgenden Bibliotheken:

– *Funktionen*
– *Funktionsbausteine*
– *Projektmakros*
– *ST-Funktionen*
– *Funktionsblöcke auf ST*

Wählen Sie im unteren Bereich des Panels ein Symbol aus, um den entsprechenden Inhalt anzuzeigen.
Die Bibliothek **Projektmakros** umfasst die Makros, die aus der Online-Datenbank erstellt, importiert oder in das Projekt aufgenommen wurden.
Die Anzeigeoptionen können mithilfe der Symbole in der Symbolleiste des Bedienfelds geändert werden.

7.1 Funktionen

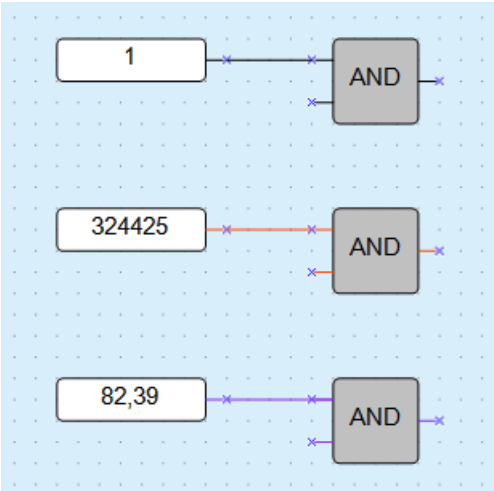
Die Bibliothek enthält folgende Funktionsgruppen:

– *Logische Operatoren*
– *Mathematische Operatoren*
– *Relationale Operatoren*
– *Bitshift Operatoren*
– *Bitoperatoren*

7.1.1 Logische Operatoren

– *Konjunktion (AND)*
– *Disjunktion (OR)*
– *Negation (NOT)*
– *Exklusives ODER (XOR)*

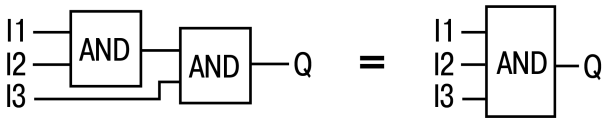
Die logischen Operatoren können mit BOOL- oder INT-Variablen operieren.
Wenn die Eingabewerte INT sind, wird die Operation bitweise ausgeführt und die Ausgabe ist ebenfalls ein INT.



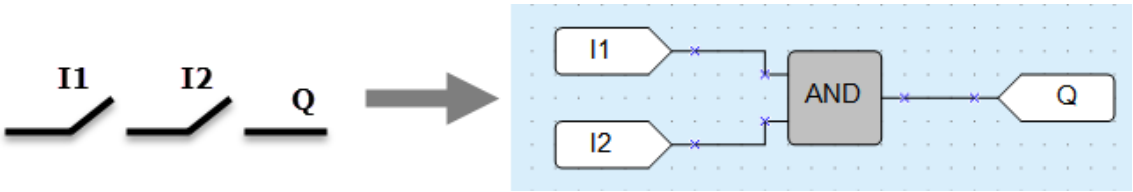
Bei den Operatoren **UND** und **ODER** ist zu beachten, dass nicht verbundene Bausteineingänge folgende Zustände haben:

– für **UND** – TRUE
– für **ODER** – FALSE

In diesem Fall fungieren die Blöcke als Signalverstärker. Um die Anzahl der Eingänge für logische Operatoren zu erhöhen, wird ihre Kaskadenschaltung verwendet:



7.1.1.1 Konjunktion (UND)



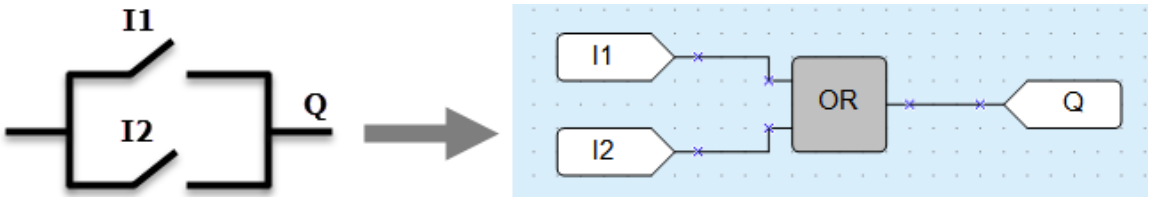
Der Ausgang **Q** ist **TRUE**, wenn beide Eingänge **TRUE** sind. Die Funktion **AND** stellt eine serielle Verbindung in einem Stromkreis dar.
Wahrheitstabelle:

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

Beispiel für eine bitweise Operation mit ganzzahligen Eingaben:

AND	0011 (Dezimalzahl 3)
	0101 (Dezimalzahl 5)
	0001 (Dezimalzahl 1)

7.1.1.2 Disjunktion (OR)



Der Ausgang **Q** ist **TRUE**, wenn mindestens einer der Eingänge **TRUE** ist. Die Funktion **OR** stellt eine Parallelschaltung in einem Stromkreis dar.
Wahrheitstabelle:

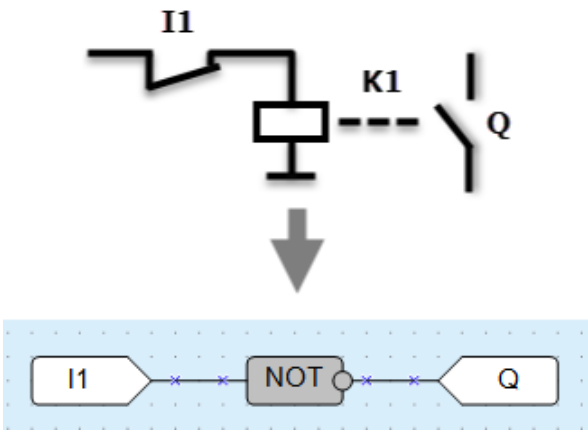
I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	1

Beispiel für eine bitweise Operation mit ganzzahligen Eingaben:

OR	0011
	0101
	0111

7.1.1.3 Negation (NOT)

Die Funktion **NOT** invertiert das Signal. Der Ausgang **Q** ist **True**, wenn der Eingang **False** ist und umgekehrt.



Wahrheitstabelle:

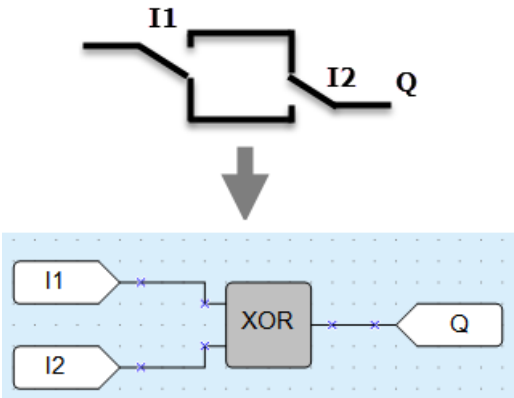
I1	Q
0	1
1	0

Beispiel für eine bitweise Operation mit ganzzahligen Eingaben:

NOT	01
	10

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

7.1.1.4 Exklusives ODER (XOR)



Die Ausgang **Q** ist **TRUE**, wenn nur eine der Eingaben **TRUE** ist.
Wahrheitstabelle:

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

Beispiel für eine bitweise Operation mit ganzzahligen Eingaben:

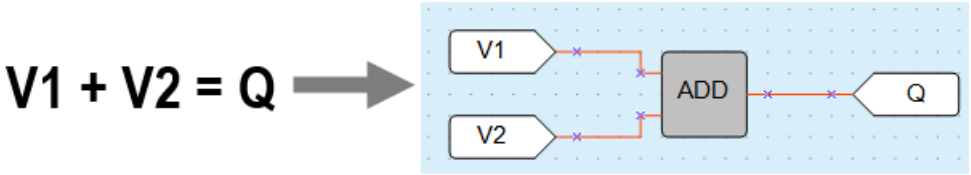
XOR	0011
	0101
	0110

7.1.2 Mathematische Operatoren

Für unterschiedliche Datentypen gibt es unterschiedliche Operatoren:

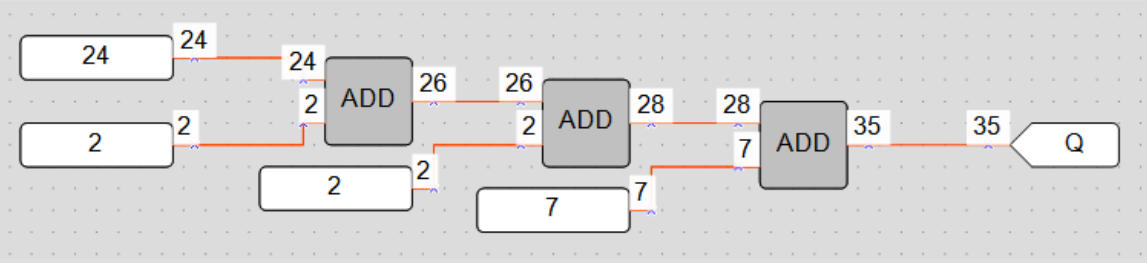
Operator	INT	REAL
Addition	*ADD*	*fADD*
Subtraktion	*SUB*	*fSUB*
Multiplikation	*MUL*	*fMUL*
Division	*DIV*	*fDIV*
Modul	*MOD*	–
Potenzfunktion	–	*fPOW*
Absoluter Wert	–	*fABS*

7.1.2.1 Addition (ADD, fADD)



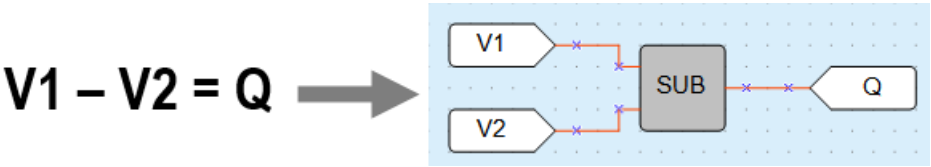
Die Funktion ADD arbeitet mit INT-Variablen, während die Funktion fADD mit REAL-Variablen arbeitet. Der Ausgabewert Q ist die Summe der Eingabewerte.

Beispiel:



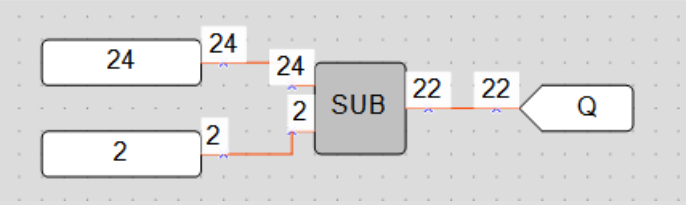
Der Ausgabewert darf 4294967295 (32 Bit) nicht überschreiten. Andernfalls werden die zusätzlichen Bits abgeschnitten.

7.1.2.2 Subtraktion (SUB, fSUB)

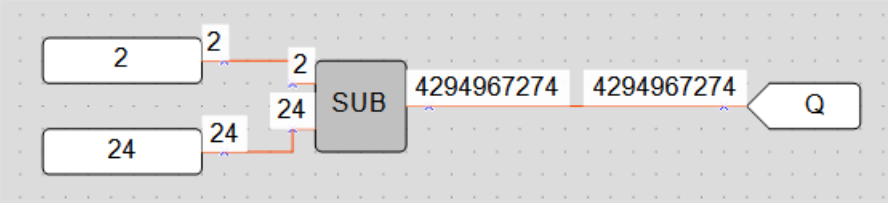


Die Funktion **SUB** arbeitet mit INT-Variablen, während die Funktion **fSUB** mit REAL-Variablen arbeitet.
Der Ausgabewert **Q** ist das Ergebnis der Subtraktion des Wertes **I2** vom Wert **I1**.

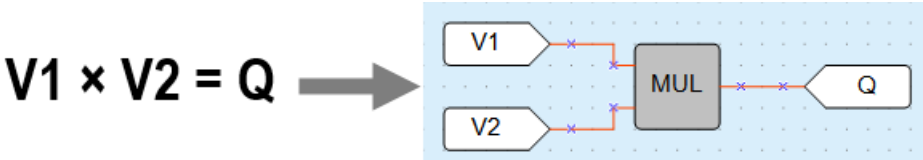
Beispiel:



Wenn der Wert **I1** kleiner als der Wert **I2** ist, wird die Ausgabe wie folgt berechnet:
Q = I1 + 0x100000000 – I2
0x100000000 = 4294967296

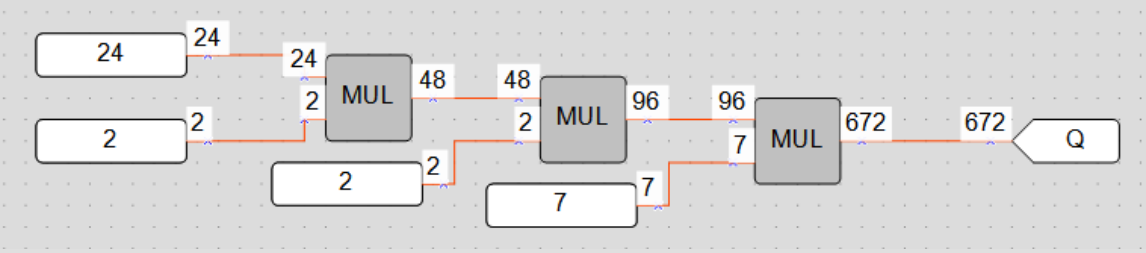


7.1.2.3 Multiplikation (MUL, fMUL)



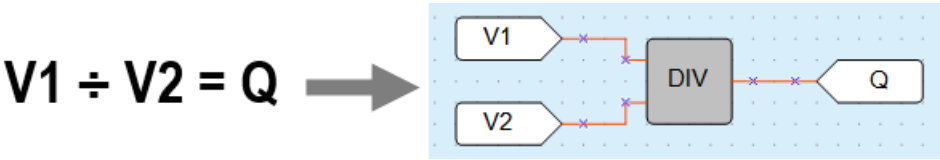
Die Funktion **MUL** arbeitet mit INT-Variablen, während die Funktion **fMUL** mit REAL-Variablen arbeitet.
Der Ausgabewert **Q** ist das Produkt der Eingabewerte.

Beispiel:



Der Ausgabewert darf 4294967295 (32 Bit) nicht überschreiten. Sollte dies dennoch passieren, werden die zusätzlichen Bits abgeschnitten.

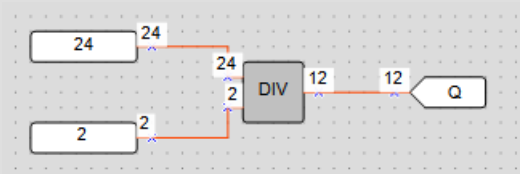
7.1.2.4 Division (DIV, fDIV)



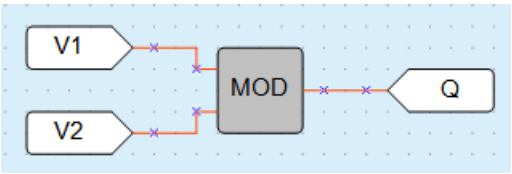
Die Funktion **DIV** operiert mit INT-Variablen, die Funktion **fDIV** operiert mit REAL-Variablen.
Der Ausgabewert **Q** ist der Quotient der Eingabewerte, wobei der Wert **I1** der Dividend und der Wert **I2** der Divisor ist.

Wenn der Quotient kein INT ist, wird er auf einen INT abgerundet.
Im Falle einer Division durch 0 ist der Ausgabewert 0xFFFFFFFF.

Beispiel:

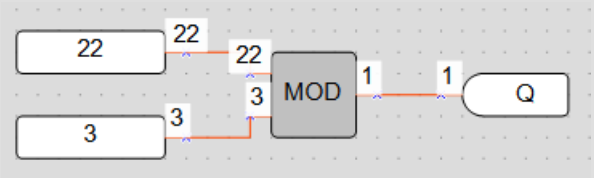


7.1.2.5 Modul (MOD)



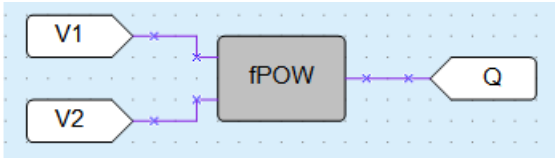
Die Funktion **MOD** arbeitet mit INT-Variablen. Die Ausgabe **Q** ist ein Divisionsrest der Eingabewerte.

Beispiel:



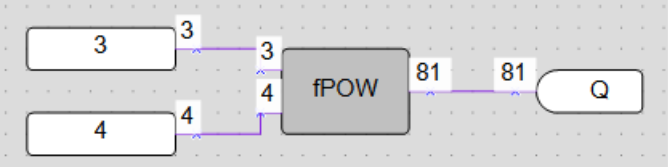
7.1.2.6 Potenzfunktion (fPOW)

$$V1^{(V2)} = Q$$



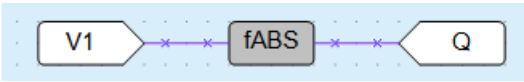
Die Funktion **fPOW** arbeitet mit REAL-Variablen.
Der Ausgabewert **Q** ist der Wert **I1** hoch dem Wert **I2**.

Beispiel:



7.1.2.7 Absolute Funktion (fABS)

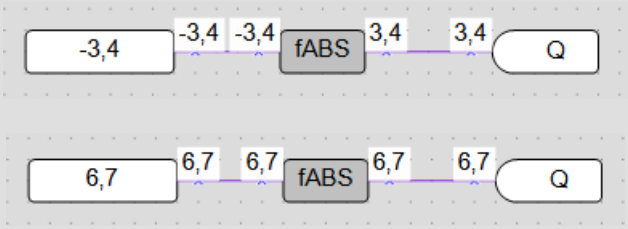
$$|V| = Q$$



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

Der Ausgabewert **Q** ist ein absoluter Wert des Eingabewerts.

Beispiel:

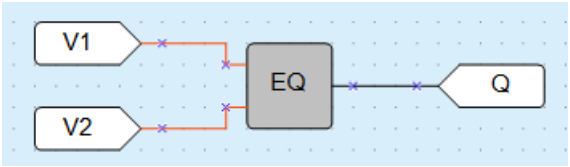


7.1.3 Relationale Operatoren

Die relationalen Operatoren sind Funktionen, die eine Art Beziehung zwischen zwei oder mehreren Werten testen oder definieren.

- *Gleich (EQ)*
- *Größer als (GT, fGT)*
- *Binäre Auswahl (SEL)*

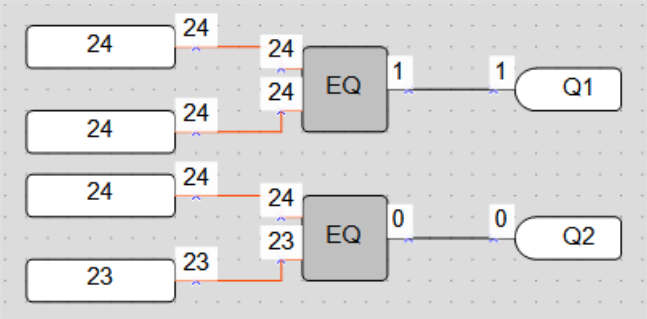
7.1.3.1 Gleich (EQ)



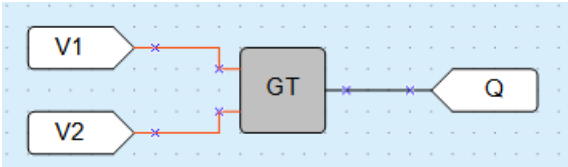
Die Funktion **EQ** arbeitet mit INT-Variablen.
Der Ausgabewert **Q** ist **True**, wenn die Werte **I1** und **I2** gleich sind.

- V1 = V2 → Q = 1
- V1 > V2 → Q = 0
- V1 < V2 → Q = 0

Beispiel:



7.1.3.2 Groesser als (GT, fGT)

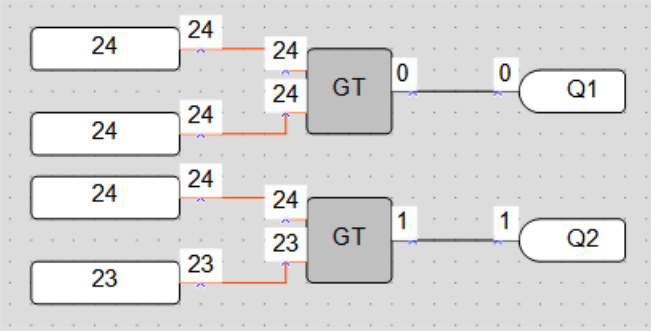


Die Funktion **GT** arbeitet mit INT-Variablen, während die Funktion **fGT** mit REAL-Variablen arbeitet.

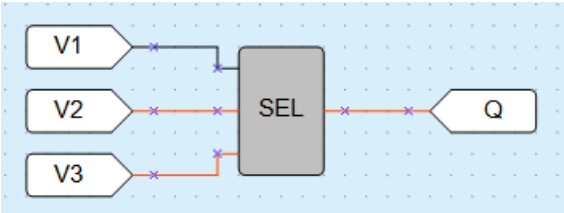
Der Ausgabewert **Q** ist **True**, wenn der Wert **I1** größer als der Wert **I2** ist.

– V1 = V2 → Q = 0
– V1 > V2 → Q = 1
– V1 < V2 → Q = 0

Beispiel:



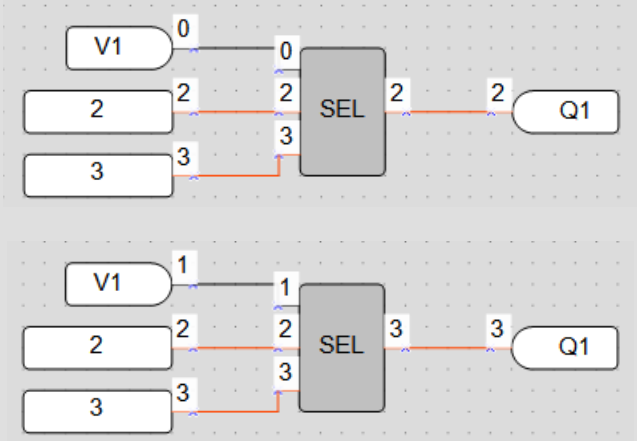
7.1.3.3 Binaere Auswahl (SEL, fSEL)



Die Funktion **SEL** arbeitet mit INT-Variablen, die Funktion **fSEL** arbeitet mit REAL-Variablen. Wenn der Wert **I1 False** ist, wird der Ausgabewert **Q** auf den Wert **I2** gesetzt, andernfalls auf den Wert **I3**.

– V1 = 0 → Q = V2
– V1 = 1 → Q = V3

Beispiel:



7.1.4 Bitshift Operatoren

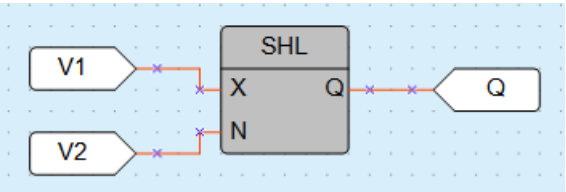
Die Bitverschiebungsoperatoren behandeln eine Variable als eine Reihe von Bits, die nach links oder rechts verschoben (verschoben) werden können.

– *Schieberegister links (SHL)*

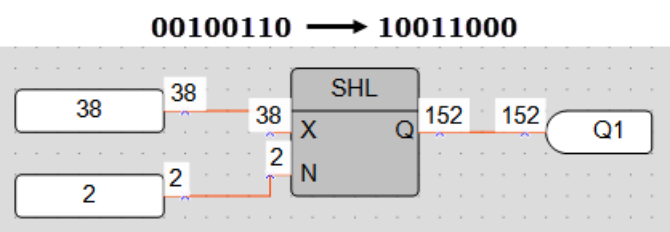
– *Schieberegister rechts (SHR)*

7.1.4.1 Schieberegister links (SHL)

Die Funktion **SHL** arbeitet mit INT-Variablen. Sie wird verwendet, um alle Bits des Operanden **X** um die Anzahl **N** Bits nach links zu verschieben; frei gewordene Bits werden mit Nullen aufgefüllt. Das Ergebnis wird auf den Ausgang **Q** gesetzt.

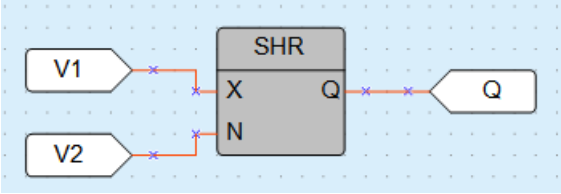


Beispiel:
Linksverschiebung der Zahl 38 (dezimal) = 00100110 (binär) um 2 Bit

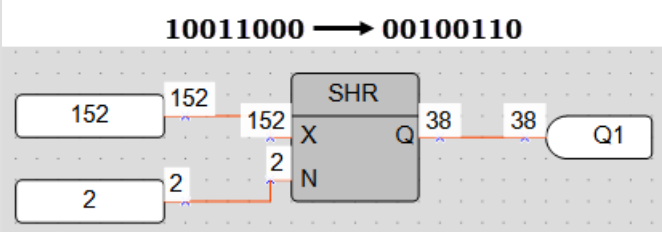


7.1.4.2 Schieberegister rechts (SHR)

Die Funktion **SHR** arbeitet mit INT-Variablen. Sie wird verwendet, um alle Bits des Operanden **X** um die Anzahl **N** Bits nach rechts zu verschieben; frei gewordene Bits werden mit Nullen aufgefüllt. Das Ergebnis wird auf den Ausgang **Q** gesetzt.



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



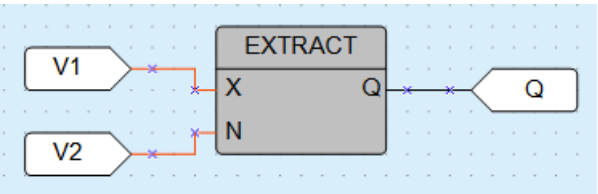
7.1.5 Bitoperatoren

Der Bitoperator behandelt einen Wert als eine Reihe von Bits, um Operationen an einem oder mehreren einzelnen Bits eines Operanden auszuführen.

– *Einzelnes Bit auslesen (EXTRACT)*

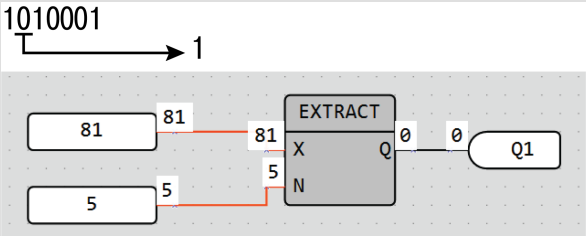
- *Einzelnes Bit setzen (PUTBIT)*
- *Decoder (DC32)*
- *Encoder (CD32)*

7.1.5.1 Einzelnes Bit auslesen (EXTRACT)

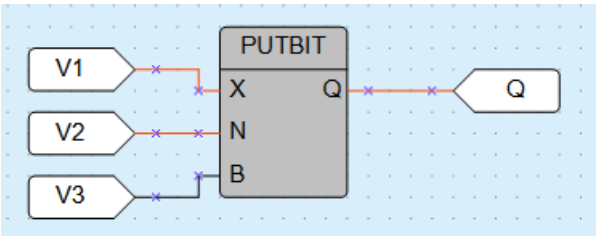


Der Ausgabewert **Q** (BOOL) der Funktion **EXTRACT** ist der Wert des Bits **N** (INT) im Operanden **X** (INT). Die Bitnummerierung ist nullbasiert.

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

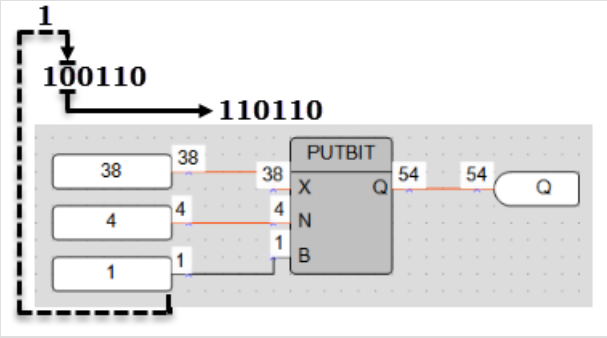


7.1.5.2 Einzelnes Bit setzen (PUTBIT)

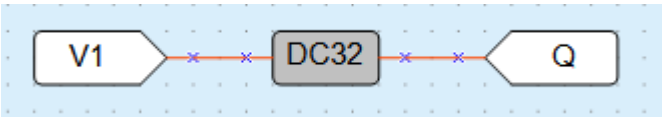


Dieser Ausgabewert **Q** (INT) ist der Wert des Operanden **X** (INT), wobei das Bit **N** (INT) auf den Wert am Eingang **B** (BOOL) gesetzt ist. Die Bitnummerierung ist nullbasiert.

Beispiel:
Setzen des 4. Bits auf 1 in der Zahl 38 (dezimal) = 100110 (binär):



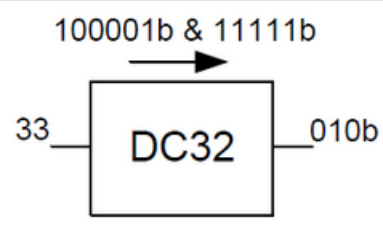
7.1.5.3 Decoder (DC32)



Der Decoder wandelt einen Binärcode am Eingang in einen Positionscode am Ausgang um.
Die Dekodierung erfolgt bitweise durch die logische Operation **AND** mit dem Operanden 0x1F (11111b).
Wahrheitstabelle:

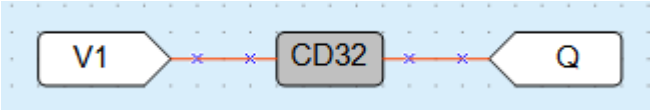
Binärcode					Positionscode								
5	4	3	2	1	32	31	…	6	5	4	3	2	1
0	0	0	0	0	0	0		0	0	0	0	0	1
0	0	0	0	1	0	0		0	0	0	0	1	0
0	0	0	1	0	0	0		0	0	0	1	0	0
0	0	0	1	1	0	0		0	0	1	0	0	0
0	0	1	0	0	0	0		0	1	0	0	0	0
…					…			…					
1	1	1	0	1	0	0		0	0	0	0	0	0
1	1	1	1	0	0	1		0	0	0	0	0	0
1	1	1	1	1	1	0		0	0	0	0	0	0

Beispiel:



7.1.5.4 Encoder (CD32)

Der Encoder (CD32) wird verwendet, um den Positionscode am Eingang in den Binärcode am Ausgang umzuwandeln.



Der Encoder wandelt einen Positionscode am Eingang in einen Binärcode am Ausgang um.
Wenn der Positionscode mehr als ein „1“-Bit enthält, arbeitet der Encoder nur mit dem höchstwertigen „1“-Bit.
Wahrheitstabelle:

Binärcode					Positionscode								
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

Binärcode					Positionscode								
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

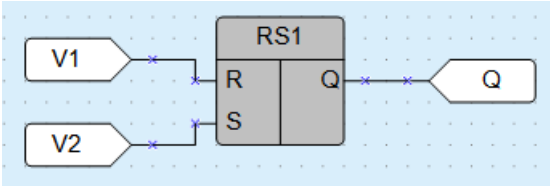
7.2 Funktionsbausteine

- *Trigger*
- *Timer*
- *Generatoren*
- *Zähler*
- *Analog*

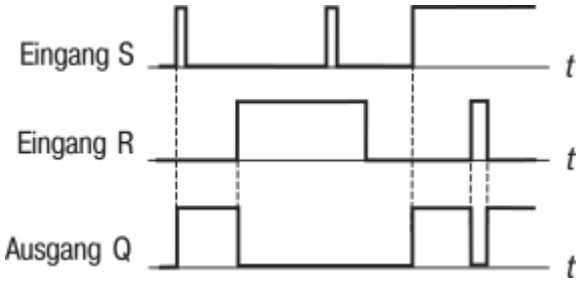
7.2.1 Trigger

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

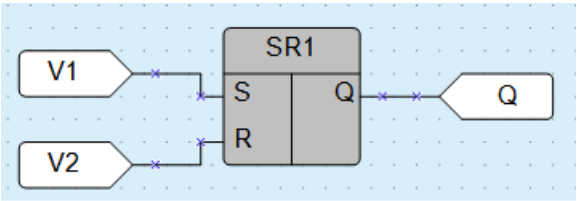
7.2.1.1 RS Trigger mit Reset-Prioritaet



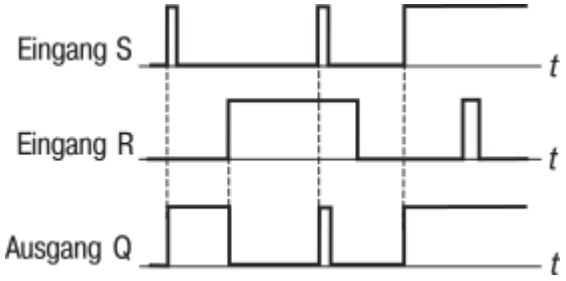
Der Ausgang **Q** ist **True** bei einer steigenden Flanke am Eingang **S** (Set) und **False** bei einer steigenden Flanke am Eingang **R** (Reset). Der Eingang **R** hat die höhere Priorität.



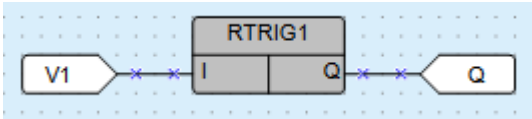
7.2.1.2 SR-Trigger mit Set-Prioritaet



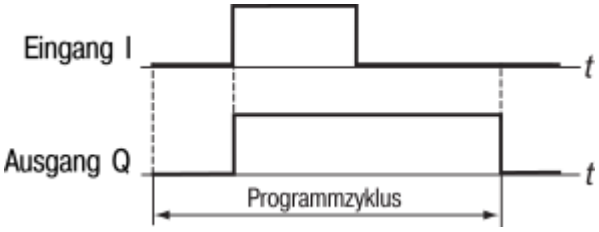
Der Ausgang **Q** ist **True** bei einer steigenden Flanke am Eingang **S** (Set) und **False** bei einer steigenden Flanke am Eingang **R** (Reset). Der Eingang **S** hat die höhere Priorität.



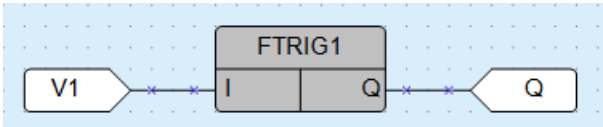
7.2.1.3 Steigende Flanke (RTRIG)



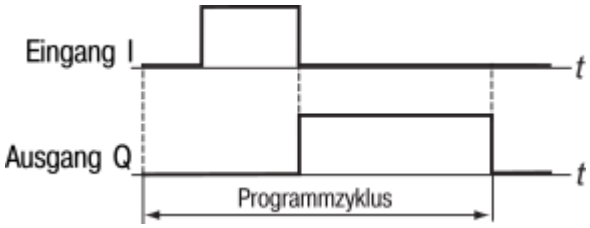
Detektor für steigende Flanke erscheint.
Der Ausgang **Q** bleibt **False**, bis am Eingang **I** eine steigende Flanke anliegt. Sobald der Eingang **I** **True** wird, wird der Ausgang **True** und bleibt für einen Programmzyklus **True**.



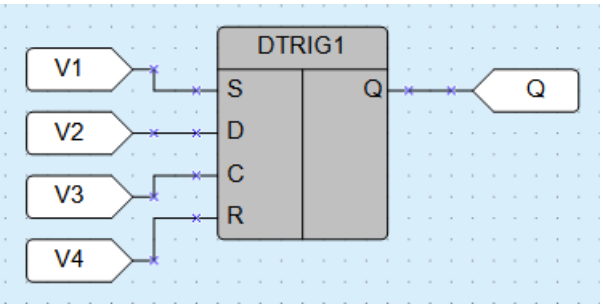
7.2.1.4 Fallende Flanke (FTRIG)



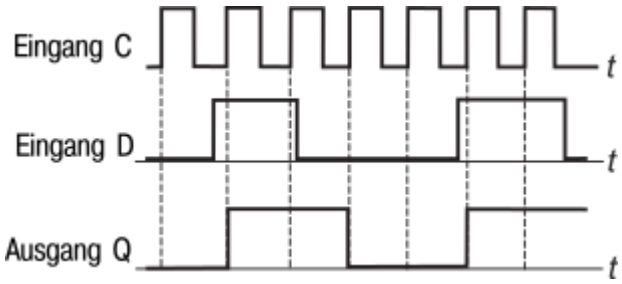
Detektor für eine fallende Flanke.
Der Ausgang **Q** bleibt **False**, bis am Eingang **I** eine fallende Flanke anliegt. Sobald der Eingang **I** **False** wird, wird der Ausgang **True** und bleibt für einen Programmzyklus **True**.



7.2.1.5 D-Trigger (DTRIG)



Der D-Trigger erzeugt einen Impuls am Ausgang **Q** mit der am Eingang **D** angegebenen Impulsdauer und synchronisiert mit dem Taktimpuls am Eingang **C**.
Wenn der Eingang **D True** ist, wird der Ausgang **Q** mit einer steigenden Flanke des Taktimpulses am Eingang **C True**.
Wenn der Eingang **D False** ist, wird der Ausgang **Q** mit einer steigenden Flanke des Taktimpulses am Eingang **C False**.

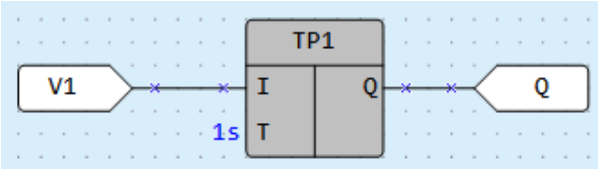


Der Ausgang **Q** kann durch eine steigende Flanke am Eingang **S** zwangsweise auf **True** gesetzt werden (Set) und durch eine steigende Flanke am Eingang **R** zwangsweise auf **False** zurückgesetzt werden (Reset), unabhängig von den Zuständen der Eingänge **C** und **D**. Der Eingang **R** hat dabei die höhere Priorität.

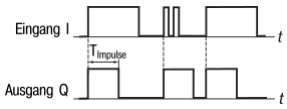
7.2.2 Timer

- *Impuls (TP)*
- *Einschaltverzögerung (TON)*
- *Ausschaltverzögerung (TOF)*
- *Schaltuhr (CLOCK)*
- *Wochenschaltuhr (CLOCKWEEK)*.

7.2.2.1 Impuls (TP)

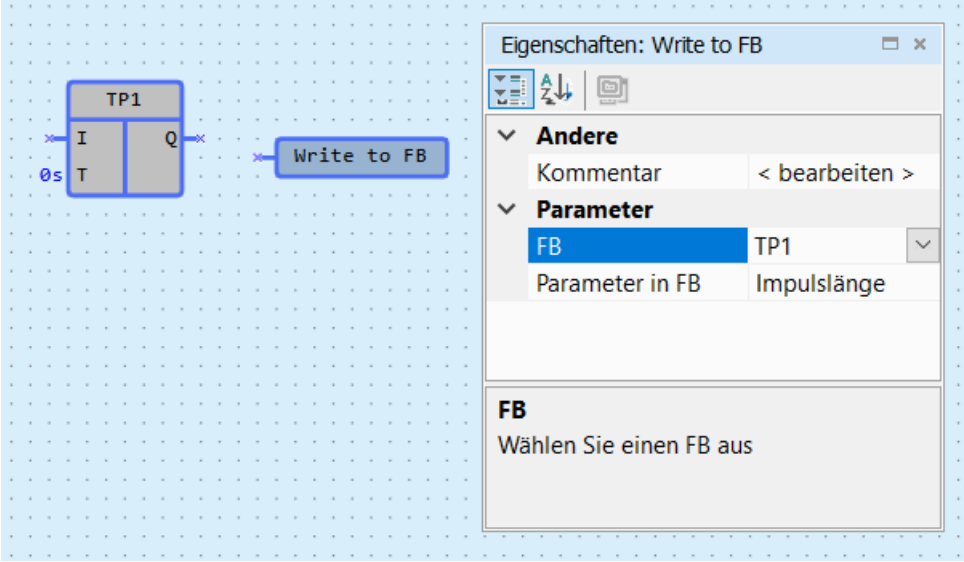


Der Block **TP** dient zur Erzeugung eines Ausgangsimpulses mit der angegebenen Impulsdauer.

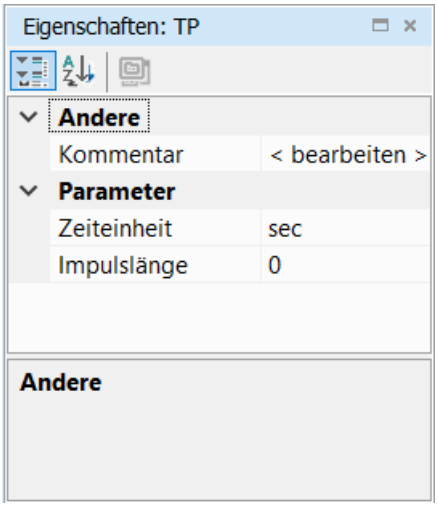


Der Ausgang **Q** wird mit einer steigenden Flanke am Eingang **I** für die am Eingang **T** angegebene Zeit **True**. Während dieser Zeit bleibt der Ausgang **Q** unabhängig vom Signalwechsel am Eingang **I** **True**. Mit Impulsende wird der Ausgang **Q** auf **False** zurückgesetzt.

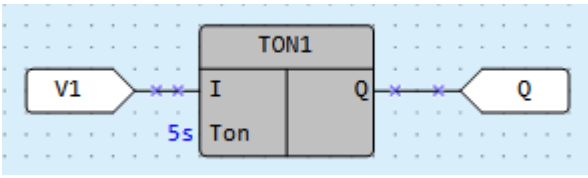
Zeitbereich: 0…4147200000 ms oder 48 Tage.

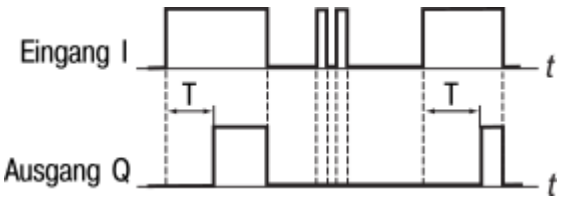


Die Werte der Parameter des Funktionsbausteins können mithilfe eines „In Funktionsbaustein schreiben“-Bausteins festgelegt werden, wie in der obigen Abbildung dargestellt, oder im Eigenschaftenfeld konfiguriert werden:



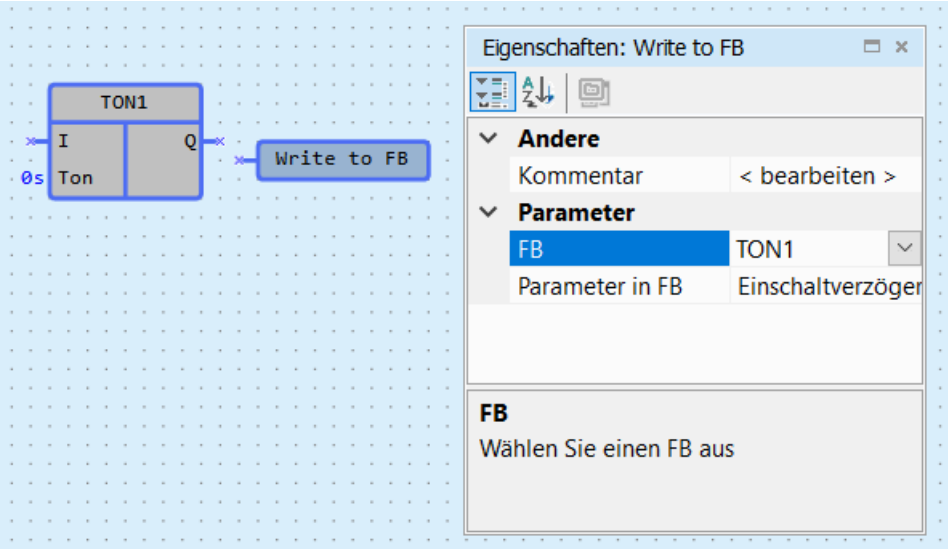
7.2.2.2 Einschaltverzoegerung (TON)



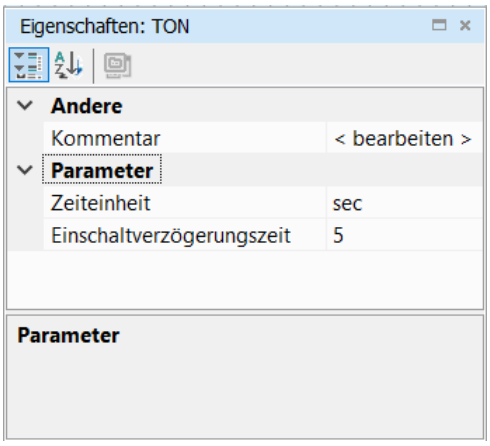


Der Ausgang **Q** = **False**, wenn der Eingang **I** = **False** ist. Die am Eingang **TON** angegebene Verzögerungszeit startet mit einer steigenden Flanke am Eingang **I**. Nach Ablauf der Zeit **TON** wird

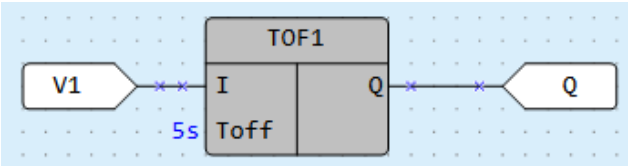
der Ausgang **QTrue** und bleibt **False**, bis am Eingang **I** eine fallende Flanke erscheint. Eingangsänderungen, die kürzer als **TON** sind, werden ignoriert.
Zeitbereich: 0…4147200000 ms oder 48 Tage.

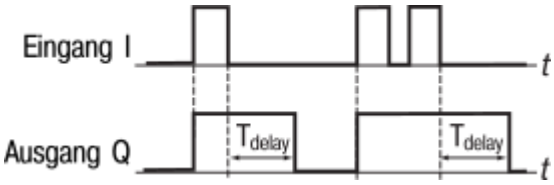


Die Werte der Parameter des Funktionsbausteins können mithilfe eines „In Funktionsbaustein schreiben“-Bausteins festgelegt werden, wie in der obigen Abbildung dargestellt, oder im Eigenschaftenfeld konfiguriert werden:



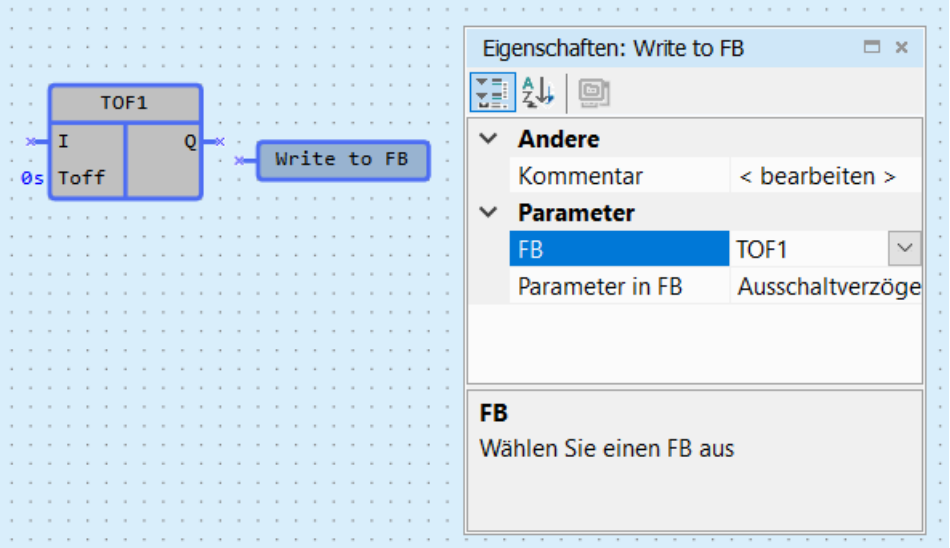
7.2.2.3 Ausschaltverzoegerung (TOF)



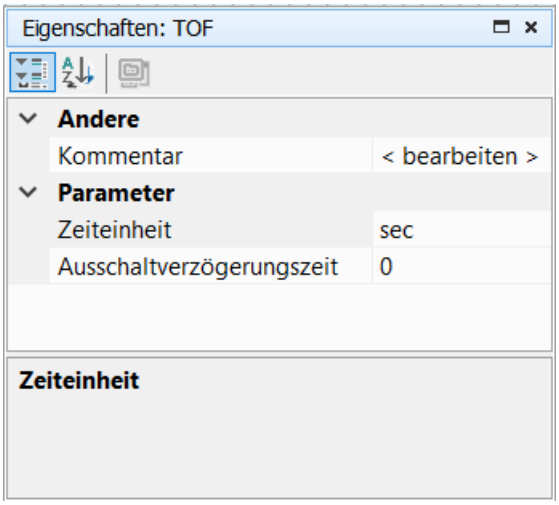


Der Ausgang **Q** = **False**, wenn der Eingang **I** = **False** ist. Die am Eingang **TOFF** angegebene Verzögerungszeit startet mit einer fallenden Flanke am Eingang **I**. Wenn die Zeit **TOFF** abgelaufen ist, wird der Ausgang **Q False** und bleibt **False**, bis am Eingang **I** eine steigende Flanke erscheint. Eingangsänderungen, die kürzer als **TOFF** sind, werden ignoriert.

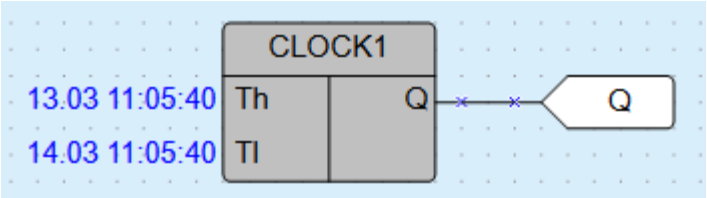
Zeitbereich: 0…4147200000 ms oder 48 Tage.



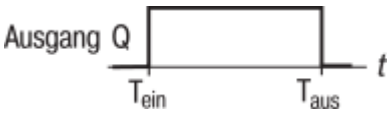
Die Werte der Parameter des Funktionsbausteins können mithilfe eines „In Funktionsbaustein schreiben“-Bausteins festgelegt werden, wie in der obigen Abbildung dargestellt, oder im Eigenschaftenfeld konfiguriert werden:



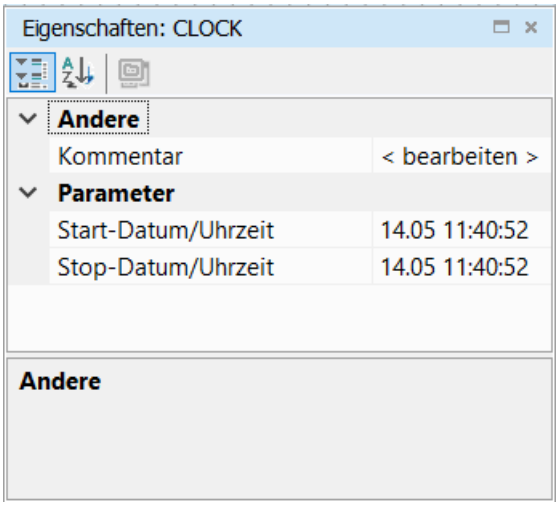
7.2.2.4 Schaltuhr (CLOCK)



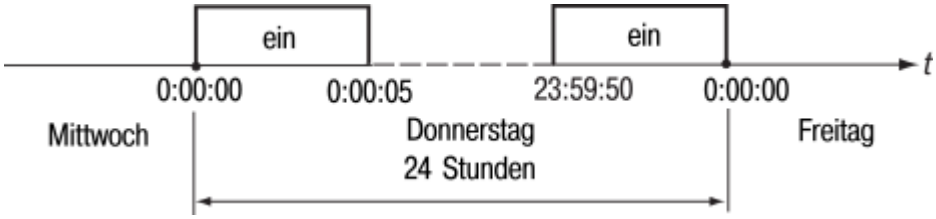
Der Baustein **CLOCK** ist ein Intervalltimer, der von einer Echtzeituhr gesteuert wird.



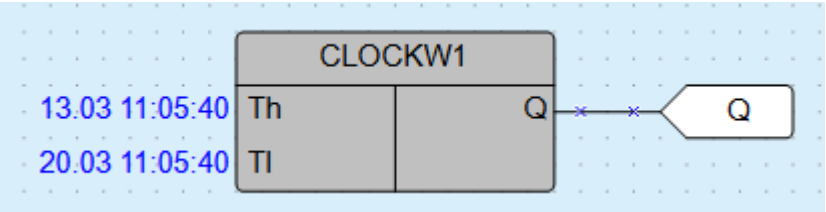
Die Zeiten **TH** und **TL** können im Eigenschaftenfeld eingestellt werden.



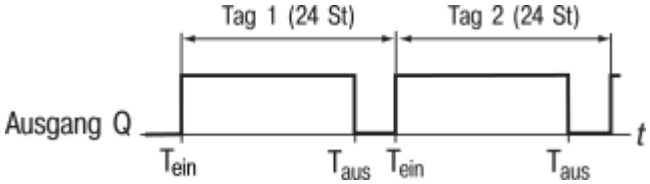
Zeitbereich: von 0,00 Sekunden bis 24 Stunden.
Wenn **TH<TL**, ist der Zustand des Ausgangs **Q** wie folgt:



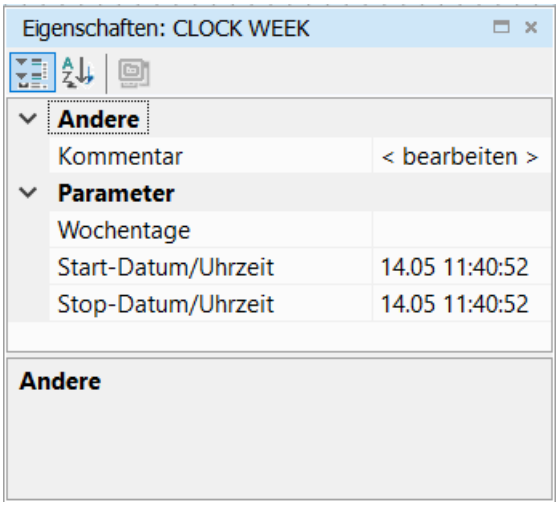
7.2.2.5 Wochenschaltuhr (CLOCKWEEK)



Der Block **CLOCKWEEK** ist ein Intervalltimer mit dem Parameter ***Wochentage***, der von einer Echtzeituhr gesteuert wird.



Die Zeiten **TH** und **TL** können im Eigenschaftenfeld eingestellt werden.

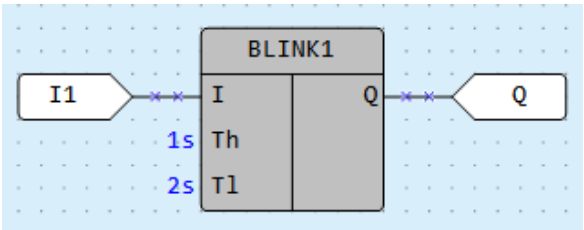


Zeitbereich: von 0,00 Sekunden bis 24 Stunden.

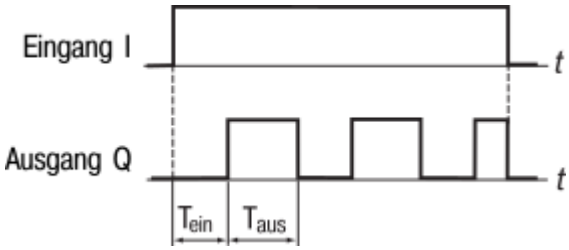
7.2.3 Generatoren

– *Impulsgenerator (BLINK)*

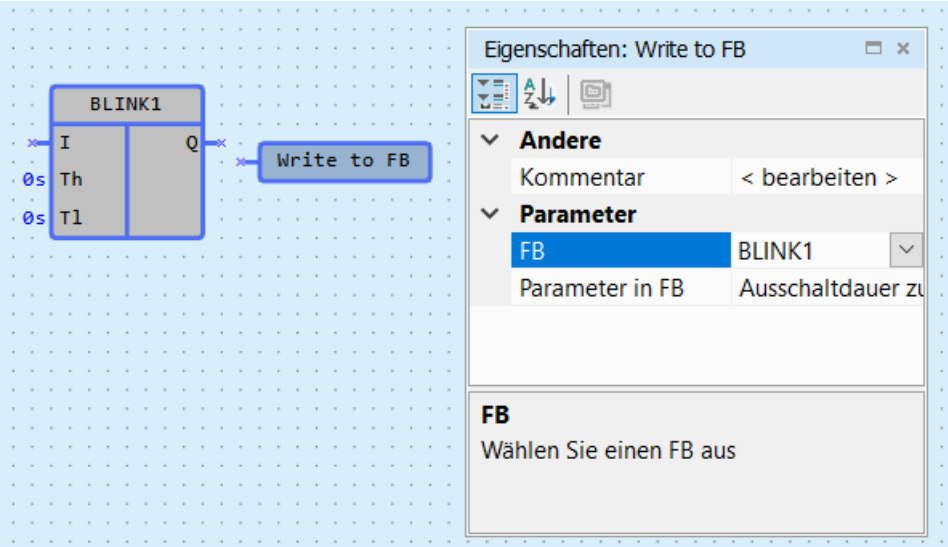
7.2.3.1 Impulsgenerator (BLINK)



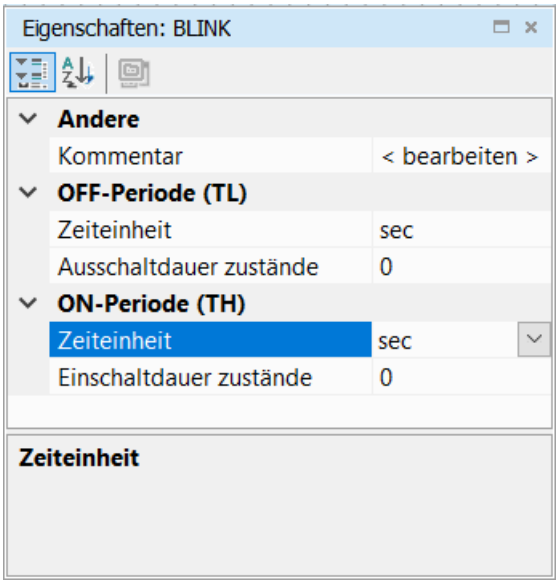
Wenn der Eingang **I True** wird, erzeugt der Block **BLINK** eine Rechteckwelle am Ausgang **Q** mit einer Periode von **TH** + **TL**, beginnend mit einem Intervall der Dauer von **TL**, gefolgt von einem Impuls der Dauer von **TH**. Dies geht so weiter, bis der Eingang **I False** ist.



Zeitbereich: 0…4233600000 Millisekunden oder 49 Tage.



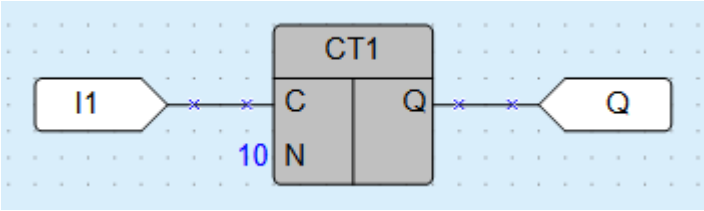
Die Werte der Parameter des Funktionsbausteins können mithilfe eines „In Funktionsbaustein schreiben“-Bausteins festgelegt werden, wie in der obigen Abbildung dargestellt, oder im Eigenschaftenfeld konfiguriert werden:



7.2.4 Zaehler

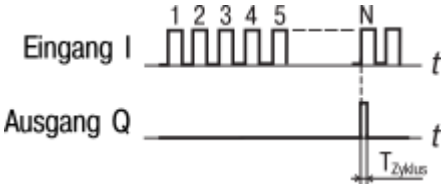
– *Schwellenwertzähler mit Auto-Reset (CT)*
– *Universalzähler (CTN)*
– *Schwellenwertzähler (CTU)*

7.2.4.1 Schwellenwertzaehler mit Auto-Reset (CT)

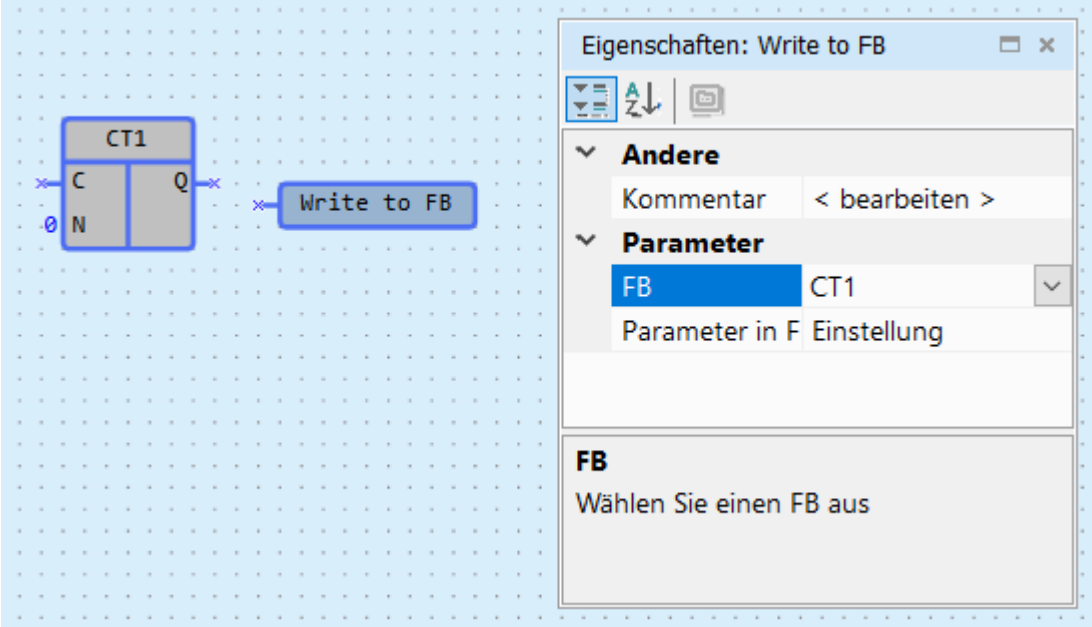


Der Ausgang **Q** ist vom Typ BOOL. Überschreitet die am Eingang **C** gezählte Impulszahl den am Eingang **N** angegebenen Schwellwert (***Einstellung***), wird der Ausgang **Q True** und bleibt für einen Programmzyklus bestehen.

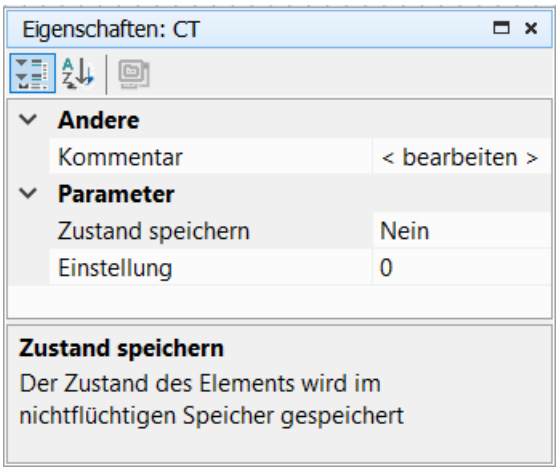
Die Funktionsweise des Zählers wird im folgenden Diagramm erläutert.



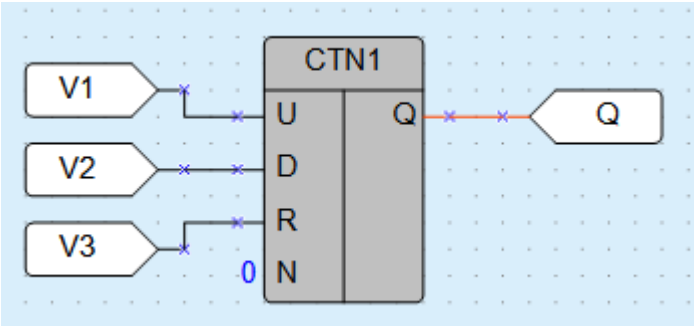
Schwellenwertbereich: 0…65535.



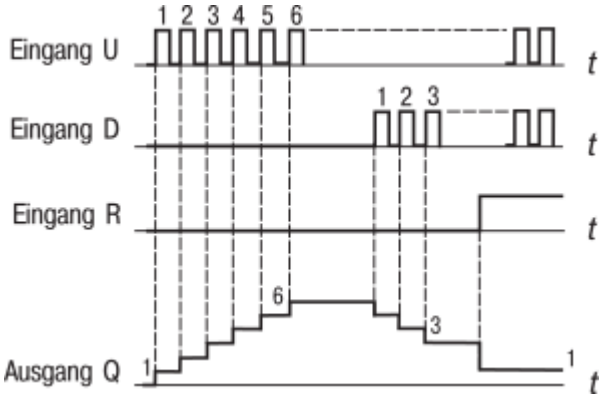
Die Werte der Parameter des Funktionsbausteins können mithilfe eines „In Funktionsbaustein schreiben“-Bausteins festgelegt werden, wie in der obigen Abbildung dargestellt, oder im Eigenschaftenfeld konfiguriert werden:



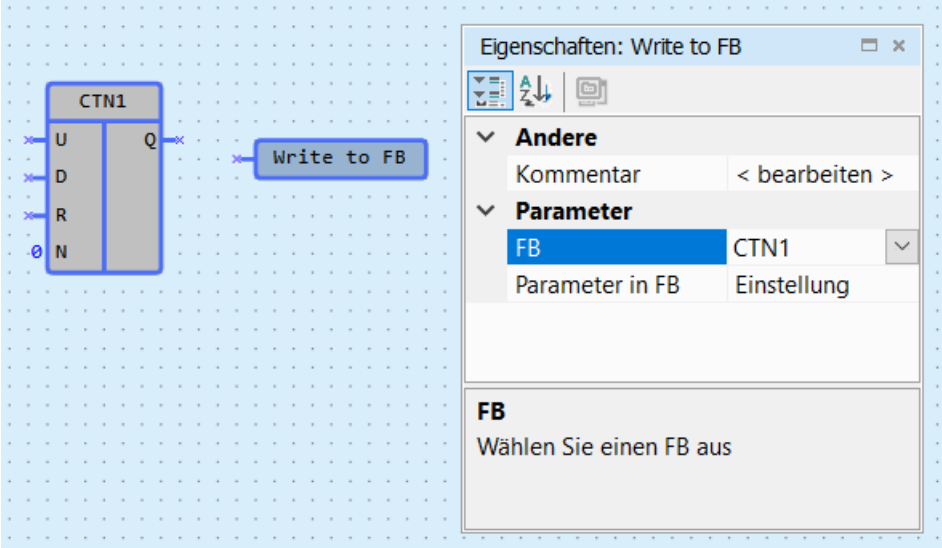
7.2.4.2 Universalzaehler (CTN)



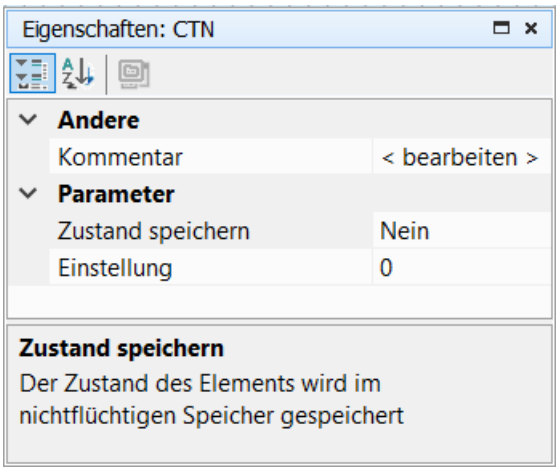
Der Ausgang **Q** ist vom Typ INT. Eine steigende Flanke am Eingang **U** erhöht den Wert am Ausgang **Q** um 1. Eine steigende Flanke am Eingang **D** verringert den Wert am Ausgang **Q** um 1.
Wenn der Eingang **R** = **True** ist, erhält der Ausgang **Q** den Wert ***Einstellung*** am Eingang **N**.



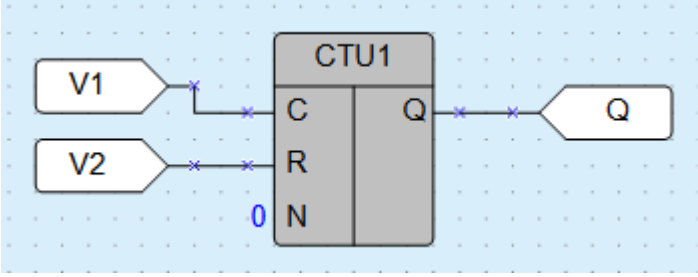
Einstellbereich: 0…65535.



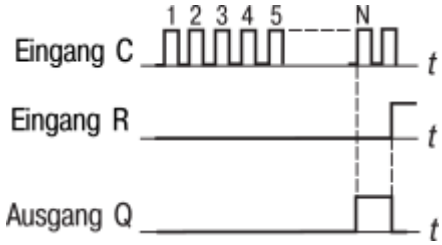
Die Werte der Parameter des Funktionsbausteins können mithilfe eines „In Funktionsbaustein schreiben“-Bausteins festgelegt werden, wie in der obigen Abbildung dargestellt, oder im Eigenschaftenfeld konfiguriert werden:



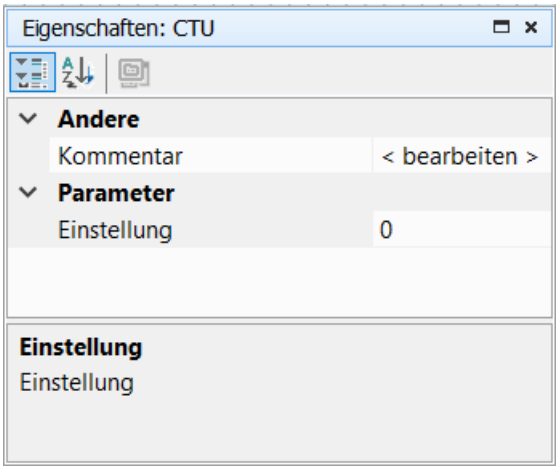
7.2.4.3 Schwellenwertzaehler (CTU)



Der Ausgang **Q** ist vom Typ Boolean. Wenn die Anzahl der am Eingang C gezählten Impulse den am Eingang **N** angegebenen Schwellenwert (***Einstellung***) überschreitet, wird der Ausgang **Q True** und bleibt **True**, bis eine steigende Flanke am Eingang **R** auftritt. Der Eingang R hat eine höhere Priorität als der Eingang **C**.
Die Funktionsweise des Zählers wird im folgenden Diagramm erläutert.



Der Parameter ***Einstellung*** kann im Eigenschaftenfeld festgelegt werden.

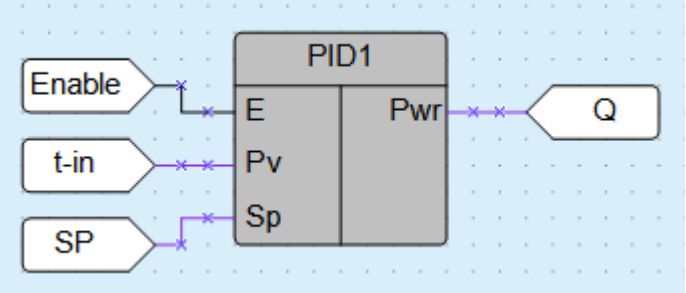


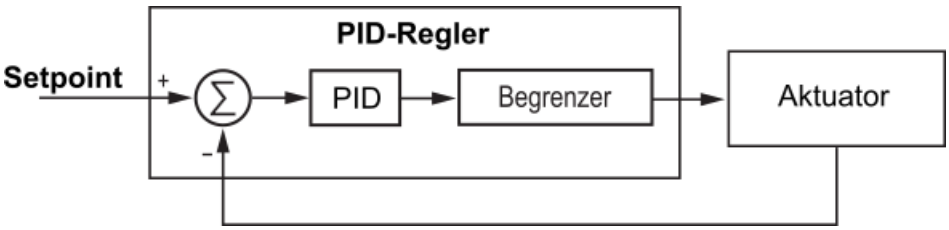
Schwellenwertbereich: 0…65535.

7.2.5 Analog

– *PID-Regler (PID) für Geräte der ersten Generation*
– *PID-Regler (PID) für Geräte der zweiten Generation*

7.2.5.1 PID-Regler (PID) fuer Geraete der ersten Generation





Der Funktionsbaustein **PID** dient zur Implementierung der Proportional-Integral-Differential-Regelung.

Tabelle 7.1 PID-Baustein-Eingänge/Ausgänge

Name	Typ	E/A	Beschreibung	Wert
E	BOOL	I	Steuerung aktivieren (0 = Aus, 1 = Ein). Wenn deaktiviert, nimmt der Parameter ***Pwr*** den Wert des Parameters ***Sicherer Ausgangszustand*** an.	0 – Aus 1 – Ein
Pv	REAL	I	Prozesswert	
Sp	REAL	I	Setpoint	
Pwr	REAL	O	Ausgangsleistung, %	0…100

Tabelle 7.2 PID-Bausteinparameter

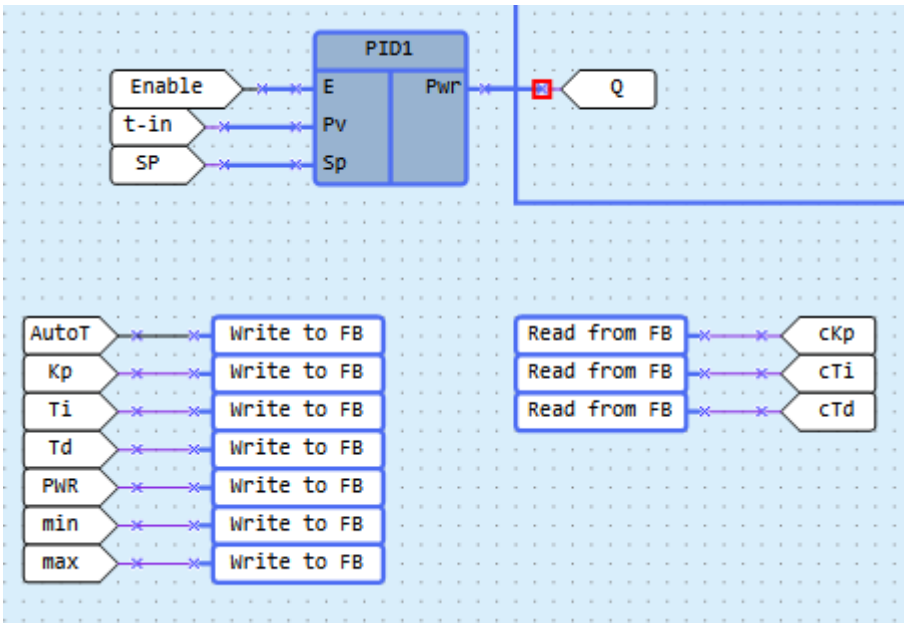
Name	Typ	Beschreibung	Wert	Zugriff		
				Eigen-schaf-ten	Write ToFB	Re-adF-romF-B
Regelungsmo-dus	BOOL	0 – Heizung 1 – Kühlung	0/1	X	X	
Sicherer Ausgangszu-stand	REAL	Ausgangswert bei deaktivierter Regelung, %	0…100	X	X	
Kp	REAL	Proportionalverstär-kung, Multiplikationsfaktor für Proportionalregelung	0…100	X	X	
Ti (s)	REAL	Integralzeit, Zeitkonstante für	–3,402823E +38… 3,402823E+38	X	X	

Name	Typ	Beschreibung	Wert	Zugriff		
				Eigen-schaf-ten	Write ToFB	Re-adF-romF-B
		Integralregelung in Sekunden				
Td (s)	REAL	Vorhaltezeit, Zeitkonstante für Vorhalteregelung in Sekunden	–3,402823E +38… 3,402823E+38	X	X	
Ausgang max.	REAL	Obere Ausgangsgrenze, % (Standard: 80 %)	0…100	X	X	
Ausgang min.	REAL	Untere Ausgangsgrenze, % (Standard 20 %)	0…100	X	X	
AT starten	BOOL	0 – Autotuning stoppen 1 – Autotuning starten	0/1		X	
AT abgeschlos-sen	BOOL	Flag: 0 – Autotuning gestoppt 1 – Autotuning gestartet	0/1			X
Kp berechnet	REAL	Berechnete proportionale Verstärkung	–3,402823E +38… 3,402823E+38			X
Ti berechnet	REAL	Berechnete Integralzeit	–3,402823E +38… 3,402823E+38			X
Td berechnet	REAL	Berechnete Vorhaltezeit	–3,402823E +38… 3,402823E+38			X

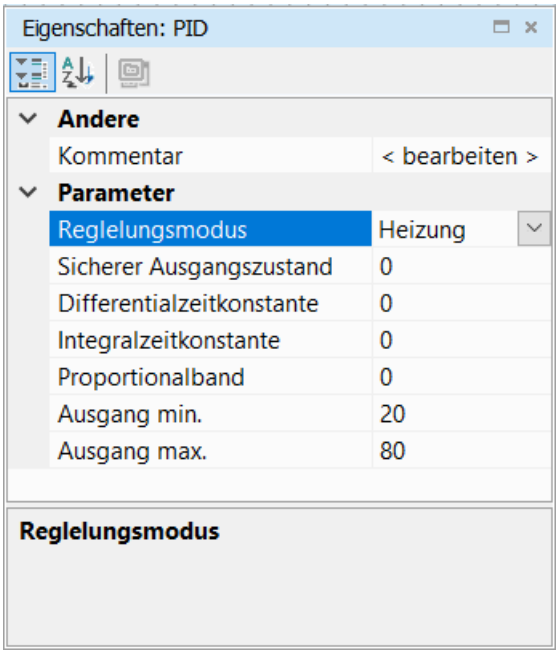
Unter Feinabstimmung eines Regelkreises versteht man die Anpassung seiner Regelparameter (***Kp***, ***Ti***, ***Td***) an die optimalen Werte für die gewünschte Regelreaktion.

Autotuning

Eine programmierbare Schleifenoptimierung kann mit den Blöcken *WriteToFB* und *ReadFromFB* durchgeführt werden.
Zum Schreiben der Parameter verwenden Sie den Baustein WriteToFB oder Eigenschaftenfenster.
Zum Lesen der Parameter verwenden Sie den Baustein ReadFromFB.



Um die automatische Optimierung zu nutzen, fügen Sie dem Schaltprogramm den Block WriteToFB hinzu und setzen Sie die Referenz auf den Parameter ***AT starten*** des PID-Blocks.
Um die automatische Optimierung zu starten, aktivieren Sie die Steuerung (***E*** = **1**) und setzen Sie den Parameter ***AT starten*** = **1**.



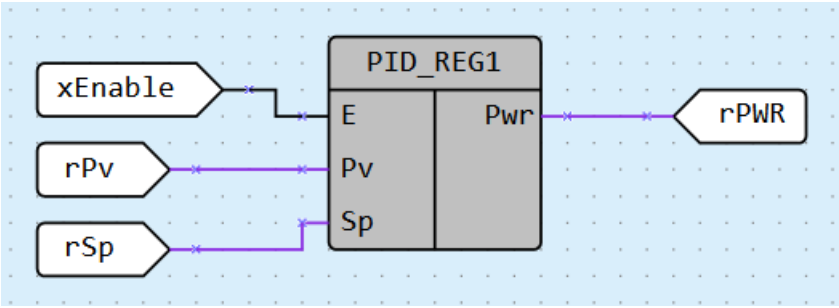
Nach Abschluss der Selbstoptimierung werden die neuen Werte der Parameter ***Kp***, ***Ti*** und ***Td*** berechnet und die Flagge ***AT abgeschlossen*** wird zu **1**.
Wenn ***AT starten*** = **0**, ist auch das Flag ***AT abgeschlossen*** = **0**.
Wenn Sie ***AT starten*** = **0** vor Abschluss der Selbstoptimierung festlegen, wird die Selbstoptimierung gestoppt, die Flagge ***AT abgeschlossen*** wird zu **0** und es werden keine neuen Koeffizienten berechnet.
Während der Selbstoptimierung wird am Ausgang **Pwr** ein durch die Parameter ***Ausgang max.*** und ***Ausgang min.*** begrenztes Testsignal angelegt.

HINWEIS
*Wenn die maximale Verstärkung nicht ausreicht, um den Sollwert zu erreichen, kann die Selbstoptimierung nicht abgeschlossen werden und wird fortgesetzt, bis sie mit **AT starten** = 0 gestoppt wird.*

7.2.5.2 PID-Regler (PID_REG) fuer Geraete der zweiten Generation

HINWEIS
Der PID_REG ist für die PR103-Geräte nicht verfügbar.

.



Der PID-Regler wird zur Implementierung des proportional-integral-derivativen (PID-)Regelgesetzes verwendet.
Die Ausgangsleistung des Reglers wird nach folgender Formel berechnet:

$$MV(t) = \frac{1}{X_p} \cdot \left(e(t) + \frac{1}{T_i} \cdot \int_0^t e(\tau)\, d\tau + T_d \cdot \frac{de(t)}{dt} \right)$$

, wobei
Xp – Proportionalband (Maßeinheiten)
Ti – Nachstellzeit [s]
Td – Vorhaltzeit [s]
e(t) – Regelabweichung
MV(t) – PID-Ausgang

Blockeingänge

Tabelle 7.3 Blockeingänge

Name	Typ	Beschreibung	Werte
E	BOOL	Reglerfreigabe. Bei Deaktivierung wird der Ausgang auf den in der Eigenschaft 'Ausgangsleistung im Aus-Zustand' angegebenen Wert gesetzt.	0 – Aus 1 – Ein
Pv	REAL	Aktueller Wert der Regelgröße	–9999...9999*
Sp	REAL	Sollwert der Regelgröße	

*Wenn ein Wert außerhalb dieses Bereichs an den Eingang angelegt wird, wird der Pwr-Ausgang des Reglers auf den in dem Parameter 'Ausgangsleistung im Aus-Zustand' angegebenen Wert zurückgesetzt.

Blockausgänge

Tabelle 7.4 Blockausgänge

Name	Typ	Beschreibung	Werte
Pwr	REAL	Ausgangsleistung des Reglers	

Eingänge verfügbar via Schreiben in FB

Tabelle 7.5 Eingänge verfügbar via Schreiben in FB

Name	Typ	Beschreibung	Werte
Reglelungs-modus	BOOL	Reglerbetriebsarten: – 'Heizer'-Modus: Wird zur Regelung von Stellgliedern verwendet, die die Regelgröße erhöhen – 'Kühler'-Modus: Wird zur Regelung von Stellgliedern verwendet, die die Regelgröße verringern	0 – Heizer 1 – Kühler
AT starten	BOOL	Steuerung der Regler-Selbsteinstellung (Auto-Tuning)	0 – Selbsteinstellung aus 1 – Selbsteinstellung gestartet
Proportional-band	REAL	Proportionalband des PID-Reglers, Maßeinheiten	0...9999
Integralzeit-konstante	REAL	Nachstellzeit, Sek.	0...4000
Differential-zeitkonstante	REAL	Vorhaltzeit, Sek.	0...4000
Sicherer Ausgangszu-stand	REAL	Zustand des Pwr-Ausgangs, wenn Eingang E 0 ist	-9999...9999
Ausgang max.	REAL	Obere Grenze der Ausgangsleistung	-9999...9999
Ausgang min.	REAL	Untere Grenze der Ausgangsleistung	-9999...9999
Todeszone	REAL	Toter Bereich um den Sollwert/2, in dem die Reglerausgangsleistung unverändert bleibt, Maßeinheiten	
Ausgangstyp	BOOL	Auswahl des Regler-Ausgangssignals	0 – Leistung 1 – Diskretes Stellglied (CRV)
Verfahrzeit	REAL	Zeit für den Ventilhub von ganz geschlossen bis ganz geöffnet, Sek.	5...999**
Minimale Impulslänge	REAL	Minimale Stellimpulsdauer für das Ventil, Sek.	0.001...100
Spielausg-leichszeit	REAL	Erfassungszeit für Ventilspiel, Sek.	0...10
Sollwertän-derungsrate	REAL	Sollwertänderungsgeschwin-digkeit, Maßeinheiten/Sek.	0 – Funktion deaktiviert 0...9999
Manuelle Bedienung des Reglers	BOOL	Flag zum Umschalten des Reglers in den Handbetrieb	0 – Automatikbetrieb 1 – Handbetrieb

Name	Typ	Beschreibung	Werte
Manuelle Ausgangs-steuerung "Mehr"	BOOL	Im Handbetrieb: Befehl zum Senden eines "Erhöhen"-Signals an den Blockausgang	
Manuelle Ausgangs-steuerung "Weniger"	BOOL	Im Handbetrieb: Befehl zum Senden eines "Vermindern"-Signals an den Blockausgang	
Anfangswert der Regelgröße für PID-Autotuning (Pv_0)	REAL	Wert der Regelgröße, der am Eingang Pv anliegt, wenn keine Leistung angelegt wird (Objekt im Ruhezustand). Dieser Parameter ist für eine korrekte Selbsteinstellung erforderlich.	-9999...9999

**Um eine neue Gesamthubzeit für das Ventil zu übernehmen, starten Sie den Block neu, indem Sie das Signal am Eingang E umschalten.

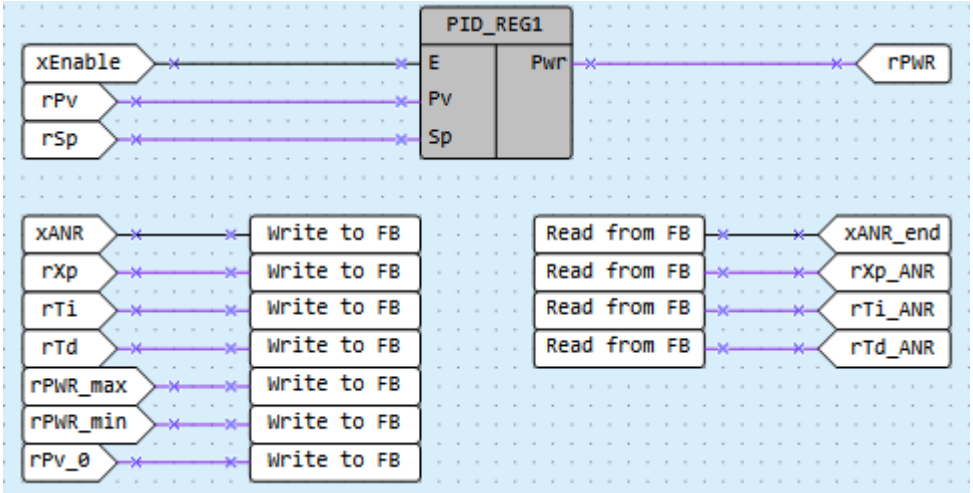
Ausgänge verfügbar via Lesen aus FB

Tabelle 7.6 Ausgänge verfügbar via Lesen aus FB

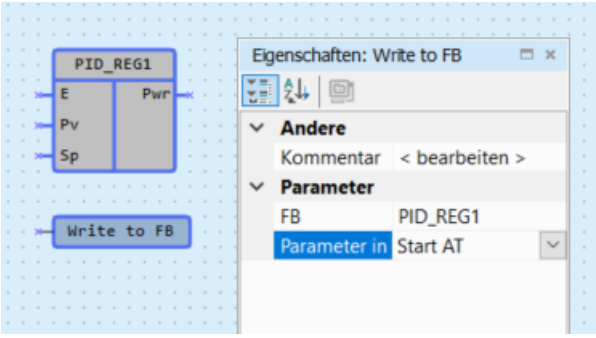
Name	Typ	Beschreibung	Werte
AT beendet	BOOL	Flag für Abschluss der Selbsteinstellung	0 – AT nicht abgeschlossen 1 – AT abgeschlossen
Berechnetes Proportional-band	REAL	Nach der Selbsteinstellung berechnetes Proportionalband, Maßeinheiten	0...9999
Berechnete Integralzeit-konstante	REAL	Nach der Selbsteinstellung berechnete Nachstellzeit, Sek.	0...4000
Berechnete differentielle Zeitkonstan-te	REAL	Nach der Selbsteinstellung berechnete Vorhaltzeit, Sek.	0...4000
"Mehr"	BOOL	Ventil-Öffnen-Signal	
"Weniger"	BOOL	Ventil-Schließen-Signal	

Selbsteinstellung (Auto-Tuning)

Die Selbsteinstellung des PID-Reglers wird mit *Lesen aus und Schreiben in FB*-Blöcken durchgeführt.



Um die Selbsteinstellung zu starten, fügen Sie einen **Write to FB**-Block hinzu und binden ihn an die Variable **AT starten** des PID-Reglers.



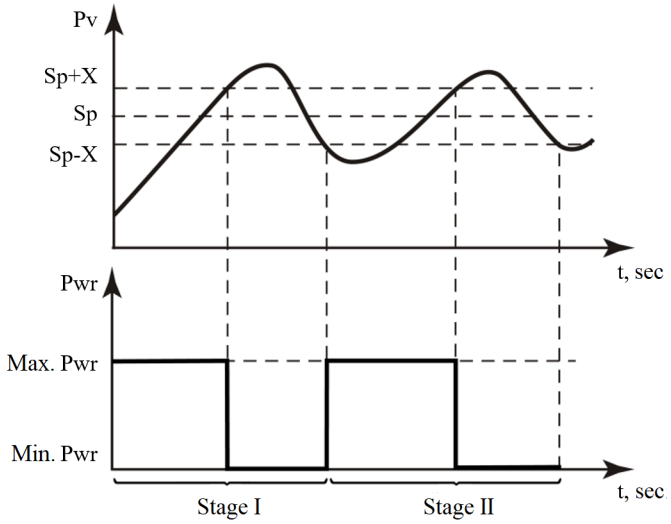
Werte für andere FB-Parameter können wie in der Abbildung oben gezeigt mit dem **Write to FB**-Block gesetzt oder im Eigenschaften-Bedienfeld konfiguriert werden.
Verwenden Sie **Read to FB**-Blöcke, um die Werte der Parameter **Berechnetes Proportionalband**, **Berechnete Integralzeitkonstante**, **Berechnete differentiale Zeitkonstante** und **AT beendet** auszulesen.
Um die Selbsteinstellung zu starten, legen Sie eine logische "1" an den Eingang **E** an.
Nach Abschluss des Selbsteinstellungsprozesses sind neue Parameterwerte zum Auslesen verfügbar: **Berechnetes Proportionalband**, **Berechnete Integralzeitkonstante** und **Berechnete differentiale Zeitkonstante**. Der Parameter **AT beendet** wird auf logisch Eins gesetzt.

VORSICHT
Das Flag für den Abschluss der Selbsteinstellung bleibt nur für einen Zyklus auf logisch Eins.

Wenn der Eingang **AT starten** vor Abschluss der Einstellung auf logisch Null zurückgesetzt wird, wird der Prozess angehalten, das Abschlussflag wird nicht gesetzt und neue Parameterwerte werden nicht berechnet.

VORSICHT
Für eine korrekte Selbsteinstellung muss der Parameter **Anfangswert der Regelgröße für PID-Autotuning** gesetzt werden. Er sollte dem Wert der Regelgröße entsprechen, der bei Null-Stellung des Stellglieds vorliegt (Objekt im Ruhezustand).

Während des Einstellvorgangs wird eine Testleistung, die durch die Parameterwerte **Ausgang max.** und **Ausgang min.** begrenzt ist, an den PID-Reglerausgang angelegt.
Ablauf der Selbsteinstellung für den 'Heizer'-Modus:



1. Wenn der Istwert um mehr als *X = 0,04*(Sp-Pv_0)* unter dem Sollwert liegt, wird maximale Leistung (gemäß den Einstellungen) an den Blockausgang angelegt.
2. Sobald der Istwert um mehr als *X = 0,04*(Sp-Pv_0)* über den Sollwert steigt, wird minimale Leistung an den Blockausgang angelegt.
3. Die Schritte 1 und 2 werden einmal wiederholt.
4. Die berechneten PID-Reglerparameter werden an die entsprechenden Ausgänge gesendet und das Abschlussflag wird gesetzt (für einen Zyklus).

Wenn die maximale Leistungseinstellung nicht ausreicht, um den Sollwert zu erreichen, wird der Selbsteinstellungsprozess nicht abgeschlossen, bis er manuell zurückgesetzt wird.

Arbeiten mit digitalen Stellgliedern (CRV)

Um mit einem diskreten Stellglied zu arbeiten, setzen Sie den Wert im entsprechenden Parameter (Ausgangstyp) Geben Sie den Wert für „digitalen Stellgliedern“ an oder schreiben Sie bei der Bearbeitung dieses Parameters über „Schreiben in FB“ den Wert „1“.Für den korrekten Betrieb müssen die Ventilparameter Vollhubzeit, Minimale Impulszeit und Spiel-Abtastzeit eingestellt werden.
Der Zugriff auf die Steuersignale (High/Low) erfolgt über die entsprechenden Leseausgänge des FB-Controller-Blocks.In diesem Modus zeigt der Ausgang Pwr die virtuelle Ventilposition an (diese kann von der tatsächlichen Position abweichen).Wenn die Logik „1“ am Eingang E entfernt wird, bleibt das Ventil in seiner aktuellen Position.Für eine schnelle Einrichtung empfehlen wir die Verwendung vorgefertigter Makros aus dem Komponenten-Manager:

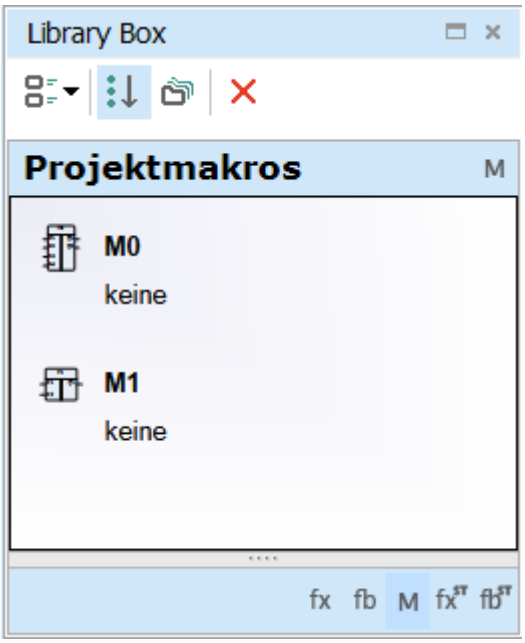
– Für die Leistungsregelung - **PID AT_**.
– Makro mit voller Funktionalität im Leistungsregelungsmodus - **PID AT_F_**.
– Zur Steuerung des diskreten CRV – PID-Ventilblocks.
– Makro mit voller Funktionalität im diskreten CRV-Steuerungsmodus— **PID Valve_F_**.

Manueller Modus

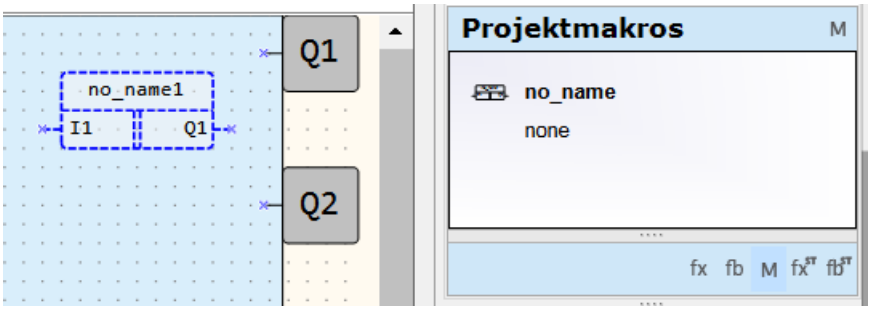
Betrieb im Leistungsregelungsmodus: Wenn der Regler in den manuellen Modus geschaltet wird (Manueller Eingang des Reglers auf 1 gesetzt), wird der berechnete Leistungswert gespeichert.
Betrieb im diskreten CRV-Regelungsmodus: Wenn der Regler in den manuellen Modus geschaltet wird (Manueller Eingang des Reglers auf 1 gesetzt), wird die berechnete Ventilposition gespeichert und die Eingänge zum manuellen Öffnen/Schließen des Ventils stehen zur Verfügung.

7.3 Projektmakros

Der Abschnitt ***Projektmakros*** enthält Makros, die *vom Benutzer erstellt* oder mit dem *Komponenten-Manager* aus der Online-Datenbank heruntergeladen wurden.



Um einem Projekt ein Makro hinzuzufügen, ziehen Sie das Makro per Drag & Drop aus der *Bibliothek* in den Arbeitsbereich.

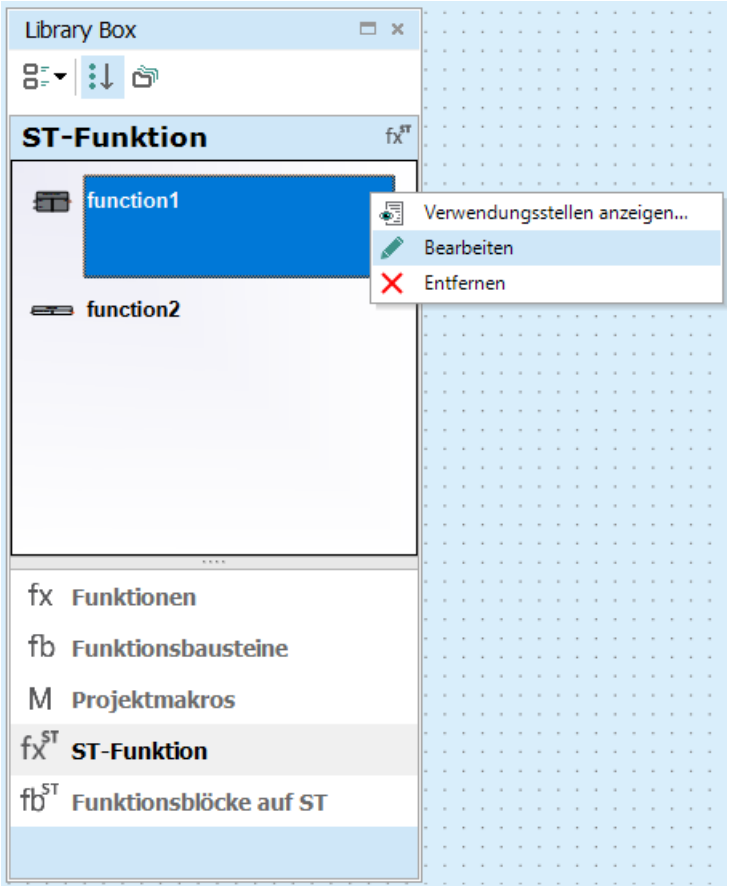


Um das Projektmakro im separaten Arbeitsbereich zum Bearbeiten zu öffnen, markieren Sie es im Arbeitsbereich oder in der Bibliothek und nutzen Sie den Punkt Makro bearbeiten im Makro-Kontextmenü.
Um das Makro aus der Bibliotheksbox zu entfernen, wählen Sie das Makro aus und klicken Sie in der Bedienfeldsymbolleiste auf das Symbol ✕.
Einzelheiten zur Erstellung, Entwicklung und Handhabung von Makros finden Sie im Abschnitt *Makroentwicklung*.

7.4 ST-Funktionen

HINWEIS
Für Geräte der Baureihen PR100(M02), PR102, PR200, PR103, PR205, PR225 und SMI200 steht die Erstellung benutzerdefinierter Funktionen in der Sprache ST zur Verfügung.

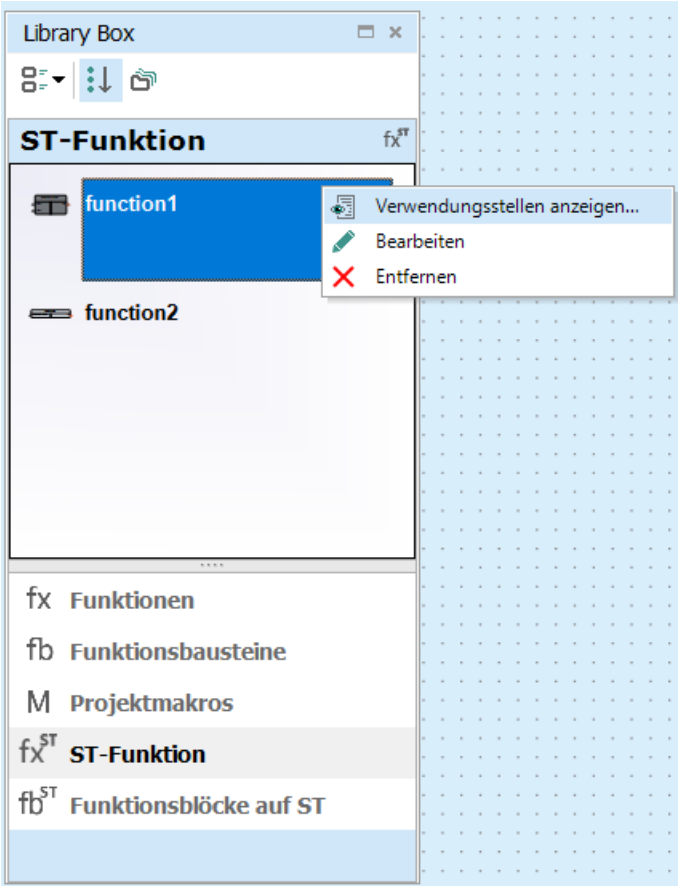
Wenn Sie in Ihrem Projekt *ST-Funktionen erstellt* haben, stehen diese in der Bibliotheksbox zur Verfügung.



Verwendungsstellen

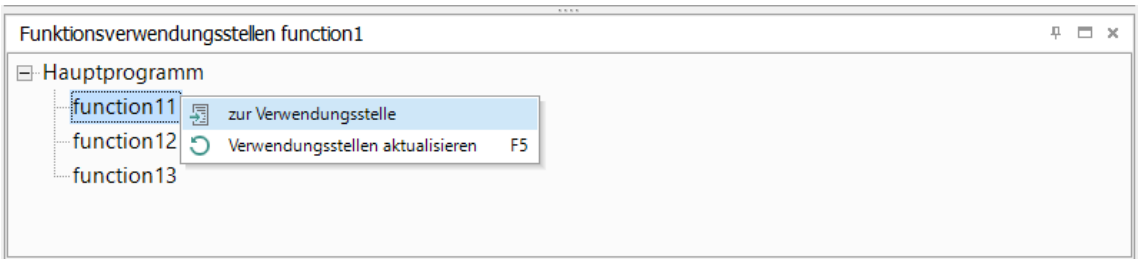
So zeigen Sie alle Stellen an, an denen die Funktion verwendet wird:

1. Klicken Sie mit der rechten Maustaste auf den Funktionsnamen.
2. Wählen Sie das Element ***Verwendungatellen anzeigen…***.



Unten im Fenster wird das Fenster **Funktionsverwendungsorte** geöffnet. Es zeigt an, wo die Funktion im Diagramm und im Funktionseditor verwendet wird.

3. Klicken Sie mit der rechten Maustaste auf die Zeile, die angibt, wo die Funktion verwendet wird.
4. Wählen Sie den Eintrag **Zur Verwendungsstelle**.



Der Fokus verschiebt sich dorthin, wo die Funktion im Diagramm oder im Funktionseditor verwendet wird.

HINWEIS
Doppelklick führt zum selben Ergebnis.

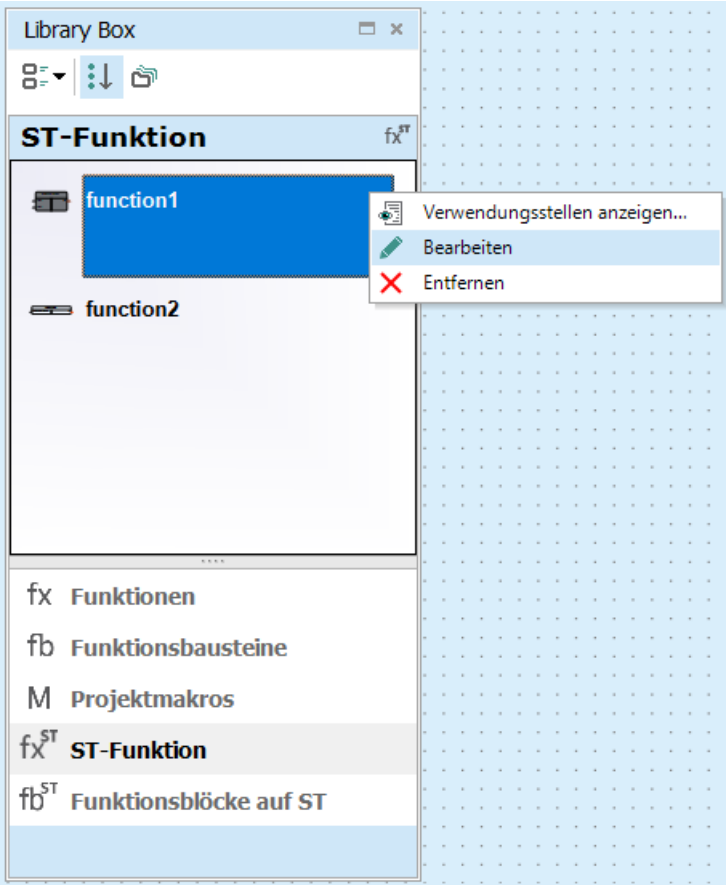
Wenn sich die Stellen, an denen Funktionen verwendet werden, während der Arbeit mit dem Programm geändert haben, sollten Sie das Bedienfeld **Funktionsverwendungsstellen** aktualisieren:

1. Klicken Sie mit der rechten Maustaste auf eine beliebige Zeile des Bedienfelds.
2. Wählen Sie den Eintrag ***Verwendungsstellen aktualisieren*** aus.

Zum Funktionseditor

So gelangen Sie zum *Funktionseditor*:

1. Klicken Sie mit der rechten Maustaste auf den Funktionsnamen.
2. Wählen Sie den Eintrag ***Bearbeiten*** aus.

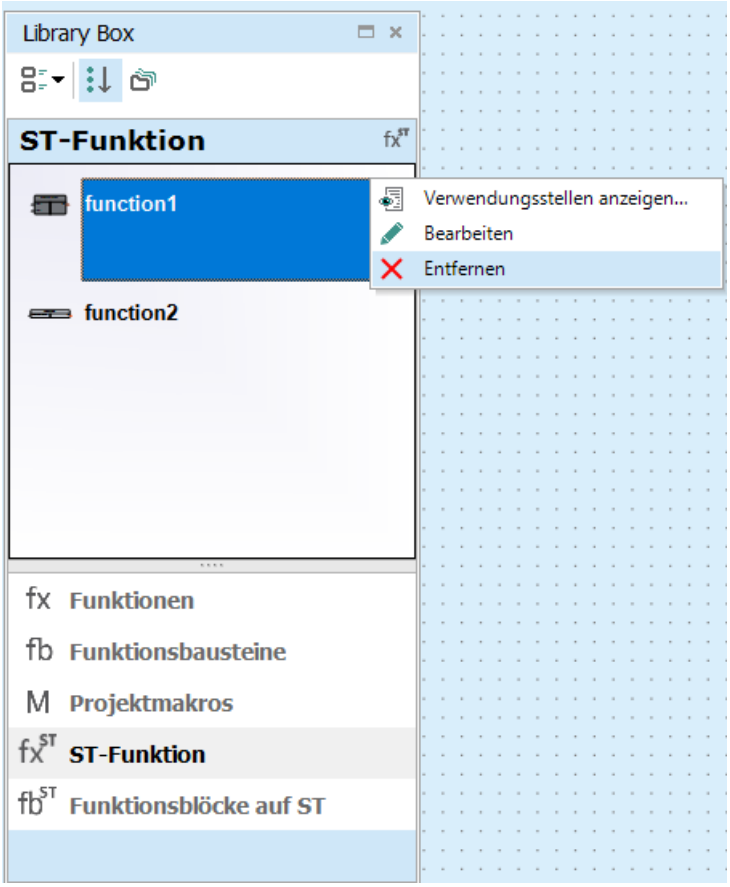


Der *Funktionseditor* wird geöffnet.

Funktion entfernen

So entfernen Sie eine Funktion aus einem Projekt:

1. Klicken Sie mit der rechten Maustaste auf den Funktionsnamen.
2. Wählen Sie den Punkt **Entfernen** aus.

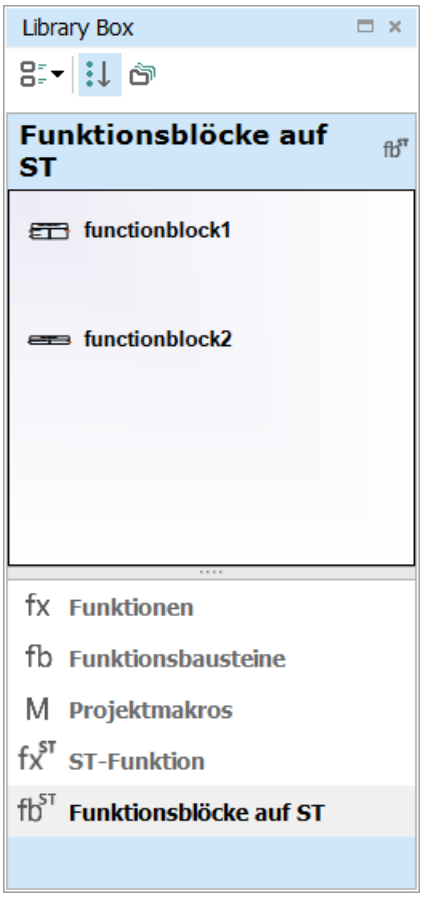


ACHTUNG
Wird die Funktion im Diagramm und/oder in anderen Funktionen verwendet, kann es durch das Löschen zu Kompilierungsfehlern kommen.

7.5 ST-Funktionsblock

HINWEIS
Für Geräte der Baureihen PR100(M02), PR102, PR200, PR103, PR205, PR225 und SMI200 steht die Erstellung benutzerdefinierter Funktionsbausteine in der Sprache ST zur Verfügung.

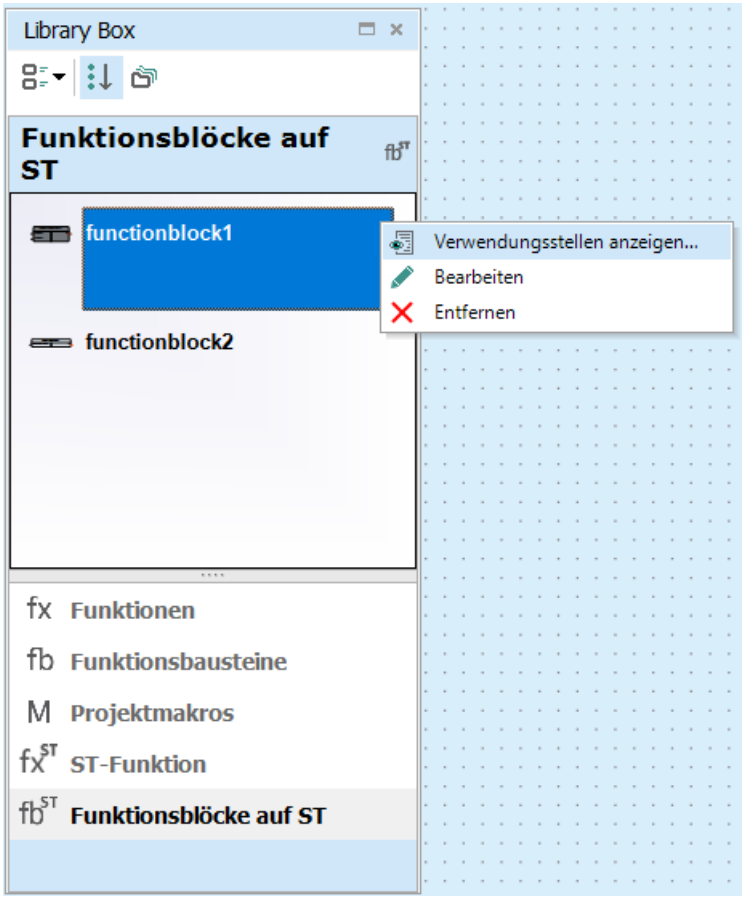
Wenn im Projekt ST-Funktionsbausteine erstellt werden, stehen diese in der Komponentenbibliothek zur Verfügung.



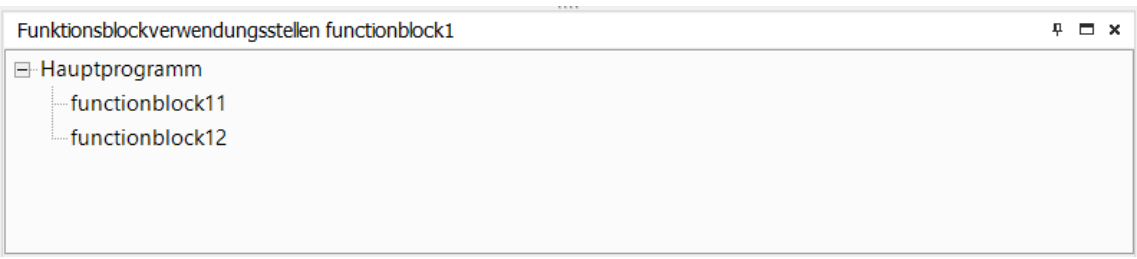
Verwendungsstellen

So zeigen Sie alle Stellen an, an denen die Funktion verwendet wird:

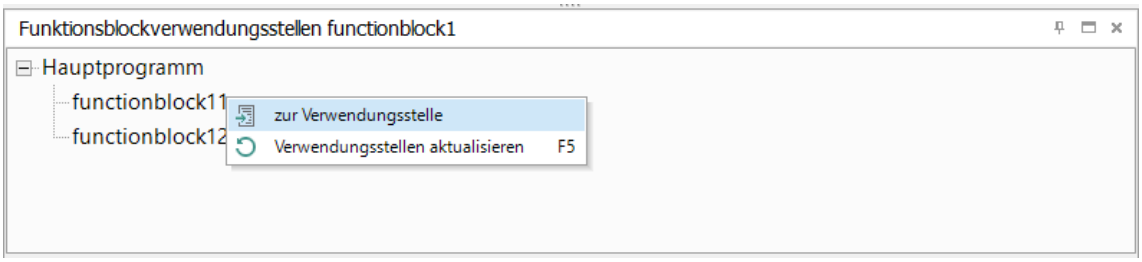
1. Klicken Sie mit der rechten Maustaste auf den Namen des Funktionsblocks.
2. Wählen Sie den Eintrag **Verwendungsstellen anzeigen…** aus.



Unten im Fenster wird das Bedienfeld **Funktionsblockverwendungsstellen** geöffnet. Es zeigt an, wo die Funktion im Hauptprogramm und im Editor verwendet wird.



3. Wählen Sie den Eintrag **Zur Verwendungsstelle**



Der Fokus verschiebt sich dorthin, wo der Funktionsblock im Diagramm oder im Editor verwendet wird.

HINWEIS
Doppelklick führt zum selben Ergebnis.

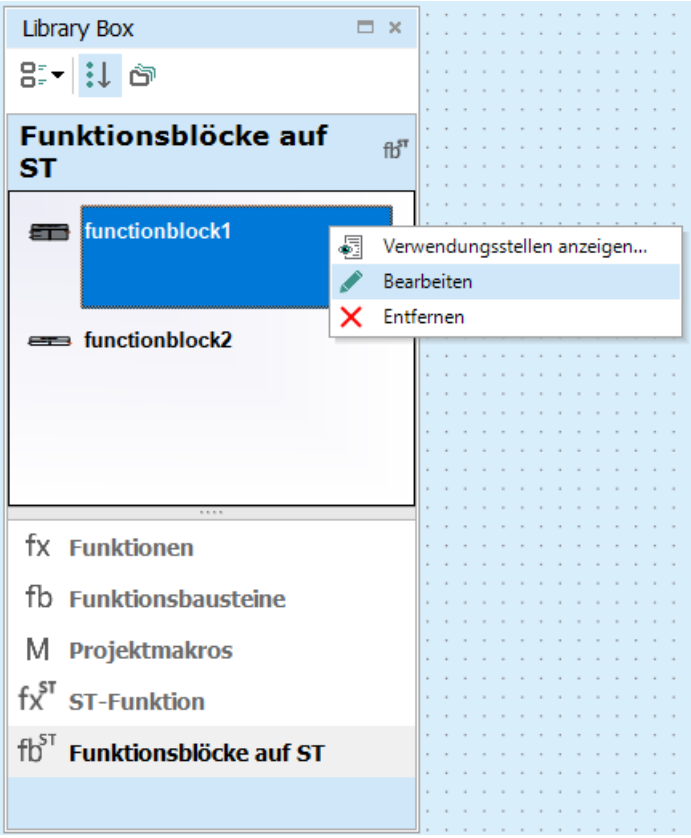
Wenn sich die Stellen, an denen Funktionsblöcke verwendet werden, während der Arbeit mit dem Programm geändert haben, sollten Sie das Fenster **Funktionsblock-Verwendungsstelle** aktualisieren:

1. Klicken Sie mit der rechten Maustaste auf eine beliebige Zeile des Bedienfelds.
2. Wählen Sie den Eintrag **Verwendungsstellen aktualisieren** aus.

Zum Funktionsblock-Editor

So gelangen Sie zum *Funktionsblock-Editor*:

1. Klicken Sie mit der rechten Maustaste auf den Funktionsblocknamen.
2. Wählen Sie den Punkt **Bearbeiten** aus.

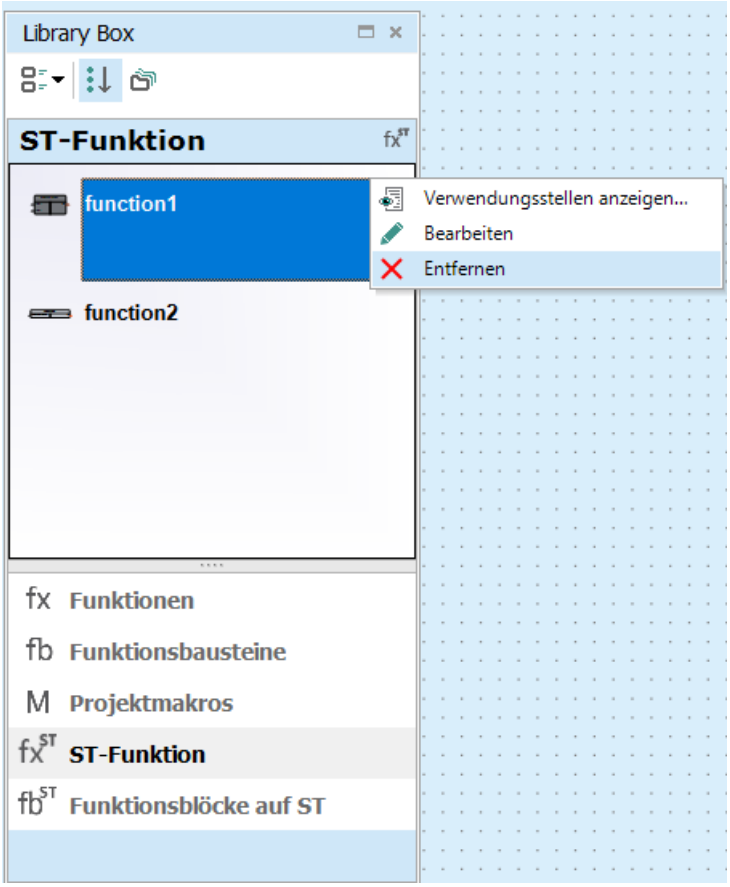


Der *Funktionsblock-Editor* wird geöffnet.

Funktionsblock entfernen

So löschen Sie einen Funktionsblock:

1. Klicken Sie mit der rechten Maustaste auf den Funktionsblocknamen.
2. Wählen Sie den Punkt **Entfernen** aus.



ACHTUNG
Wird der Funktionsbaustein im Hauptprogramm und/oder in anderen Funktionen verwendet, kann es beim Löschen zu Kompilierungsfehlern kommen.

7.6 Display-Elemente

Ist der Arbeitsbereich mit Displayform aktiv, stehen in der Bibliothek nur Display-Elemente zur Verfügung. Mit diesen Blöcken lassen sich die auf dem Gerätedisplay angezeigten Informationen steuern. Die Display-Elemente lassen sich per Drag & Drop innerhalb der Displayform platzieren. Folgende Elemente stehen zur Verfügung:

– *Text-Box*
– *I/O-Box (INT/REAL)*
– *I/O-Box (BOOL)*
– *Dynamische Box*
– *ComboBox*

Verwenden Sie das Eigenschaftenfeld, um ein Element anzupassen.
Gemeinsame Parameter für alle Elemente:

– ***Koordinate X*** – Bestimmt die Position des ersten (linken) Zeichenplatzhalters des Elements auf der X-Achse.
– ***Koordinate Y*** – Bestimmt die Position des ersten (linken) Zeichenplatzhalters des Elements auf der Y-Achse.
– Ermöglicht die Festlegung der Koordinaten als konstant oder variabel. Um eine von einer Variablen abhängige Koordinate zu verwenden, wählen Sie die Koordinate aus und öffnen Sie die Liste rechts neben dem Eingabefeld.
  – ***Konstant*** – Legt die Koordinaten direkt im Eigenschaftenfeld oder per Drag & Drop im Formular fest.
  – ***Variable*** – Ermöglicht die Auswahl einer INT-Variable zur dynamischen Steuerung der Koordinaten.

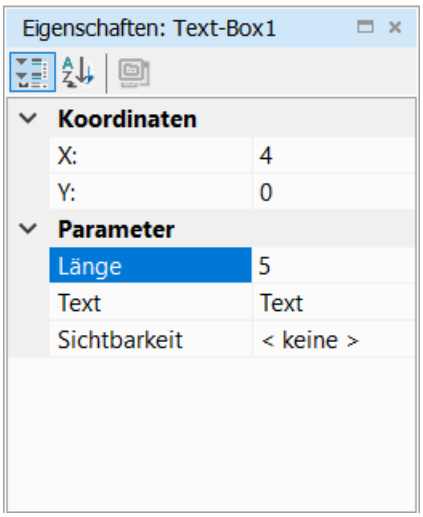
– ***Länge*** – Definiert die Anzahl der reservierten Zeichen. Das Display-Element belegt eine Zeile und kann eine Länge von 1 bis 16 Zeichen haben.

7.6.1 Text-Box

Element **Text-Box** Entwickelt, um einen Textblock in einer Zeile auf dem Bildschirm zu platzieren.

Eigenschaften einrichten

Um eine korrekte Anzeige auf dem Gerätebildschirm zu gewährleisten, sollten Sie die Standortkoordinaten angeben und den Text im Eigenschaftenfenster eingeben.

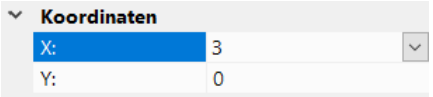


Koordinaten

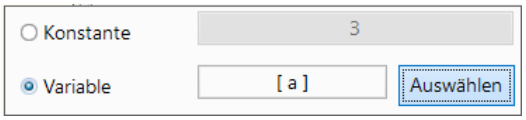
Um das erste Symbol eines Elements entlang der X- und Y-Achse zu positionieren, müssen Koordinaten angegeben werden. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse von links nach rechts von 0 bis 15 (abhängig von der Eigenschaft **Länge**).
– Entlang der Y-Achse von oben nach unten (abhängig von der Anzahl der im Display-Editor eingestellten Bildschirmzeilen).

Der Koordinatenwert (Ganzzahl) kann im Dropdown-Menü in der Koordinatenzeile eingestellt werden (X und Y sind gleich konfiguriert):



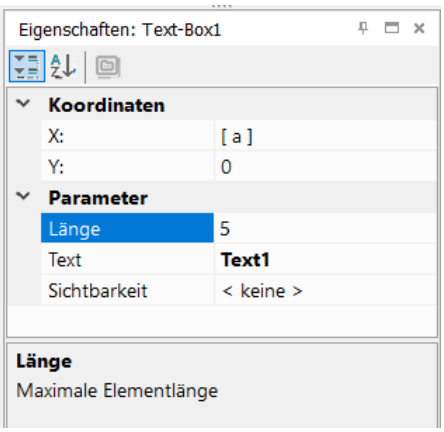
Im Dropdown-Menü können Sie die Methode zur Angabe der Koordinaten auswählen: Konstante oder Variable. Um einen festen Standort für ein Element auszuwählen, geben Sie den Koordinatenwert im Eingabefeld an.
Um eine durchgehende Linie zu erstellen, wählen Sie die Option **Variable** aus und drücken Sie **Auswählen**. Es wird sich *Variablentabelle 6* öffnen. Darin sollten Sie eine Variable vom Typ Integer auswählen, die den Wert der Koordinate bestimmt, und die Schaltfläche **OK** drücken. Die ausgewählte Variable wird im Eingabefeld angezeigt:



Parameter

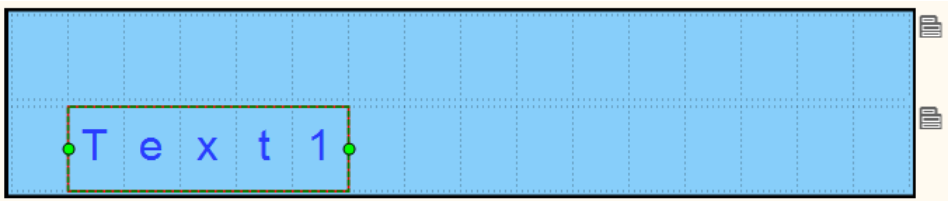
– **Text** — Geben Sie den Text ein, der auf dem Gerätebildschirm angezeigt werden soll. Die Textlänge darf den für den Parameter **Länge** angegebenen Wert nicht überschreiten.

– **Länge** — Legen Sie die maximale Anzahl der anzuzeigenden Blockzeichen fest.
– **Sichtbarkeit** — Weisen Sie dem Parameter eine boolesche Variable zu, die die Sichtbarkeit des Elements bestimmt.



Anzeige

Der Display-Editor zeigt an, wie das Element **Text-Box** auf dem Gerätebildschirm aussehen wird.

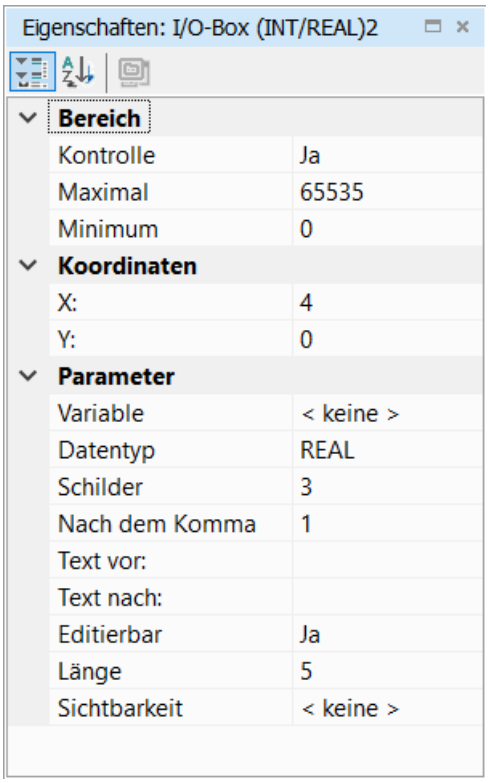


7.6.2 I/O-Box (INT/REAL)

Das Element **I/O-Box (INT/REAL)** ermöglicht die Anzeige und Eingabe numerischer Werte. Der Datentyp kann entweder INT für Ganzzahlen oder REAL für Dezimalzahlen sein.

Eigenschaften einrichten

Damit das Element korrekt funktioniert, müssen Sie die Standortkoordinaten angeben und die Eingabe-/Ausgabeparameter im Eigenschaftenfenster konfigurieren.



Bereich

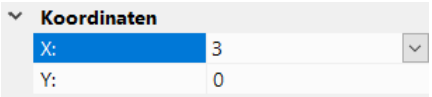
- **Kontrolle** — Aktivieren Sie die Kontrolle, um den vom Benutzer eingegebenen Wert auf den festgelegten Bereich zu begrenzen. Die Einschränkung gilt nicht für den Wert, der während der Programmausführung erzeugt wird.
- **Maximal** — Legen Sie den Maximalwert des Eingabeparameters fest.
- **Minimum** — Legen Sie den Minimalwert des Eingabeparameters fest.

Koordinaten

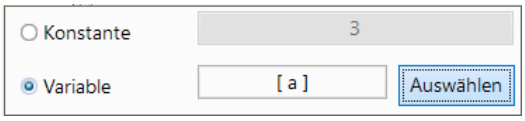
Um das erste Symbol eines Elements entlang der X- und Y-Achse zu positionieren, müssen Koordinaten angegeben werden. Die Koordinaten beider Achsen beginnen bei 0:

- Entlang der X-Achse von links nach rechts von 0 bis 15 (abhängig von der Eigenschaft **Länge**).
- Entlang der Y-Achse von oben nach unten (abhängig von der Anzahl der im Display-Editor eingestellten Bildschirmzeilen).

Der Koordinatenwert (Ganzzahl) kann im Dropdown-Menü in der Koordinatenzeile eingestellt werden (X und Y sind gleich konfiguriert):



Im Dropdown-Menü können Sie die Methode zur Angabe der Koordinaten auswählen: Konstante oder Variable. Um einen festen Standort für ein Element auszuwählen, geben Sie den Koordinatenwert im Eingabefeld an.
Um eine durchgehende Linie zu erstellen, wählen Sie die Option **Variable** aus und drücken Sie **Auswählen**. Es wird sich *Variablentabelle* öffnen. Darin sollten Sie eine Variable vom Typ Integer auswählen, die den Wert der Koordinate bestimmt, und die Schaltfläche **OK** drücken. Die ausgewählte Variable wird im Eingabefeld angezeigt:



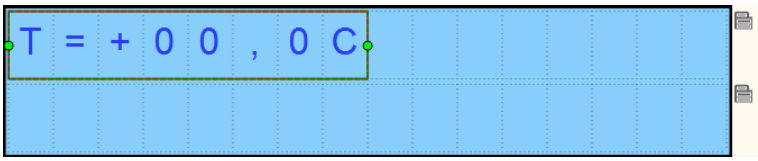
Parameter

- **Variable** — Binden Sie das Element an eine Projektvariable vom Typ Ganzzahl oder Gleitkommazahl. Wählen Sie die Variable über „…“ aus der *Variablentabelle 6* aus.
- **Datentyp** — Legen Sie den Variablentyp fest: Integer oder Real. Wenn noch keine Variable gebunden ist, muss der Typ manuell festgelegt werden.
- **Stellen** — Legen Sie die Gesamtzahl der anzuzeigenden Stellen vor und nach dem Komma fest.
- **Nach dem Komma** — Legen Sie die Anzahl der Nachkommastellen fest. Es sind bis zu 6 Stellen oder die automatische Ermittlung möglich.
- **Text vor** — Geben Sie den Text ein, der links neben dem numerischen Wert angezeigt wird.
- **Text nach** — Geben Sie den Text ein, der rechts neben dem Zahlenwert angezeigt wird.
- **Editierbar** — Aktivieren Sie diese Option, damit der Wert der verknüpften Variable über die Tasten auf der Vorderseite des Geräts geändert werden kann.
- **Länge** — Legen Sie die maximale Anzahl der anzuzeigenden Zeichen fest.
- **Sichtbarkeit** — Weisen Sie dem Parameter eine boolesche Variable zu, die die Sichtbarkeit des Elements bestimmt.

Parameter	
Variable	< keine >
Datentyp	REAL
Schilder	3
Nach dem Komma	1
Text vor:	**T=**
Text nach:	**C**
Editierbar	Ja
Länge	5
Sichtbarkeit	< keine >

Anzeige

Der Display-Editor zeigt an, wie das Element **I/O-Box (INT/REAL)** auf dem Gerätebildschirm aussehen wird.

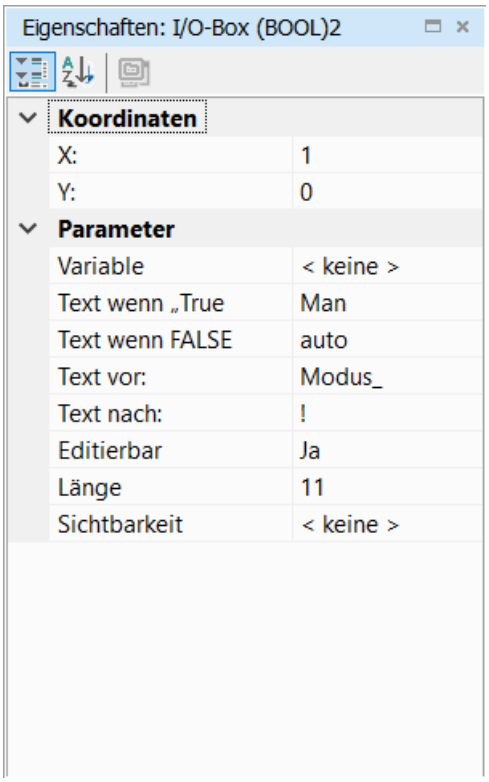


7.6.3 I/O-Box (BOOL)

Element **I/O-Box (BOOL)** ist dazu ausgelegt, den Wert einer booleschen Variable auf dem Bildschirm anzuzeigen oder vom Bildschirm abzulesen.

Eigenschaften einrichten

Damit das Element korrekt funktioniert, müssen Sie die Standortkoordinaten angeben und die Eingabe-/Ausgabeparameter im Eigenschaftenfenster konfigurieren.

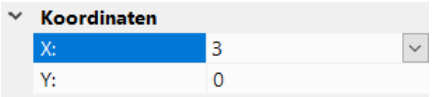


Koordinaten

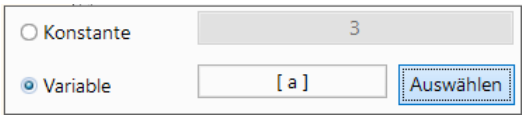
Um das erste Symbol eines Elements entlang der X- und Y-Achse zu positionieren, müssen Koordinaten angegeben werden. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse von links nach rechts von 0 bis 15 (abhängig von der Eigenschaft **Länge**).
– Entlang der Y-Achse von oben nach unten (abhängig von der Anzahl der im Display-Editor eingestellten Bildschirmzeilen).

Der Koordinatenwert (Ganzzahl) kann im Dropdown-Menü in der Koordinatenzeile eingestellt werden (X und Y sind gleich konfiguriert):



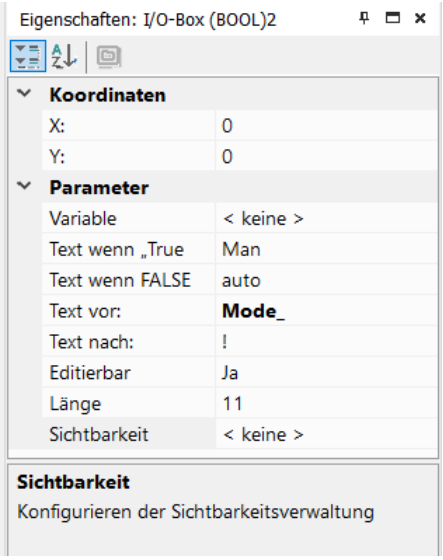
Im Dropdown-Menü können Sie die Methode zur Angabe der Koordinaten auswählen: Konstante oder Variable. Um einen festen Standort für ein Element auszuwählen, geben Sie den Koordinatenwert im Eingabefeld an.
Um eine durchgehende Linie zu erstellen, wählen Sie die Option **Variable** aus und drücken Sie **Auswählen**. Es wird sich *Variablentabelle* öffnen. Darin sollten Sie eine Variable vom Typ Integer auswählen, die den Wert der Koordinate bestimmt, und die Schaltfläche **OK** drücken. Die ausgewählte Variable wird im Eingabefeld angezeigt:



Parameter

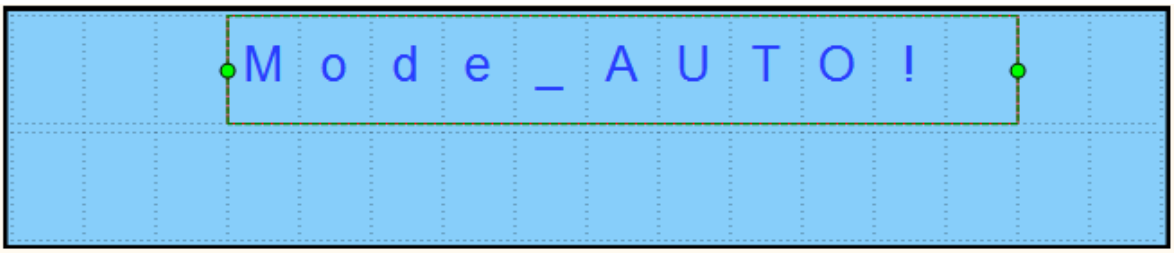
– **Variable** — Binden Sie das Element an eine Projektvariable vom Typ BOOL. Wählen Sie die Variable über „…“ aus der *Variablentabelle 6* aus.
– **Text wenn TRUE** — Geben Sie den Text ein, der angezeigt wird, wenn der Variablenwert TRUE ist.
– **Text wenn FALSE** — Geben Sie den Text ein, der angezeigt wird, wenn der Variablenwert FALSE ist

- **Text vor** — Geben Sie den Text ein, der links neben dem Ausgabewert angezeigt wird.
- **Text nach** — Geben Sie den Text ein, der rechts neben dem Ausgabewert angezeigt wird.
- **Editierbar** — Aktivieren Sie diese Option, damit der Wert der verknüpften Variable über die Tasten auf der Vorderseite des Geräts geändert werden kann.
- **Länge** — Legen Sie die maximale Anzahl der anzuzeigenden Zeichen fest.
- **Sichtbarkeit** — Weisen Sie dem Parameter eine boolesche Variable zu, die die Sichtbarkeit des Elements bestimmt.



Anzeige

Der Display-Editor zeigt an, wie das Element **I/O-Box (BOOL)** auf dem Gerätebildschirm aussehen wird.

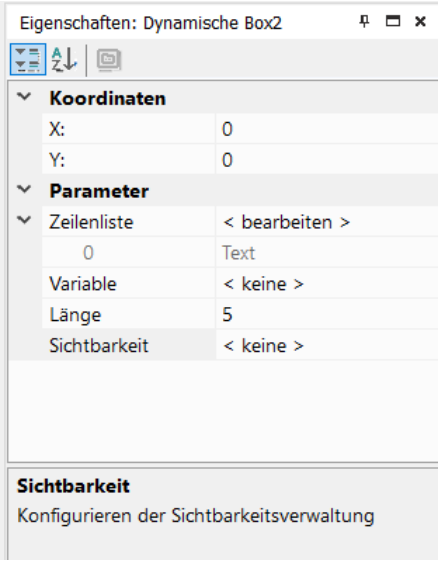


7.6.4 Dynamische Box

Element **Dynamische Box** ist so konzipiert, dass je nach Wert der verknüpften Variablen eine von mehreren Textzeilen auf dem Gerätebildschirm angezeigt wird.

Eigenschaften einrichten

Um eine korrekte Anzeige auf dem Gerätebildschirm zu gewährleisten, sollten Sie die Standortkoordinaten angeben und den Text im Eigenschaftenfenster eingeben.

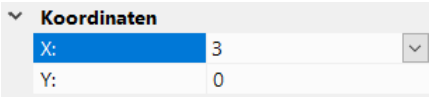


Koordinaten

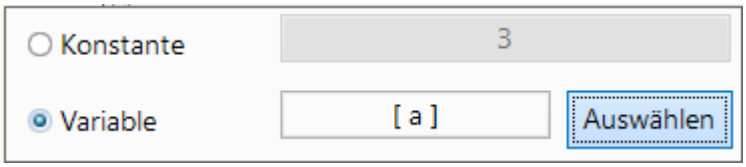
Um das erste Symbol eines Elements entlang der X- und Y-Achse zu positionieren, müssen Koordinaten angegeben werden. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse von links nach rechts von 0 bis 15 (abhängig von der Eigenschaft **Länge**).
– Entlang der Y-Achse von oben nach unten (abhängig von der Anzahl der im Display-Editor eingestellten Bildschirmzeilen).

Der Koordinatenwert (Ganzzahl) kann im Dropdown-Menü in der Koordinatenzeile eingestellt werden (X und Y sind gleich konfiguriert):

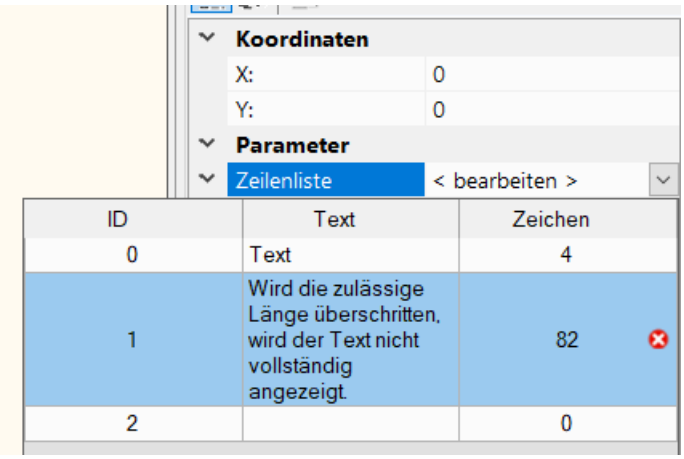


Im Dropdown-Menü können Sie die Methode zur Angabe der Koordinaten auswählen: konstant oder variabel. Um einen festen Standort für ein Element auszuwählen, geben Sie den Koordinatenwert im Eingabefeld an.
Um eine durchgehende Linie zu erstellen, wählen Sie die Option **Variable** aus und drücken Sie **Auswählen**. Es wird sich *Variablentabelle* öffnen. Darin sollten Sie eine Variable vom Typ Integer auswählen, die den Wert der Koordinate bestimmt, und die Schaltfläche **OK** drücken. Die ausgewählte Variable wird im Eingabefeld angezeigt:



Parameter

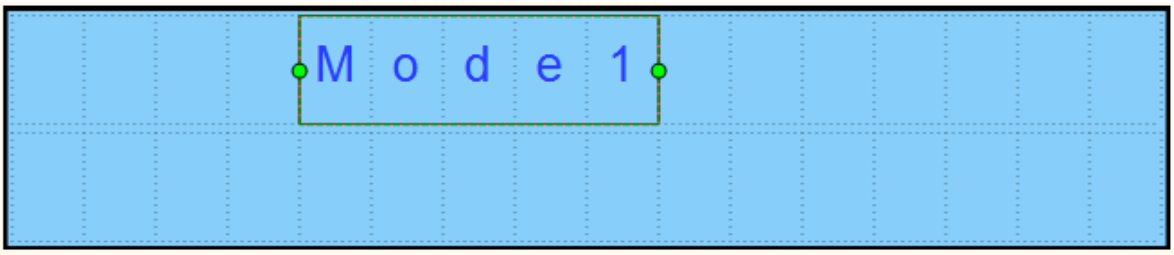
– **Variable** — Binden Sie das Element an eine ganzzahlige Projektvariable. Wählen Sie die Variable über „…“ aus der *Variablentabelle* aus.
– **Zeilenliste** — Bearbeiten Sie die Tabelle, um den Zahlenwerten der gebundenen Variable bestimmte Texte oder Zeichen zuzuordnen. Geben Sie in der Spalte ID den Zahlenwert, in der Spalte Text den anzuzeigenden Text und in der Spalte Zeichen die Zeichen-ID ein.



– **Länge** — Legen Sie die maximale Feldlänge fest. Wenn die zulässige Länge überschritten wird, wird der Text nicht vollständig angezeigt.
– **Sichtbarkeit** — Weisen Sie dem Parameter eine boolesche Variable zu, die die Sichtbarkeit des Elements bestimmt.

Anzeige

Der Display-Editor zeigt an, wie das Element **Dynamische Box** auf dem Gerätebildschirm aussehen wird.

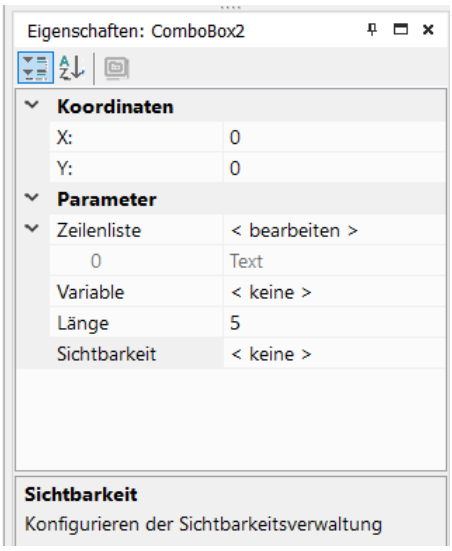


7.6.5 ComboBox

Element **ComboBox** Dient dazu, mit den Tasten des Geräts eine Zeile aus mehreren Textzeilen auszuwählen und den der Auswahl entsprechenden Wert in die verknüpfte Variable zu schreiben.

Eigenschaften einrichten

Damit das Element korrekt funktioniert, müssen Sie die Standortkoordinaten angeben und die Parameter von Variablen und Zeilen im Eigenschaftenfenster konfigurieren.

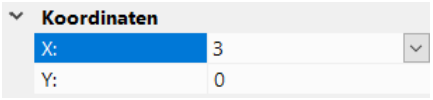


Koordinaten

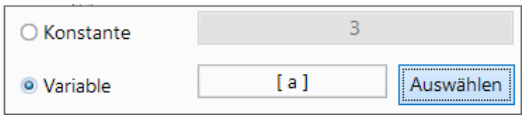
Um das erste Symbol eines Elements entlang der X- und Y-Achse zu positionieren, müssen Koordinaten angegeben werden. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse von links nach rechts von 0 bis 15 (abhängig von der Eigenschaft **Länge**).
– Entlang der Y-Achse von oben nach unten (abhängig von der Anzahl der im Display-Editor eingestellten Bildschirmzeilen).

Der Koordinatenwert (Ganzzahl) kann im Dropdown-Menü in der Koordinatenzeile eingestellt werden (X und Y sind gleich konfiguriert):

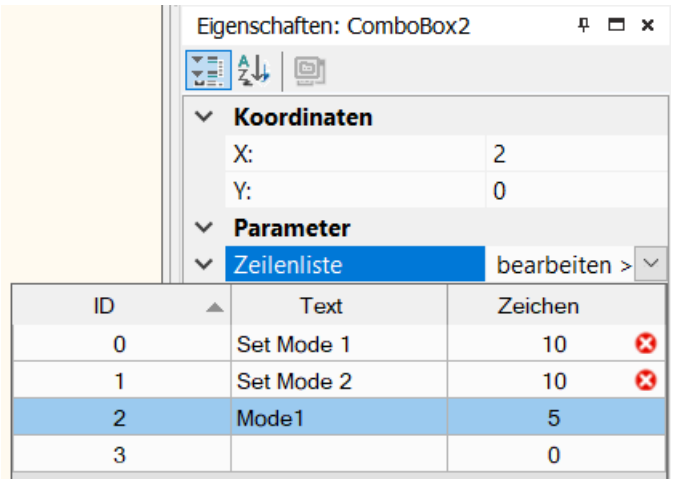


Im Dropdown-Menü können Sie die Methode zur Angabe der Koordinaten auswählen: Konstante oder Variable. Um einen festen Standort für ein Element auszuwählen, geben Sie den Koordinatenwert im Eingabefeld an.
Um eine durchgehende Linie zu erstellen, wählen Sie die Option **Variable** aus und drücken Sie **Auswählen**. Es wird sich *Variablentabelle* öffnen. Darin sollten Sie eine Variable vom Typ Integer auswählen, die den Wert der Koordinate bestimmt, und die Schaltfläche **OK** drücken. Die ausgewählte Variable wird im Eingabefeld angezeigt:



Parameter

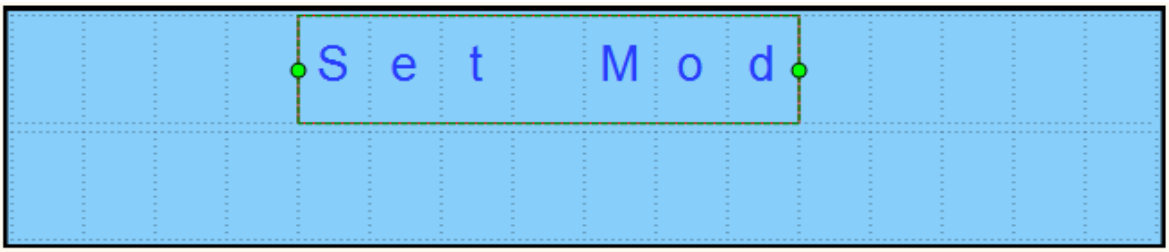
– **Variable** — Binden Sie das Element an eine ganzzahlige Projektvariable. Wählen Sie die Variable über „…“ aus der *Variablentabelle* aus.
– **Zeilenliste** — Bearbeiten Sie die Tabelle, um den Werten der gebundenen Integer-Variable bestimmte Texte oder Zeichen zuzuordnen. Geben Sie in der Spalte ID den Zahlenwert, in der Spalte Text den anzuzeigenden Text und in der Spalte Zeichen die Zeichen-ID ein.



– **Länge** — Legen Sie die maximale Feldlänge fest. Wenn die zulässige Länge überschritten wird, wird der Text nicht vollständig angezeigt.
– **Sichtbarkeit** — Weisen Sie dem Parameter eine boolesche Variable zu, die die Sichtbarkeit des Elements bestimmt.

Anzeige

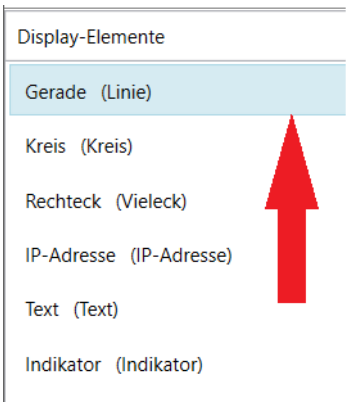
Der Display-Editor zeigt an, wie das Element **ComboBox** auf dem Gerätebildschirm aussehen wird.



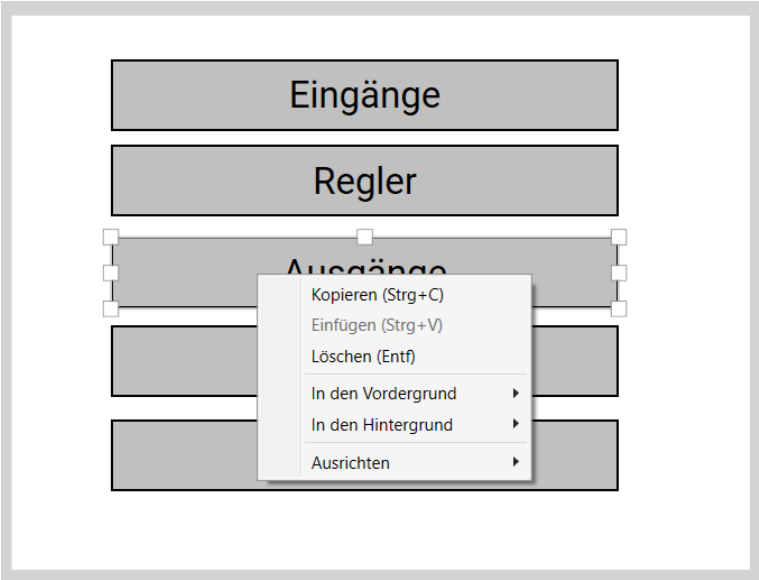
7.7 Einfache grafische Elemente

– *Text*
– *Indikator*
– *Fortschrittsanzeige*
– *Dynamische Box*
– *REAL/INT Ein-/Ausgabe*
– *Zeiteingabe/-ausgabe*
– *IP-Adresse*
– *Linie*
– *Vieleck*
– *Kreis*
– *Taste*
– *Schalter*
– *Schaltergruppe*
– *Bild*
– *Menü*
– *Diagramm*

Diese Elemente dienen der Gestaltung der Anzeige von Geräten mit grafischem Farb-LCD. Werden der Displayform mehrere editierbare Visualisierungselemente hinzugefügt, richtet sich die Reihenfolge ihrer Auswahl mit der SEL-Taste nach der Reihenfolge in der Liste „Display-Elemente“ von unten nach oben (die unteren Elemente werden zuerst hervorgehoben):



Sie können die Reihenfolge der Elemente ändern, indem Sie ein Element per Drag & Drop in die Liste ziehen oder das Kontextmenü verwenden, indem Sie mit der rechten Maustaste auf ein Element auf der Arbeitsfläche klicken:



Die folgenden Daten geben Auskunft über die benötigte RAM-Menge (Visualisierungs-RAM) zur Anzeige von Visualisierungselementen auf Geräten mit grafischem Farb-LCD:

Visualisierungs-element	Verwendeter Visualisierungs-RAM, Bytes	Zusätzliche Nutzung	Notiz
Leerer Bildschirm	252		
Physische Tasten	80	RAM-Berechnung: 80 Bytes (pro Schaltfläche) + 20 Bytes * Anzahl der Variablen + 10 Bytes * Anzahl der Aktionen + 10 Bytes (obligatorische Reserve) Berechnungsbeispiele: 1. Eine Variable bei kurzem Tastendruck ändern = 80+20*1+10*1 +10 = 120 Bytes	

Visualisierungs-element	Verwendeter Visualisierungs-RAM, Bytes	Zusätzliche Nutzung	Notiz
		2. Bildschirmübergang bei kurzem Drücken = 80+20*0 +10*1+10 = 100 Bytes 3. Eine Variable bei kurzem und langem Drücken ändern und bei kurzem Drücken einen Bildschirmübergang auslösen = 80 +20*2+10*3+10 = 160 Bytes	
Text	464	Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen.	
Indikator	Standardgröße, ohne Rahmen – 630	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen.	Die Menge des verwendeten RAM hängt direkt von den Abmessungen des Elements und der Rahmendicke ab: – Größere Elementabmessungen benötigen mehr Speicherplatz für die Anzeige. – Frame 0 px (deaktiviert) - minimaler RAM-Verbrauch für ein Element dieser Größe. – Frame 1 px - Der RAM-Verbrauch steigt im Vergleich zu einem Element ohne Frame. – Bei einem Rahmen von 2 Pixeln oder mehr steigt der RAM-Verbrauch im Vergleich zu einem Element mit einem 1 Pixel breiten Rahmen. Sobald die Rahmendicke 2 Pixel überschreitet, erhöht sich der RAM-Verbrauch nicht weiter und entspricht dem eines Elements mit

Visualisierungs-element	Verwendeter Visualisierungs-RAM, Bytes	Zusätzliche Nutzung	Notiz
			einem 2 Pixel breiten Rahmen.
Fortschrittsanzeige	408	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen. Wertsichtbarkeit: +16 Bytes für die Anzeige des Werts (Wert, sichtbar = Ja).	
Dynamische Box	836 (für eine Zeile)	+4 Bytes für jede weitere Zeile. Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen. Eingabesteuerung: +12 Bytes zum Ermöglichen der Bearbeitung (Bearbeitung deaktivieren = Nein).	
REAL/INT Eingang/ Ausgang	776	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen. Eingabesteuerung: +24 Bytes zum Ermöglichen der Bearbeitung (Bearbeitung deaktivieren = Nein).	
Zeiteingabe/-ausgabe	688 nur für Datum **oder** Uhrzeit. 704 für Datum **und** Uhrzeit.	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen.	

Visualisierungs-element		Verwendeter Visualisierungs-RAM, Bytes	Zusätzliche Nutzung	Notiz
			Eingabesteuerung: +28 Bytes zum Ermöglichen der Bearbeitung (Bearbeitung deaktivieren = Nein).	
Ein-/Ausgabe der IP-Adresse		676	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen. Eingabesteuerung: +24 Bytes zum Ermöglichen der Bearbeitung (Bearbeitung deaktivieren = Nein).	
Linie	Gerade	182	Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen.	
	Gestri-chelte Linie	190		
Polygon	Recht-eck	338	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen. Füllung: +132 Bytes (Darstellung, Füllung = Ja). Aktionskonfiguration (Bildschirmübergang oder Variablenänderung) = 10 Bytes * Anzahl der Aktionen + 26 Bytes.	
	Dreieck	334		
	Diamant	338		
Kreis	Kreis	Standardgröße, ohne Füllung – 978 Standardgröße, mit Füllung – 1346	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen. Aktionskonfiguration (Bildschirmübergang oder	Die Menge des verwendeten RAM hängt direkt von den Abmessungen des Elements und dem Vorhandensein einer Füllung ab: – Größere Elementabmessun-gen benötigen mehr Speicherplatz für die Anzeige.
	Kreiss-ektor	Standardgröße, ohne Füllung – 1502 Standardgröße, mit Füllung – 1878		

Visualisierungs-element		Verwendeter Visualisierungs-RAM, Bytes	Zusätzliche Nutzung	Notiz
			Variablenänderung) = 10 Bytes * Anzahl der Aktionen + 26 Bytes.	– Der RAM-Verbrauch für ein Element mit Füllung ist größer als für ein Element ohne Füllung. Bei bestimmten Kombinationen von Einstellungen kann der Parameter „Dicke“ des Fokusrahmens den verwendeten Visualisierungs-RAM geringfügig beeinflussen.
Taste		Rechteckig, ohne abgerundete Ecken, ohne Rahmen – 894	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen. Aktionskonfiguration (Bildschirmübergang oder Variablenänderung) = 10 Bytes * Anzahl der Aktionen + 26 Bytes. +6 Bytes für das Hinzufügen eines 1px Rahmens (für einen rechteckigen Button ohne abgerundete Ecken). +20 Bytes für das Hinzufügen eines Rahmens mit einer Dicke von 2 Pixeln oder mehr (für einen rechteckigen Button ohne abgerundete Ecken).	Bei einem runden Button oder einem rechteckigen Button mit abgerundeten Ecken hängt die benötigte RAM-Menge direkt von den Abmessungen des Elements ab: – Größere Elementabmessungen benötigen mehr Speicherplatz für die Anzeige. Bei bestimmten Kombinationen von Einstellungen kann der Parameter „Dicke“ des Fokusrahmens den verwendeten Visualisierungs-RAM geringfügig beeinflussen.
Schalten		Standardgröße, ohne Rahmen– 1192	Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen.	Die Menge des verwendeten Arbeitsspeichers hängt direkt von den Abmessungen des Elements ab: – Größere Elementabmessungen benötigen mehr Speicherplatz für die Anzeige. Bei bestimmten Kombinationen von Einstellungen kann der Parameter „Dicke“ des Fokusrahmens den

Visualisierungs-element	Verwendeter Visualisierungs-RAM, Bytes	Zusätzliche Nutzung	Notiz
			verwendeten Visualisierungs-RAM geringfügig beeinflussen.
Schaltergruppe	1688 (für eine Gruppe mit einer Zelle)	+ 208 Bytes für jede zusätzliche Zelle. Bei Verwendung einer Variablen: +20 Bytes pro Variable. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen.	
Bild	372	Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen. Aktionskonfiguration (Bildschirmübergang oder Variablenänderung) = 10 Bytes * Anzahl der Aktionen + 26 Bytes. + 32 Bytes zum Hinzufügen eines Frames.	Berechnung des Flash-Speicherverbrauchs für die Benutzervisualisierung = Breite * Höhe * 2. Die Berechnung verwendet die Breite und Höhe (in Pixeln) des Originalbildes, das in den Visualisierungseditor geladen wurde. Wenn dasselbe Bild kopiert und mit mehreren Visualisierungselemen-ten verknüpft wird, wird kein zusätzlicher Flash-Speicher für die Benutzervisualisierung verwendet.
Speisekarte	1870 (für ein Menü mit einem Artikel)	+ 298 Bytes für jede zusätzliche Zeile. Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen des Elements oder der Zeile.	

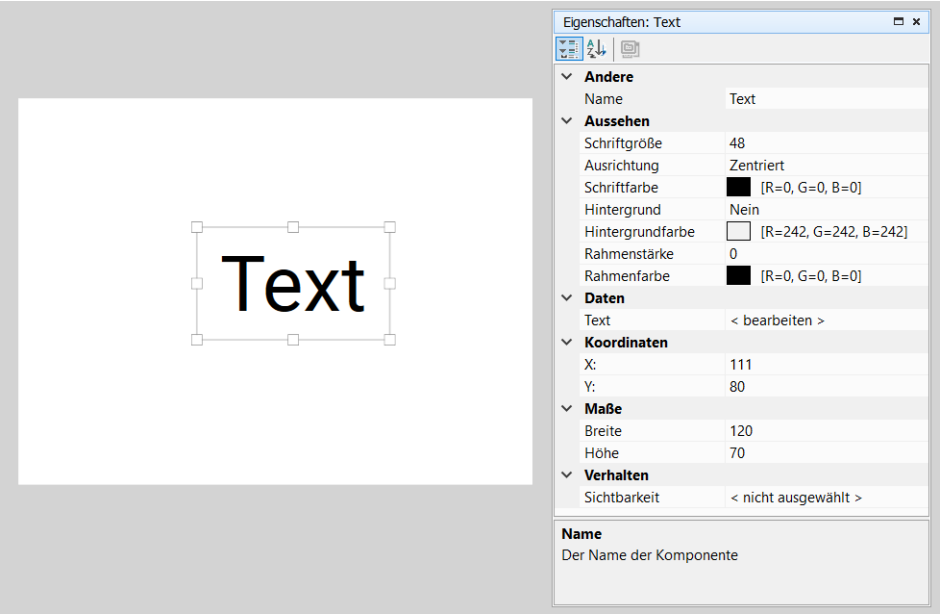
Visualisierungs-element		Verwendeter Visualisierungs-RAM, Bytes	Zusätzliche Nutzung	Notiz
Dia-gramm	Hinter-grund-verarbei-tung deakti-viert	Ohne Stifte, Zeitskala 2 Sek. – 982	Zusätzliche Kosten für die Verlängerung des Anzeigezeitraums: Von 2 Bytes für 3 Sekunden bis zu 916 Bytes für 23:59:59. Zusätzliche Kosten pro Stift: + 20 Bytes (erster Stift) + jeder weitere Stift abhängig vom Zeitbereich (von 74 Bytes für 2 Sekunden bis zu 990 Bytes für 23:59:59). Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen des Elements oder des Stifts.	Die Menge des verwendeten Arbeitsspeichers hängt direkt von der Breite des Elements ab: – Ein breiteres Element benötigt mehr Speicherplatz für die Anzeige.
	Hinter-grund-operati-on aktiviert	Ohne Stifte, Zeitrahmen 2 Sekunden– 1002	Zusätzliche Kosten für die Verlängerung des Anzeigezeitraums: Von 6 Bytes für 3 Sekunden bis zu 2748 Bytes für 23:59:59. Zusätzliche Kosten pro Stift: + 20 Bytes (erster Stift) + jeder weitere Stift abhängig vom Zeitbereich (von 94 Bytes für 2 Sekunden bis zu 2842 Bytes für 23:59:59). Wenn die Sichtbarkeit durch eine Variable gesteuert wird: +20 Bytes für die Bindung der Sichtbarkeitsvariablen des Elements oder des Stifts.	

7.7.1 Text

Das **Text**-Element dient zum Platzieren eines Textblocks im Visualisierungsbildschirm.

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben, Text eingeben und andere Parameter im Eigenschaftenfenster festlegen.



Koordinaten

Die Position des Elements wird durch die X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Elements auf dem Bildschirm festgelegt. Die Koordinaten für beide Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Maße

Die Maße eines Elements werden durch die X- und Y-Achse bestimmt und im Eigenschaftenfenster oder durch Ändern der Elementgrenzen auf dem Bildschirm festgelegt.

Parameter

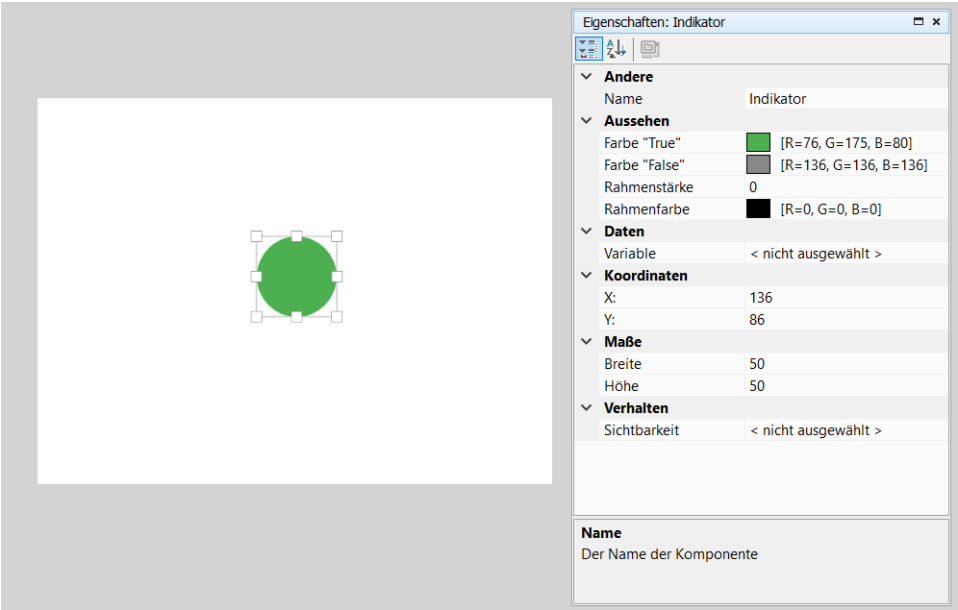
Gruppe	Parameter	Beschreibung
Andere	**Name**	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.
Aussehen	**Schriftgröße**	Bestimmt die Größe des im Element angezeigten Textes.
	Ausrichtung	Legt die Ausrichtung des im Element angezeigten Textes fest.
	Schriftfarbe	Bestimmt die Farbe des im Element angezeigten Textes.
	Hintergrund	Definiert den Hintergrund des im Element angezeigten Textes.
	Hintergrundfarbe	Bestimmt die Hintergrundfarbe des im Element angezeigten Textes.
	Rahmenstärke	Legt die Stärke des Elementrahmens fest.
	Rahmenfarbe	Bestimmt die Farbe des Elementrahmens.
Daten	**Text**	Definiert den im Element angezeigten Textinhalt.
Verhalten	**Sichtbarkeit**	Bestimmt die Sichtbarkeit des Elements über eine gebundene boolesche Variable.

7.7.2 Indikator

Das Element **Indikator** dient zur Anzeige des Wertes einer booleschen Variable im Visualisierungsbildschirm. Wenn der Variablenwert **1** ist, nimmt der Indikator die Farbe „True“ an. Wenn der Variablenwert **0** ist, nimmt der Indikator die Farbe „False“ an.

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter im Eigenschaftenfenster festlegen.



Koordinaten

Die Position des Elements wird durch die X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Elements auf dem Bildschirm festgelegt. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Maße

Die Maße eines Elements werden durch die X- und Y-Achse bestimmt und können im Eigenschaftenfenster oder durch Ändern der Grenzen des Elements auf dem Bildschirm festgelegt werden.

Parameter

Gruppe	Parameter	Beschreibung
Andere	**Name**	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.
Aussehen	**Farbe “True”**	Bestimmt die Farbe des Elements, wenn der Wert der gebundenen Variable TRUE ist.
	Farbe “False”	Bestimmt die Farbe des Elements, wenn der Wert der gebundenen Variable FALSE ist.
	Rahmenstärke	Legt die Stärke des Elementrahmens fest.

Gruppe	Parameter	Beschreibung
	Rahmenfarbe	Bestimmt die Farbe des Elementrahmens.
Daten	**Variable**	Bindet eine boolesche Variable zur Anzeige an den Parameter.
Verhalten	**Sichtbarkeit**	Bestimmt die Sichtbarkeit des Elements über eine gebundene boolesche Variable.

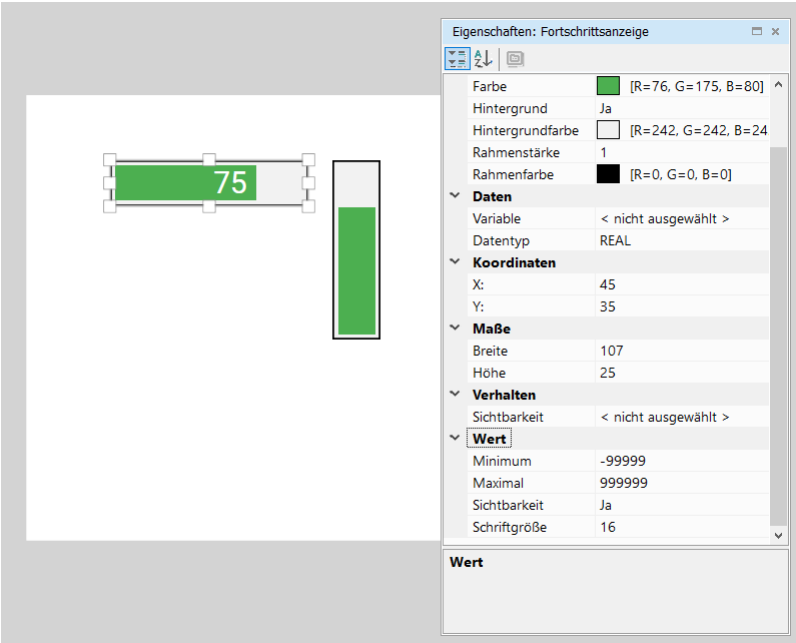
HINWEIS
Wenn keine Variable an den Parameter **Variable** gebunden ist, ist die Farbe des Elements immer **True**.

7.7.3 Fortschrittanzeige

Das Fortschrittsanzeige-Element dient dazu, den Wert einer Ganzzahl- oder Gleitkommavariable als Skala auf dem Visualisierungsbildschirm anzuzeigen.

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter im Eigenschaftenfenster festlegen.



Koordinaten

Die Position des Elements wird durch die X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Elements auf dem Bildschirm festgelegt. Die Koordinaten für beide Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Maße

Die Abmessungen des Elements werden durch die X- und Y-Achsen definiert und können im Eigenschaftenfenster oder durch Ändern der Elementränder im Bildschirmfeld festgelegt werden. Für die vertikale Ausrichtung des Elements muss der Y-Wert größer als der X-Wert sein.

Parameters

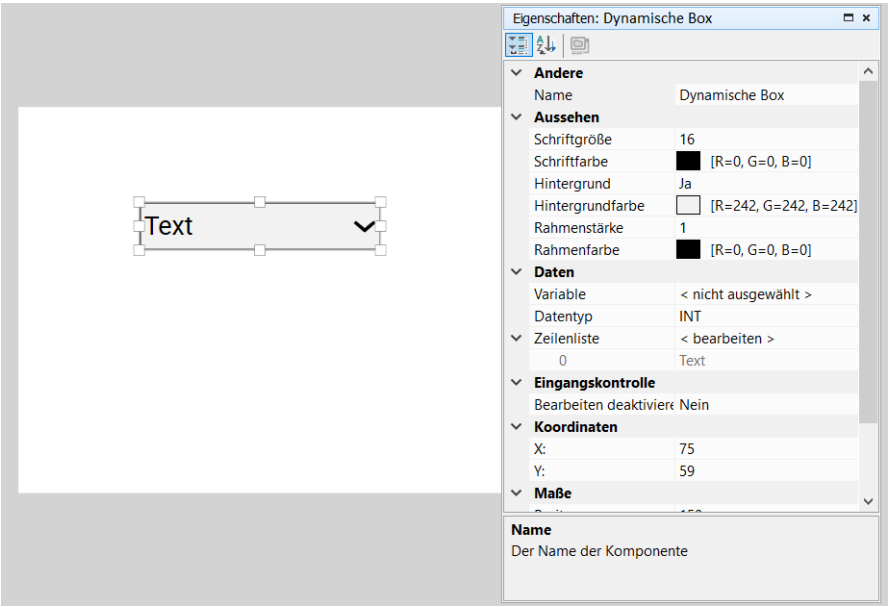
Gruppe	Parameter	Beschreibung
Andere	Name	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.
Aussehen	Farbe	Bestimmt die Elementskalenfarbe.
	Hintergrund	Definiert den Hintergrund des angezeigten Elements.
	Hintergrundfarbe	Bestimmt die Hintergrundfarbe des angezeigten Elements.
	Rahmenstärke	Legt die Stärke des Elementrahmens fest.
	Rahmenfarbe	Bestimmt die Farbe des Elementrahmens.
Daten	Variable	Bindet eine Integer- oder Gleitkommavariable zur Anzeige an den Parameter.
	Datentyp	Legt den Datentyp der gebundenen Variable als REAL oder INT fest.
Wert	Minimum	Bestimmt den minimalen Skalenwert als 0 %.
	Maximal	Bestimmt den maximalen Skalenwert als 100 %
	Sichtbarkeit	Steuert die Anzeige des Skalenwerts im Element.
	Schriftgröße	Bestimmt die Schriftgröße des angezeigten Textes.
Verhalten	Sichtbarkeit	Bestimmt die Sichtbarkeit des Elements über eine gebundene boolesche Variable.

7.7.4 Dynamische Box

Das Element **Dynamische Box** dient zur Auswahl und Anzeige des Werts einer Ganzzahlvariablen aus einer Dropdown-Liste auf dem Visualisierungsbildschirm. Dem Ganzzahlwert kann eine Textbezeichnung zugewiesen werden, die in der Dropdown-Liste angezeigt wird.

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter und Daten im Eigenschaftenfenster festlegen.



Koordinaten

Die Position des Elements wird durch die X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Elements auf dem Bildschirm festgelegt. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Maße

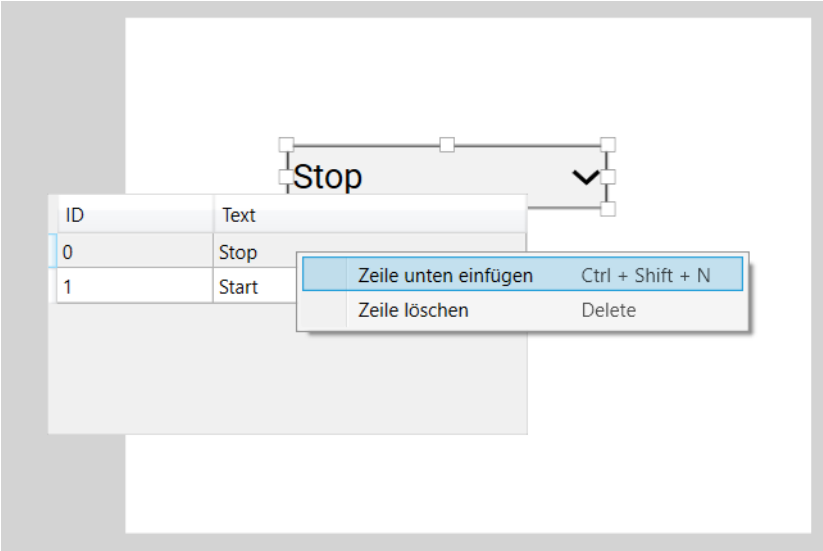
Die Maße eines Elements werden durch die X- und Y-Achse bestimmt und im Eigenschaftenfenster oder durch Ändern der Elementgrenzen auf dem Bildschirm festgelegt.

Parameter

Gruppe	Parameter	Beschreibung
Andere	**Name**	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.
Aussehen	**Schriftgröße**	Bestimmt die Größe des im Element angezeigten Textes.
	Schriftfarbe	Bestimmt die Farbe des im Element angezeigten Textes.
	Hintergrund	Definiert den Hintergrund des im Element angezeigten Textes.
	Hintergrundfarbe	Bestimmt die Hintergrundfarbe des im Element angezeigten Textes.
	Rahmenstärke	Legt die Stärke des Elementrahmens fest.
	Rahmenfarbe	Bestimmt die Farbe des Elementrahmens.
Daten	**Variable**	Bindet eine Variable an den Parameter, um Werte zu speichern.
	Datentyp	Legt den Datentyp der gebundenen Variable als INT oder BOOL fest. Wenn ein BOOL-Variablentyp ausgewählt ist, kann die Dropdown-Liste des Elements nur zwei Zeilen enthalten. Wenn ein INT-Variablentyp ausgewählt ist, kann die Dropdown-Liste maximal 128 Zeilen enthalten.
	Zeilenliste	Zeigt in tabellarischer Form die Variablenwerte und zugehörigen Namen in der Dropdown-Liste an.
Eingangskon-trolle	**Bearbeiten deaktivieren**	Deaktiviert bei aktivierter Option die Bearbeitung und erlaubt nur die Anzeige des Werts. Überschreibt andernfalls eine Eingabe im Element den Variablenwert.
Verhalten	**Sichtbarkeit**	Bestimmt die Sichtbarkeit des Elements über eine gebundene boolesche Variable.

Um der Dropdown-Liste eines Elements eine neue Zeile hinzuzufügen, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie im Bildschirmeditor auf das Element **Dynamische Box**.
2. Klicken Sie im sich öffnenden Fenster mit der rechten Maustaste auf eine der Zeilen in der Tabelle.
3. Wählen Sie **Zeile unten einfügen**.



Beispiel:

Tabelle 7.7 Zeilenliste

0	Stop
1	Start
2	Pause

7.7.5 REAL/INT Ein-/Ausgabe

Das Element **REAL/INT Ein-/Ausgabe** dient der Eingabe und Anzeige des Wertes einer Integer- oder Floating-Point-Variablen auf dem Visualisierungsbildschirm.

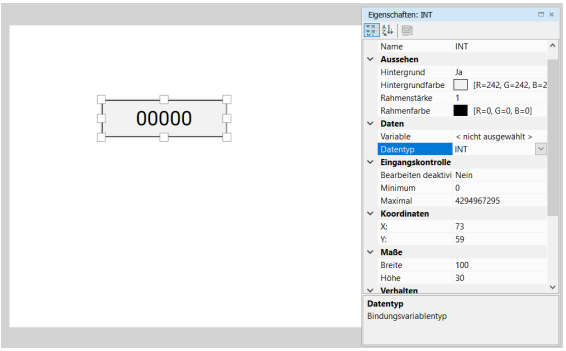


Abb. 7.1 INT Ein-/Ausgabe

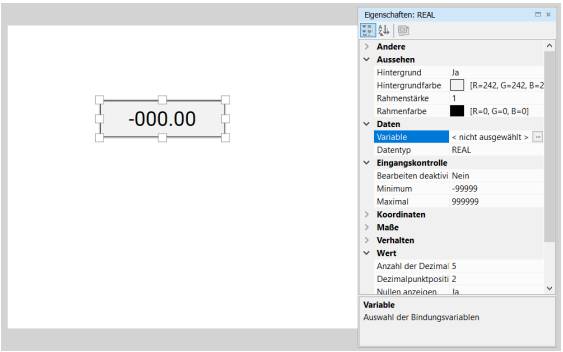


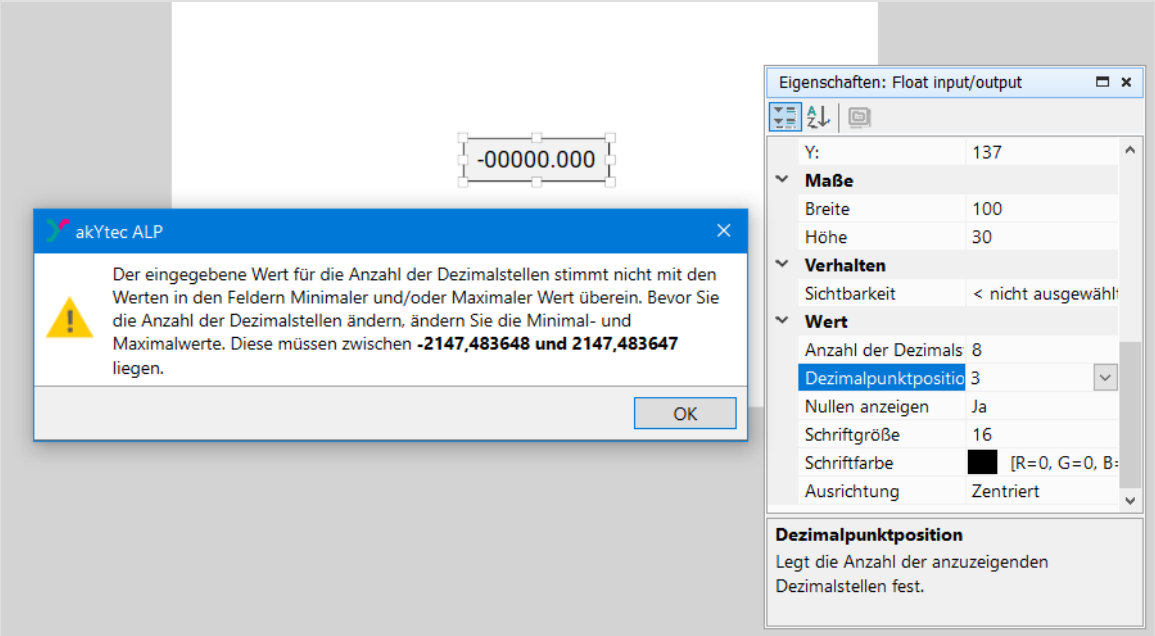
Abb. 7.2 REAL Ein-/Ausgabe

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
Aussehen	**Hintergrund**	Definiert den Hintergrund des im Element angezeigten Textes.	
	Hintergrund-farbe	Bestimmt die Hintergrundfarbe des im Element angezeigten Textes.	
	Rahmenstär-ke	Legt die Stärke des Elementrahmens fest.	
	Rahmenfarbe	Bestimmt die Farbe des Elementrahmens.	
Daten	**Variable**	Bindet eine Variable an den Parameter.	

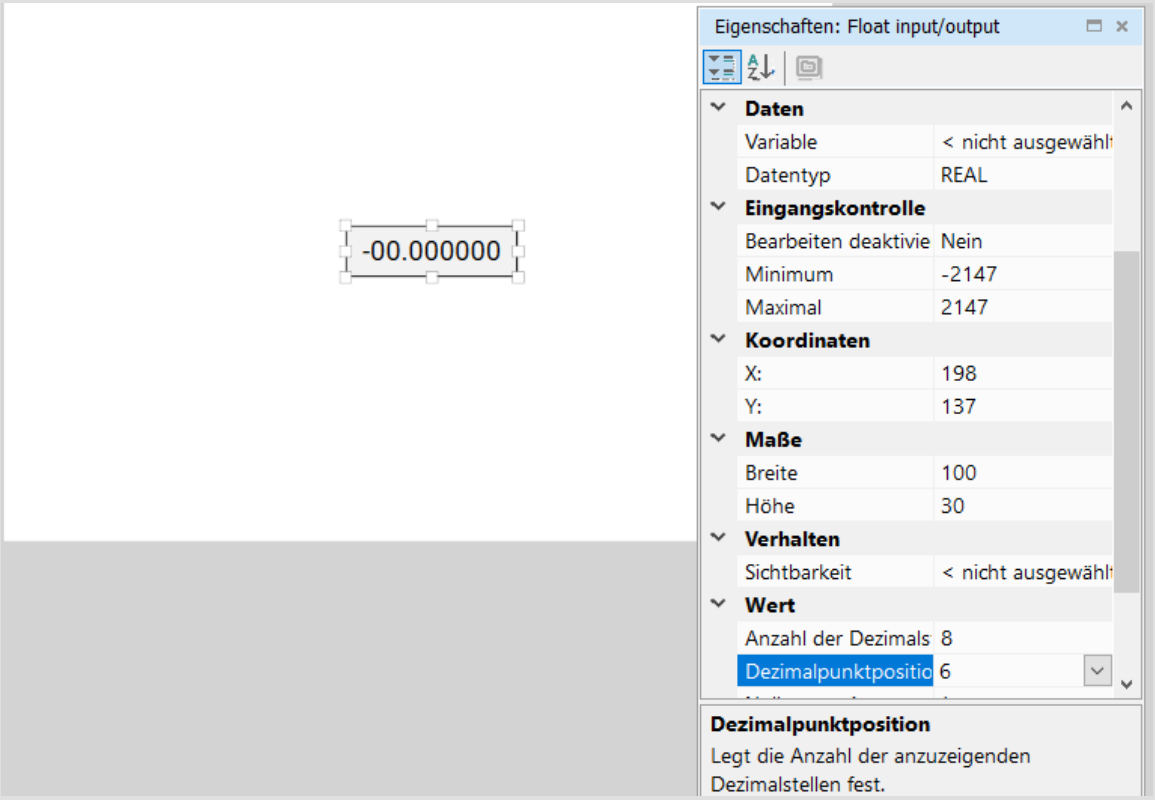
Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
	Datentyp	Um einen ganzzahligen Wert ein- oder auszugeben, müssen Sie den Typ **INT** angeben. Für die Ein-/Ausgabe von Gleitkommawerten muss der Datentyp **REAL** angegeben werden.	
Wert	**Anzahl der Dezimalstel-len**	Bestimmt die Anzahl der im Element angezeigten Dezimalstellen.	Bei der Auswahl einer Gleitkomma-Variablen: darf nicht kleiner oder gleich dem Wert von **Anzahl der Dezimalstellen** sein.
	Nullen anzeigen (Diese Einstellung ist nur verfügbar, wenn die Gleitkomma-Variable ausgewählt ist)	Legt die Anzeige von nachgestellten Nullen, sofern eine Gleitkomma-Variable ausgewählt ist fest.	Erfordert bei Änderung des Wertes eine Anpassung von Minimum und Maximum.
	Schriftgröße	Bestimmt die Größe des im Element angezeigten Textes.	
	Schriftfarbe	Bestimmt die Farbe des im Element angezeigten Textes.	
	Ausrichtung	Legt die Ausrichtung des im Element angezeigten Textes fest.	
Eingangs-kontrolle	**Bearbeiten deaktivieren**	Bei der Einstellung **Ja** kann der Wert der gebundenen Variable nur angezeigt werden. Die Eingabemöglichkeit ist nicht verfügbar. Andernfalls überschreibt die Eingabe eines Wertes im Element den Variablenwert.	
	Minimum	Mindestwert zur Eingabe in ein Element	Bei der Auswahl einer Gleitkomma-Variablen: Der Grenzwert für den Wert wird mit folgender Formel berechnet: 2147483648 / 10^Wert der Eigenschaft **Anzahl der Dezimalstellen** Eine Änderung des Wertes kann eine Änderung des Wertes der Eigenschaft **Anzahl der Dezimalstellen** erfordern

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
			(siehe Beispiel nach der Tabelle)
	Maximum	Maximalwert für Eingabe im Element	Bei der Auswahl einer Gleitkomma-Variablen: Der Grenzwert für den Wert wird mit folgender Formel berechnet: – 2147483648 / 10^ (Wert der Eigenschaft **Anzahl der Dezimalstellen**) Eine Änderung des Wertes kann eine Änderung des Wertes der Eigenschaft **Anzahl der Dezimalstellen** erfordern (siehe Beispiel nach der Tabelle)
Koordina-ten	**X**	Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.	Die Koordinaten können durch Verschieben des Objekts über den Bildschirm festgelegt werden
	Y	Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.	
Verhalten	**Sichtbarkeit**	An den Parameter ist eine Boolesche Variable gebunden, die die Sichtbarkeit des Elements bestimmt.	
Andere	**Name**	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.	
Maße	**Breite**	Elementbreite entlang der X-Achse	Die Größe eines Elements kann durch Ändern der Grenzen des Objekts im Bildschirmbereich festgelegt werden
	Höhe	Elementhöhe entlang der Y-Achse	

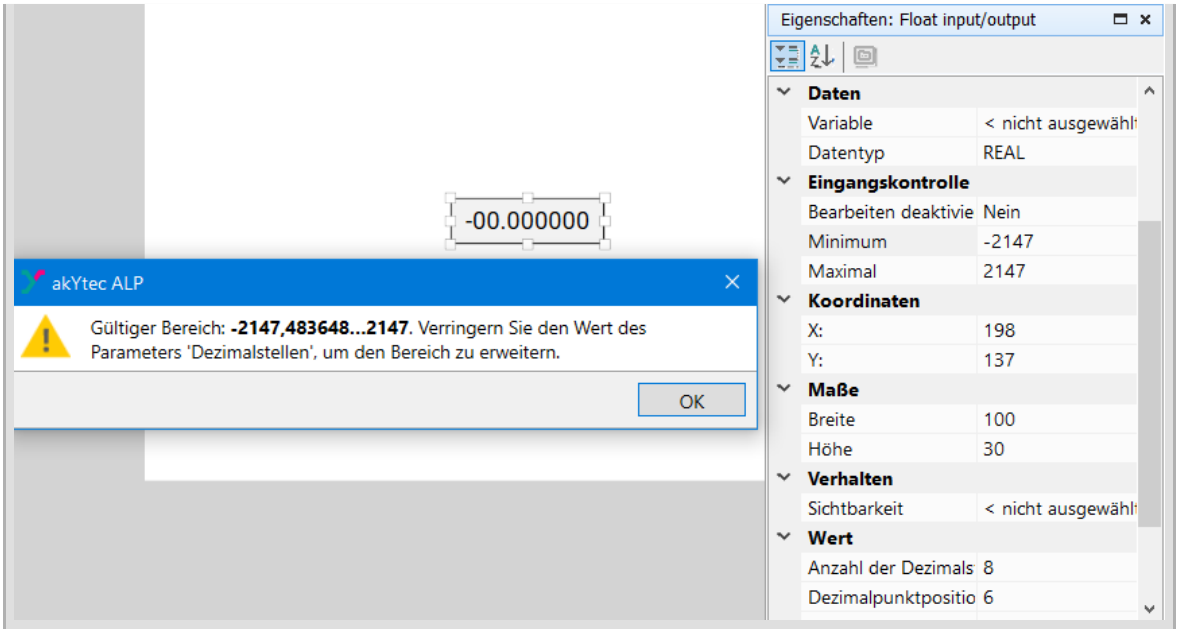
Beispiel:



Wenn Sie für die Eigenschaft **Anzahl der Dezimalstellen** den Wert 6 auswählen, wird eine Fehlermeldung angezeigt, die die **Minimum/Maximum** Wertegrenzen angibt, die anhand der folgenden Formel berechnet werden:
Minimalwert = -2147483648 / 10^6 = -2147,483648
Maximalwert = 2147483647 / 10^6 = 2147,483647
Geben Sie die Werte für Minimum und Maximum innerhalb des angegebenen Bereichs ein und wählen Sie für die Eigenschaft **Anzahl der Dezimalstellen** den Wert 6 aus:



Der Wert hat sich geändert, das Visualisierungselement zeigt 6 Dezimalstellen an.
Wenn Sie versuchen, den Wert für **Minimum** oder **Maximum** auf einen Wert außerhalb des mit der obigen Formel berechneten Bereichs zu ändern, wird eine Fehlermeldung angezeigt:



Die Eigenschaften **Anzahl der Dezimalstellen** und **Minimum** und **Maximum** stehen in einem wechselseitigen Zusammenhang.

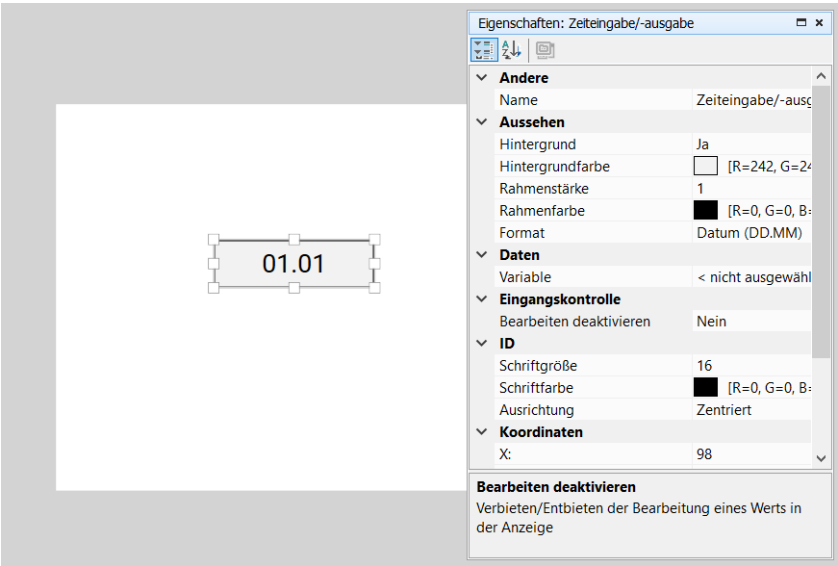
HINWEIS
Wenn die Ausgabevariable die in den Parametern **Minimum** / **Maximum** festgelegten Grenzwerte überschreitet, zeigt das Element auf dem Gerät den aktuellen Zustand der Variablen an und hebt den Rahmen des Elements rot hervor.

7.7.6 Zeitangabe/-ausgabe

Das Element **Zeiteingabe/-ausgabe** dient zur Eingabe und Anzeige von Datums- und Uhrzeitwerten auf dem Visualisierungsbildschirm. Eine Integer-Variable speichert Datums- und Uhrzeitwerte und gibt die Anzahl der Sekunden seit dem 1. Januar 2000 (00:00:00 Uhr) an.

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter und Daten im Eigenschaftenfenster festlegen.



Koordinaten

Die Position des Elements wird durch die X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Elements über den Bildschirm festgelegt. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Maße

Die Abmessungen eines Elements werden durch die X- und Y-Achse bestimmt und im Eigenschaftenfenster oder durch Ändern der Elementgrenzen auf dem Bildschirm festgelegt.

Parameter

Gruppe	Parameter	Beschreibung
Aussehen	**Hintergrund**	Definiert den Hintergrund des angezeigten Elements.
	Hintergrundfarbe	Bestimmt die Hintergrundfarbe des angezeigten Elements.
	Rahmenstärke	Legt die Stärke des Elementrahmens fest.
	Rahmenfarbe	Bestimmt die Farbe des Elementrahmens.
	Format	Wählen Sie aus der Dropdown-Liste das Format für die Anzeige von Datum und Uhrzeit aus. Formatangaben: – **ss** – Sekunden – **mm** – Minuten – **hh** – Stunden – **DD** – Tag – **MM** – Monat – **YYYY** – Jahr
Daten	**Variable**	Bindet eine Variable an den Parameter, um Werte zu speichern.
Eingangskon-trolle	**Bearbeiten deaktivieren**	Bei der Einstellung **Ja** kann der Wert der gebundenen Variable nur angezeigt werden. Die Eingabemöglichkeit ist nicht verfügbar. Andernfalls überschreibt die Eingabe eines Wertes im Element den Variablenwert.
ID	**Schriftgröße**	Bestimmt die Größe des im Element angezeigten Textes.
	Schriftfarbe	Bestimmt die Farbe des im Element angezeigten Textes.
	Ausrichtung	Ausrichtung des in einem Element angezeigten Textes.
Verhalten	**Sichtbarkeit**	An den Parameter ist eine BOOL-Variable gebunden, die die Sichtbarkeit des Elements bestimmt.
Andere	**Name**	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.

7.7.7 IP-Adresse

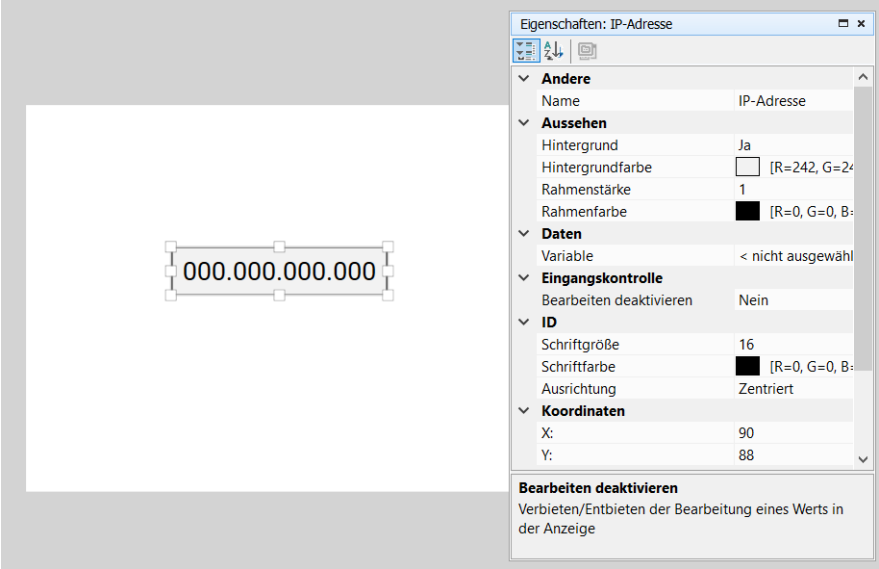
Das Element **IP-Adresse** dient zur Eingabe und Anzeige der IP-Adresse (IPv4) im Visualisierungsbildschirm. Der IP-Adresswert wird in einer Integer-Variable gespeichert, deren Wert mit folgender Formel berechnet wird:

$$int = o1 * 256^3 + o2 * 256^2 + o3 * 256^1 + o4 * 256^0,$$

wobei *int* der Wert einer Integer-Variable ist
o1, o2, o3, o4 – Oktette der IP-Adresse

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter und Daten im Eigenschaftenfenster festlegen.



Koordinaten

Die Position des Elements wird durch die X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Elements über den Bildschirm festgelegt. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Maße

Die Abmessungen eines Elements werden durch die X- und Y-Achse bestimmt und im Eigenschaftenfenster oder durch Ändern der Elementgrenzen auf dem Bildschirm festgelegt.

Parameter

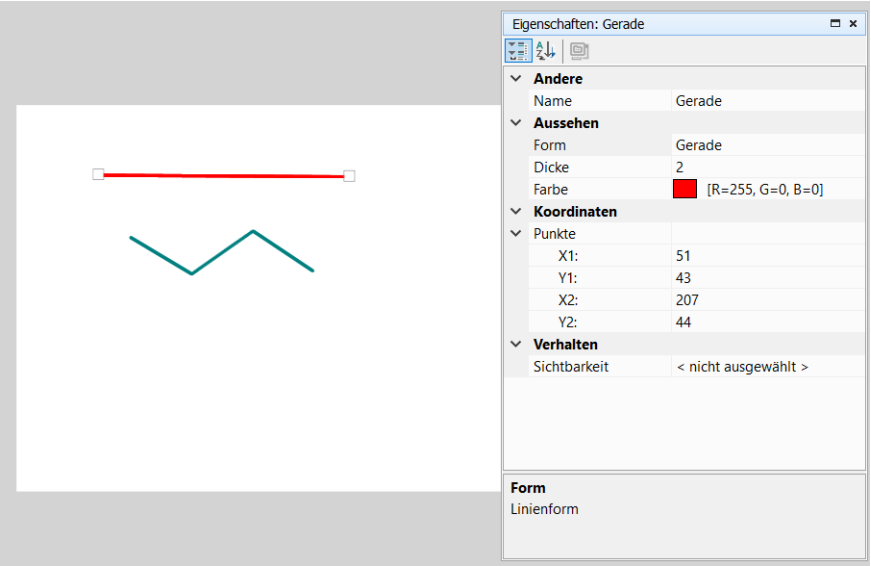
Gruppe	Parameter	Beschreibung
Aussehen	**Hintergrund**	Legt den Hintergrund des im Element angezeigten Textes fest.
	Hintergrundfarbe	Legt die Hintergrundfarbe des im Element angezeigten Textes fest.
	Rahmenstärke	Legt die Stärke des Elementrahmens fest.
	Rahmenfarbe	Legt die Farbe des Elementrahmens fest.
Daten	**Variable**	Bindet eine Variable an den Parameter.
	Datentyp	Legt den Datentyp für die Ein- und Ausgabe fest.
Eingangskot-rolle	**Bearbeiten deaktivieren**	Deaktiviert die Bearbeitung. Zeigt bei „Ja“ nur den Variablenwert an und verhindert Eingaben, andernfalls überschreibt die Eingabe den Variablenwert.

Gruppe	Parameter	Beschreibung
ID	Schriftgröße	Legt die Größe des im Element angezeigten Textes fest.
	Schriftfarbe	Legt die Farbe des im Element angezeigten Textes fest.
	Ausrichtung	Legt die Ausrichtung des im Element angezeigten Textes fest.
Verhalten	Sichtbarkeit	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.
Andere	Name	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.

7.7.8 Linie

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter im Eigenschaftenfenster festlegen.



Koordinaten

Die Position der Geraden wird durch zwei Punkte bestimmt, deren Koordinaten entlang der X- und Y-Achse im Eigenschaftenfenster oder durch Verschieben der Punkte auf dem Bildschirm festgelegt werden. Die Koordinaten entlang beider Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Die Lage der gestrichelten Linie wird durch vier Punkte bestimmt.

Parameter

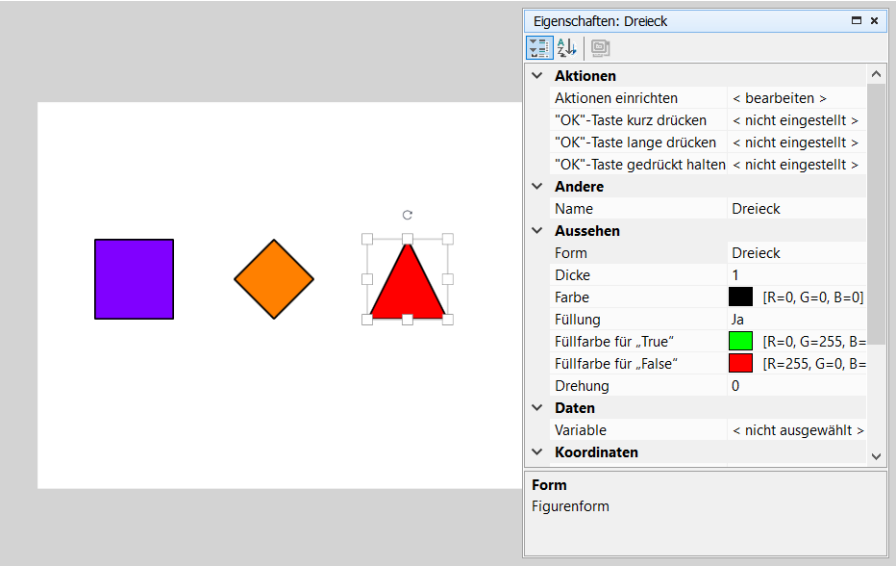
Gruppe	Parameter	Beschreibung
Andere	Name	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.
Aussehen	Form	Legt die Form der Anzeigelinie fest (gerade, gestrichelt).
	Stärke	Legt die Linienstärke fest.

Gruppe	Parameter	Beschreibung
	Farbe	Legt die Linienfarbe fest.
Verhalten	**Sichtbarkeit**	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.

7.7.9 Vieleck

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter im Eigenschaftenfenster festlegen.



Koordinaten

Die Position des Polygons wird durch die Koordinaten seines Mittelpunkts entlang der X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Objekts über den Bildschirm festgelegt. Die Koordinaten entlang beider Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt

Maße

Die Abmessungen eines Elements werden durch die X- und Y-Achse bestimmt und im Eigenschaftenfenster oder durch Ändern der Elementgrenzen auf dem Bildschirm festgelegt.

Parameter

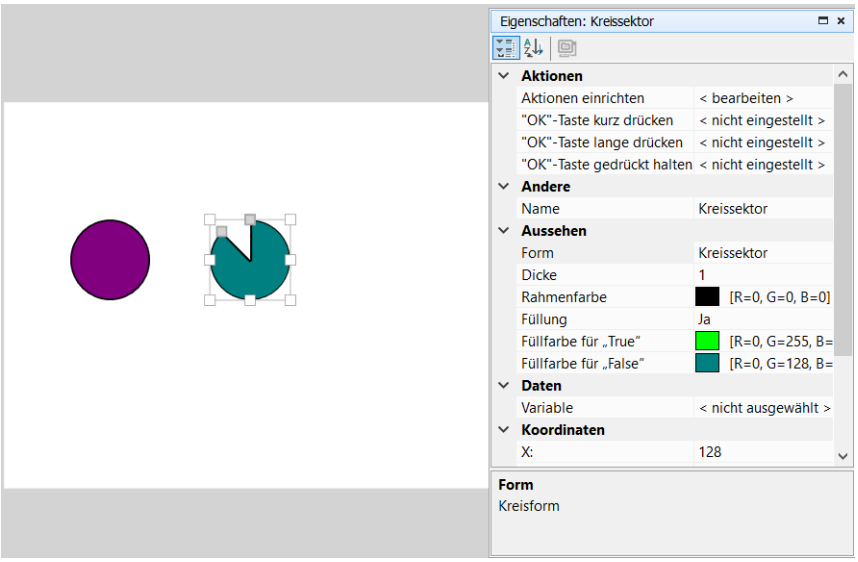
Gruppe	Parameter	Beschreibung
Andere	**Name**	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.
Aktionen	**Aktion einrichten**	Legt die Aktion zum Navigieren zwischen Bildschirmen fest.
	“OK”-Taste kurz drücken	Legt die bei kurzem Drücken der „OK“-Taste auszuführenden Aktionen fest.
	“OK”-Taste lange drücken	Legt die bei langem Drücken der „OK“-Taste auszuführenden Aktionen fest.

Gruppe	Parameter	Beschreibung
	“OK”-Taste gedrückt halten	Legt die beim Gedrückthalten der „OK“-Taste auszuführenden Aktionen fest.
Aussehen	**Form**	Legt die Form des angezeigten Vielecks fest (Dreieck, Rechteck, Raute).
	Dicke	Legt die Dicke der Vieleckumrisse fest.
	Farbe	Legt die Farbe des Vieleckumrisses fest.
	Füllung	Legt die Füllung der Form fest.
	Füllfarbe für «True»	Legt die Füllfarbe der Form für den Wert „True“ fest.
	Füllfarbe für «False»	Legt die Füllfarbe der Form für den Wert „False“ fest.
	Drehung	Legt die Drehung des Elements um seinen Mittelpunkt in Grad fest.
Daten	**Variable**	Bindet eine Variable an den Parameter.
Verhalten	**Sichtbarkeit**	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.

7.7.10 Kreis

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter im Eigenschaftenfenster festlegen.



Koordinaten

Die Position des Kreises wird durch die Koordinaten seines Mittelpunkts entlang der X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Objekts über den Bildschirm festgelegt. Die Koordinaten beider Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Maße

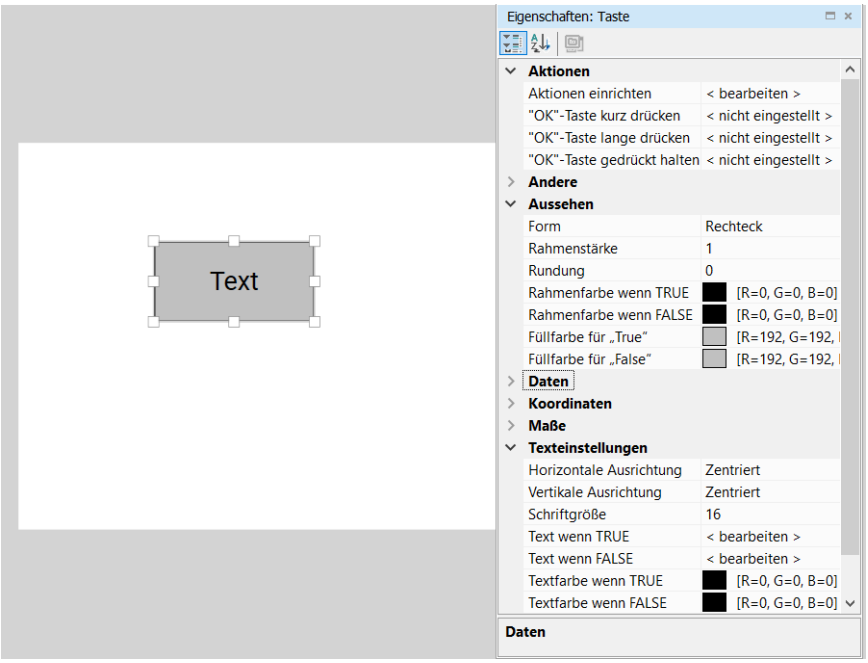
Die Abmessungen eines Elements werden durch die X- und Y-Achse bestimmt und im Eigenschaftenfenster oder durch Ändern der Elementgrenzen auf dem Bildschirm festgelegt.

Parameter

Gruppe	Parameter	Beschreibung
Aktionen	**Aktionen einrichten**	Legt die Aktion zum Navigieren zwischen Bildschirmen fest.
	“OK”-Taste kurz drücken	Legt die bei kurzem Drücken der „OK“-Taste auszuführenden Aktionen fest.
	“OK”-Taste lange drücken	Legt die bei langem Drücken der „OK“-Taste auszuführenden Aktionen fest.
	“OK”-Taste gedrückt halten	Legt die beim Gedrückthalten der „OK“-Taste auszuführenden Aktionen fest.
Andere	**Name**	Zeigt den Namen in der Liste der verwendeten Elemente im Visualisierungseditor an.
Aussehen	**Form**	Legt die Form des angezeigten Kreises fest (Kreis, Kreissektor).
	Stärke	Legt die Dicke des Umrisses der Figur fest.
	Rahmenfarbe	Legt die Farbe des Umrisses der Figur fest.
	Füllung	Legt die Füllung der Form fest.
	Füllfarbe für “True”	Legt die Füllfarbe der Form für den Wert „True“ fest.
	Füllfarbe für “False”	Legt die Füllfarbe der Form für den Wert „False“ fest.
Daten	**Variable**	Bindet eine Variable an den Parameter.
Verhalten	**Sichtbarkeit**	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.

7.7.11 Taste

Das **Taste**-Element dient zum Umschalten des Status einer BOOL-Variablen.



Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter im Eigenschaftenfenster festlegen.

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
Aktionen	**Aktionen einrichten**	Legt die Aktion zum Navigieren zwischen Bildschirmen fest.	
	“OK”-Taste kurz drücken	Legt die bei kurzem Drücken der „OK“-Taste auszuführenden Aktionen fest.	
	“OK”-Taste lange drücken	Legt die bei langem Drücken der „OK“-Taste auszuführenden Aktionen fest.	
	"OK"-Taste gedrückt halten	Legt die beim Gedrückthalten der „OK“-Taste auszuführenden Aktionen fest.	
Daten	**Variable**	Bindet eine Variable an den Parameter.	
Aussehen	**Form**	Legt die Form der anzuzeigenden Taste fest (**Rechteck** oder **Kreis**).	
	Rahmenstär-ke	Legt die Stärke des Tastenrahmens fest.	
	Rundung	Legt die Rundung der Rahmenecken fest (nur bei Rechteck). Nur verfügbar, wenn die Rechteckform ausgewählt ist.	
	Rahmenfar-be wenn "True"	Legt die Rahmenfarbe für den Wert „True“ fest.	
	Rahmenfar-be wenn "False"	Legt die Rahmenfarbe für den Wert „False“ fest.	
	Füllfarbe für "True"	Legt die Füllfarbe der Form für den Wert „True“ fest.	

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
	Füllfarbe für "False"	Legt die Füllfarbe der Form für den Wert „False“ fest.	
Koordinaten	**X**	Legt die X-Position des Elements fest (abhängig von Elementgröße und Bildschirmbreite).	Die Koordinaten können durch Verschieben des Objekts über den Bildschirm festgelegt werden.
	Y	Legt die Y-Position des Elements fest (abhängig von Elementgröße und Bildschirmhöhe).	
Texteinstel-lungen	**Horizontale Ausrichtung**	Legt die horizontale Ausrichtung des im Element angezeigten Textes fest (links, zentriert, rechts).	
	Vertikaletale Ausrichtung	Legt die vertikale Ausrichtung des im Element angezeigten Textes fest (oben, zentriert, unten).	
	Schriftgröße	Legt die Größe des im Element angezeigten Textes fest.	
	Text wenn "True"	Legt den anzuzeigenden Text bei gedrückter Taste fest.	
	Text wenn "False"	Legt den anzuzeigenden Text bei losgelassener Taste fest.	
	Textfarbe wenn "True"	Legt den anzuzeigenden Text bei losgelassener Taste fest.	
	Textfarbe wenn "False"	Legt die Textfarbe bei losgelassener Taste fest.	
Verhalten	**Sichtbarkeit**	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.	
Andere	**Name**	Zeigt den Namen des Elements in der Liste der verwendeten Elemente im Visualisierungseditor an.	
Maße	**X**	Legt die X-Position des Elements fest (ausgehend von 0, von links nach rechts; begrenzt durch Elementgröße und Bildschirmbreite).	Die Größe eines Elements kann durch Ändern der Grenzen des Objekts im Bildschirmbereich festgelegt werden,
	Y	Legt die Y-Position des Elements fest (ausgehend von 0, von oben nach unten; begrenzt durch Elementgröße und Bildschirmhöhe).	

7.7.12 Schalter

Das **Schalter**-Element dient dazu, den Zustand einer booleschen Variable zu steuern sowie ihren aktuellen Zustand anzuzeigen.

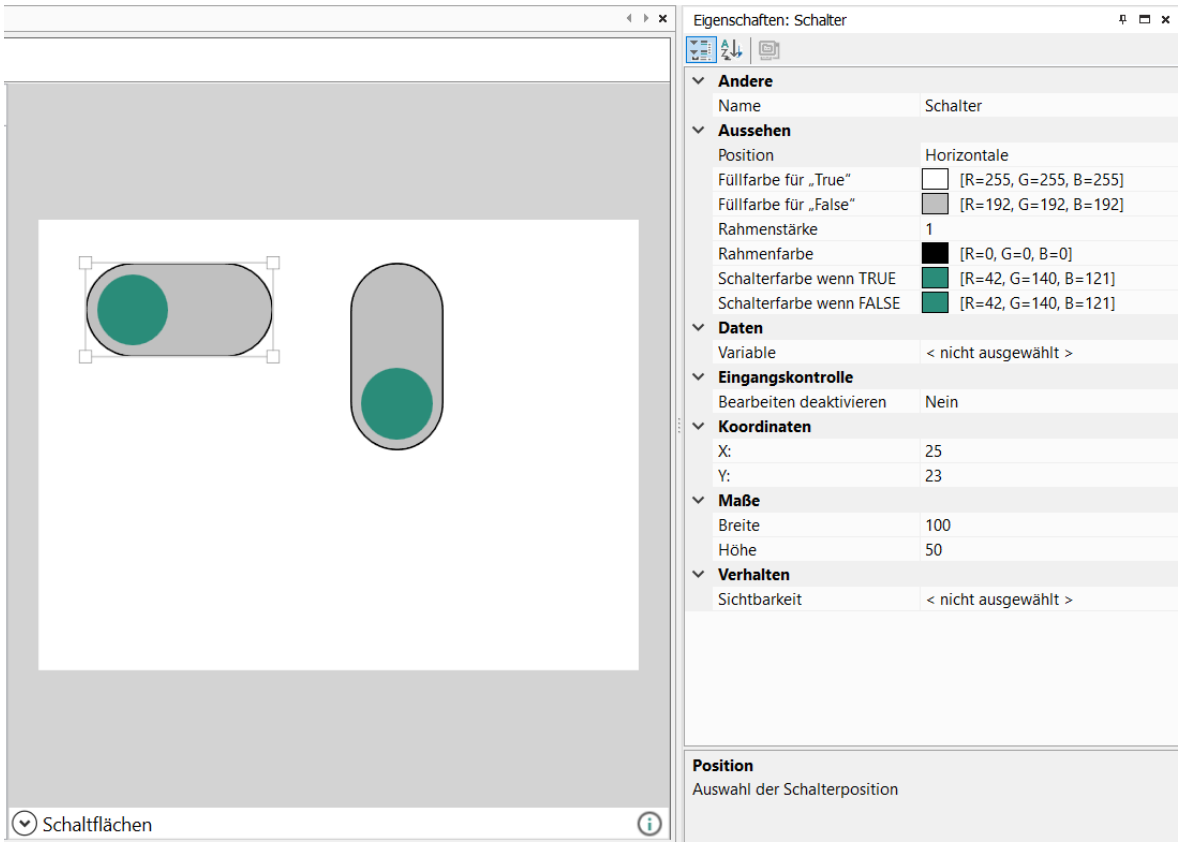


Abb. 7.3 Darstellung und Eigenschaften des Schalter-Elements

Eigenschaften einrichten

Um das Erscheinungsbild des **Schalter**-Elements auf dem Gerätebildschirm anzupassen, stehen folgende Eigenschaften zur Verfügung:

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
Aussehen	**Position**	Legt die Anordnung des Schalters fest (vertikal oder horizontal).	
	Füllfarbe für „True“	Legt die Füllfarbe des Elements im eingeschalteten Zustand fest.	
	Füllfarbe für „False“	Legt die Füllfarbe des Elements im ausgeschalteten Zustand fest.	
	Rahmenstärke	Legt die Stärke der Elementumrandung fest.	
	Rahmenfarbe	Legt die Farbe der Elementumrandung fest.	
	Schalterfarbe wenn TRUE	Legt die Farbe des Schalterknopfs im eingeschalteten Zustand fest.	
	Schalterfarbe wenn FALSE	Legt die Farbe des Schalterknopfs im ausgeschalteten Zustand fest.	
Daten	**Variable**	Bindet eine boolesche Variable an den Parameter.	
Eingangs-kontrolle	**Bearbeiten deaktivieren**	Deaktiviert die Bearbeitung des Schalters über den Gerätebildschirm.	

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
Koordinaten	**X**	Legt die X-Position des Elements fest (ausgehend von 0, von links nach rechts; begrenzt durch Elementgröße und Bildschirmbreite).	Die Koordinaten können durch Verschieben des Objekts über den Bildschirm festgelegt werden.
	Y	Legt die Y-Position des Elements fest (ausgehend von 0, von oben nach unten; begrenzt durch Elementgröße und Bildschirmhöhe).	
Verhalten	**Sichtbarkeit**	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.	
Andere	**Name**	Zeigt den Namen in der Liste der verwendeten Elemente im Visualisierungseditor an.	
Maße	**Breite**	Legt die Breite des Elements fest.	Die Größe eines Elements kann durch Ändern der Grenzen des Objekts im Bildschirmbereich festgelegt werden. Die Proportionen des Schalters werden gemäß den folgenden Regeln eingehalten: – horizontale Position: Länge = Breite x 2 – vertikale Position: Länge = Breite / 2
	Höhe	Legt die Höhe des Elements fest.	

7.7.13 Schaltergruppe

Das **Schaltgruppe**-Element dient zur Verwaltung von Parameterlisten und zur Anzeige des Status dieser Listen:

– **Optionsfeld (Radio Button)** – Erlaubt die Auswahl genau eines Parameters aus der Liste.
– **Kontrollkästchen (Checkbox)** – Erlaubt die Auswahl eines oder mehrerer Parameter aus der Liste.

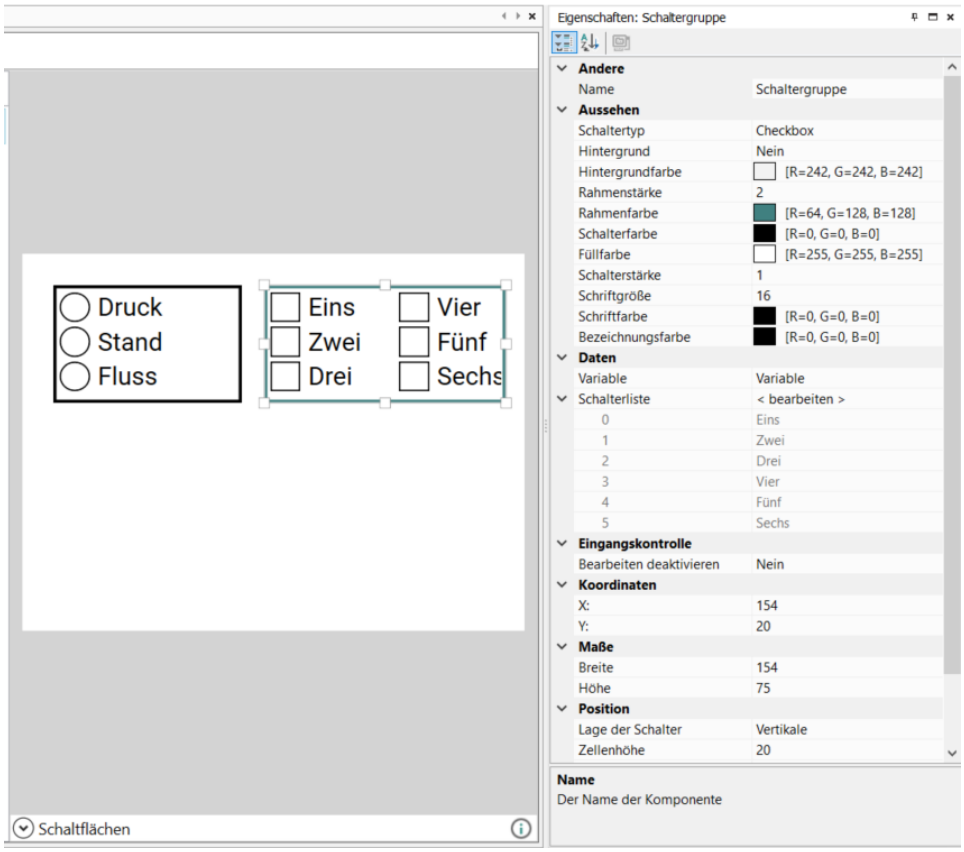
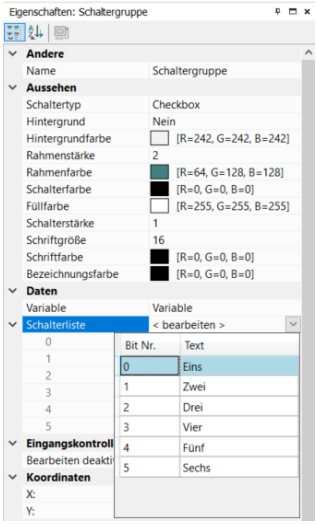
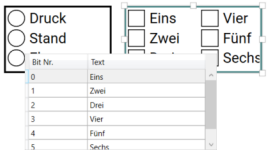


Abb. 7.4 Darstellung und Eigenschaften des Schaltgruppe-Elements

Eigenschaften einrichten

Um die gewünschte Darstellung des **Schaltgruppe**-Elements auf dem Gerätebildschirm zu konfigurieren, stehen folgende Eigenschaften zur Verfügung:

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
Aussehen	**Schaltertyp**	Legt den Schaltertyp fest (Optionsfeld oder Checkbox).	
	Hintergrund	Legt fest, ob der im Element angezeigte Text einen Hintergrund hat.	
	Hintergrundfarbe	Legt die Hintergrundfarbe des im Element angezeigten Textes fest.	
	Rahmenstärke	Legt die Stärke des Elementrahmens fest.	
	Rahmenfarbe	Legt die Farbe des Elementrahmens fest.	
	Schalterfarbe	Legt die Rahmenfarbe des Beschriftungsfelds fest.	
	Schalterstärke	Legt die Rahmenstärke des Beschriftungsfelds fest.	
	Schriftgröße	Legt die Größe des im Element angezeigten Textes fest.	

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
	Schriftfarbe	Legt die Farbe des im Element angezeigten Textes fest.	
	Beschriftungsfarbe	Legt die Farbe der Beschriftung fest.	
Daten	**Variable**	Bindet eine Integer-Variable an den Parameter. Um Zeilen hinzuzufügen, klicken Sie auf „Bearbeiten“ und verwenden Sie das Kontextmenü der sich öffnenden Tabelle:  Geben Sie den Wert der Variablen sowie den im Element anzuzeigenden Text an	Das Hinzufügen von Zeilen zur Parameterliste ist auch im Bildschirmfeld möglich: 1. Doppelklicken Sie mit der linken Maustaste auf das Element auf dem Bildschirm 2. Um Zeilen hinzuzufügen, verwenden Sie das Kontextmenü der sich öffnenden Tabelle:  3. Geben Sie den Wert der Variablen sowie den im Element anzuzeigenden Text an.
	Schalterliste (max. 32 Zeilen)	Legt die Zuordnung von Variablenwerten zu den im Element angezeigten Texten fest.	
Eingagns-kontrolle	**Bearbeiten deaktivieren**	Aktiviert oder deaktiviert die Interaktion über den Gerätebildschirm (Wechseln zwischen Listenelementen bzw. Aktivieren/Deaktivieren von Listenelementen).	
Koordina-ten	**X**	Legt die X-Position des Elements fest (von links nach rechts; begrenzt durch Elementgröße und Bildschirmbreite).	Die Größe eines Elements kann durch Ändern der Grenzen des Objekts im Bildschirmbereich festgelegt werden.
	Y	Legt die Y-Position des Elements fest (von oben nach unten; begrenzt durch Elementgröße und Bildschirmhöhe).	

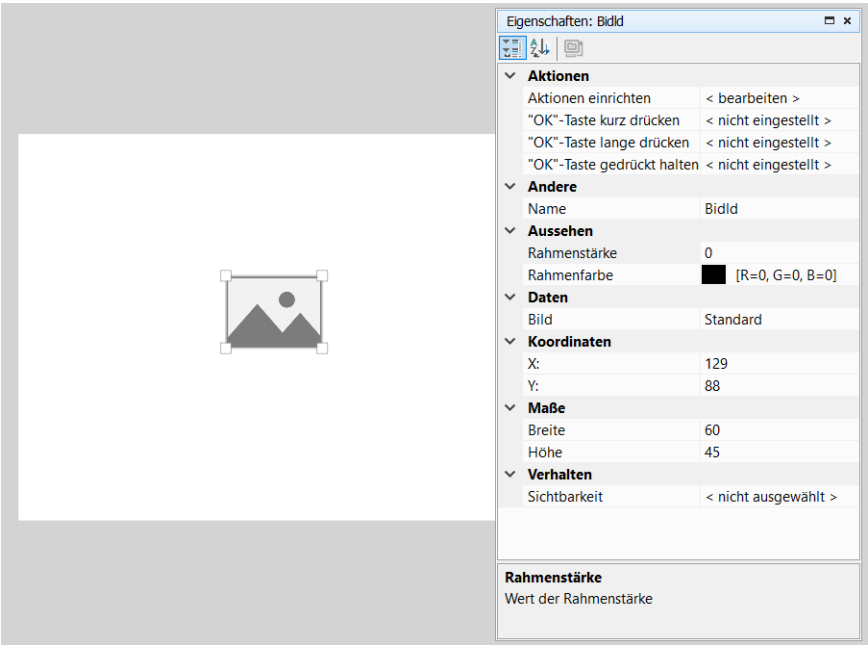
Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
Verhalten	**Sichtbarkeit**	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.	
Position	**Lage der Schalter**	Legt die Anordnung der Schalter fest (horizontal oder vertikal).	
	Zellenhöhe	Legt die Zellenhöhe zur Anordnung der Listenelemente fest.	
	Zellenbreite	Legt die Zellenbreite zur Anordnung der Listenelemente fest.	
	Anzahl der aktiven Checkboxen	Begrenzt die Anzahl gleichzeitig aktiver Checkboxen (nur bei Checkbox-Schaltertyp).	
Andere	**Name**	Zeigt den Namen in der Liste der verwendeten Elemente im Visualisierungseditor an.	
Maße	**Breite**	Legt die Breite des Elements fest.	Die Koordinaten können durch Verschieben des Objekts über den Bildschirm festgelegt werden.
	Höhe	Legt die Höhe des Elements fest.	

7.7.14 Bild

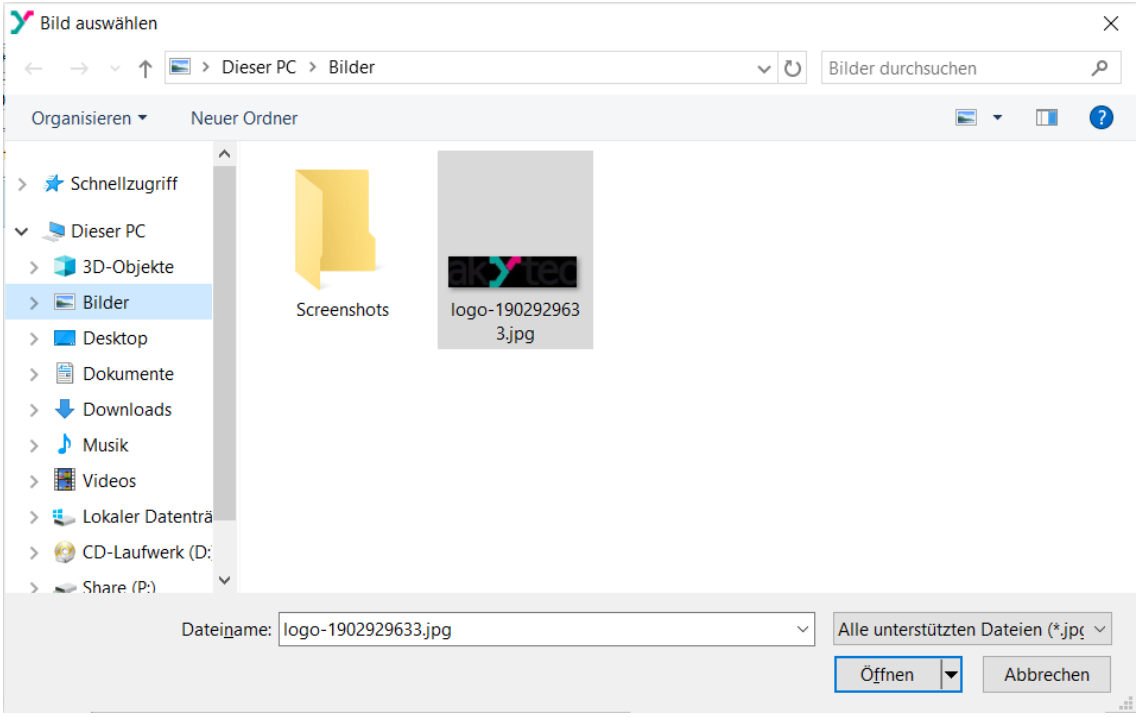
Hinzufügen eines benutzerdefinierten Bilds zu einem Projekt

Um ein Bild hinzuzufügen, sollten Sie:

1. Platzieren Sie ein **Bild**-Objekt im Arbeitsbereich.



2. Doppelklicken Sie mit der linken Maustaste auf die platzierte Figur. Das Bildauswahlfenster wird geöffnet.



HINWEIS
Zur Auswahl stehen folgende Dateitypen: (*.jpg; *.jpeg; *.jpe, *.bmp).
Die maximale Bildgröße wird durch die Bildschirmauflösung des Geräts begrenzt. Das Bild kann im Editor verkleinert werden ALP.

3. Wählen Sie ein Bild aus und klicken Sie auf **Öffnen**.

Einrichten von Eigenschaften

Für eine korrekte Anzeige auf dem Gerätebildschirm sollten Sie die Standortkoordinaten und Elementabmessungen angeben und die Parameter im Eigenschaftenfenster festlegen.

Koordinaten

Die Position des Polygons wird durch die Koordinaten seines Mittelpunkts entlang der X- und Y-Achse bestimmt. Die Koordinaten werden im Eigenschaftenfenster oder durch Verschieben des Objekts auf dem Bildschirm festgelegt. Die Koordinaten entlang beider Achsen beginnen bei 0:

– Entlang der X-Achse – von links nach rechts. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmbreite bestimmt.
– Entlang der Y-Achse – von oben nach unten. Der endgültige Wert wird durch die Größe des Elements und die Bildschirmhöhe bestimmt.

Maße

Die Maße eines Elements werden durch die X- und Y-Achse bestimmt und im Eigenschaftenfenster oder durch Ändern der Elementgrenzen auf dem Bildschirm festgelegt.

Parameter

Gruppe	Parameter	Beschreibung
Aktionen	**Aktionen einrichten**	Legt die Aktion zum Navigieren zwischen Displayformen fest.
	“OK”-Taste kurz drücken	Legt die bei kurzem Drücken der „OK“-Taste auszuführenden Aktionen fest.

Gruppe	Parameter	Beschreibung
	“OK”-Taste lange drücken	Legt die bei langem Drücken der „OK“-Taste auszuführenden Aktionen fest.
	"OK"-Taste gedrückt halten	Legt die beim Gedrückthalten der „OK“-Taste auszuführenden Aktionen fest.
Andere	**Name**	Zeigt den Namen in der Liste der verwendeten Elemente im Visualisierungseditor an.
Aussehen	**Rahmenstärke**	Legt die Stärke des Bilderrahmens fest.
	Rahmenfarbe	Legt die Farbe des Bilderrahmens fest.
Daten	**Bild**	Legt die anzuzeigende Bilddatei fest.
Verhalten	**Sichtbarkeit**	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.

HINWEIS
Beim Kopieren und Verknüpfen desselben Bildes mit mehreren Visualisierungselementen wird das zusätzliche Volumen des Benutzervisualisierungs-ROM nicht belegt (Anzeige **Benutzervisualisierungs-ROM** in der Statusleiste).

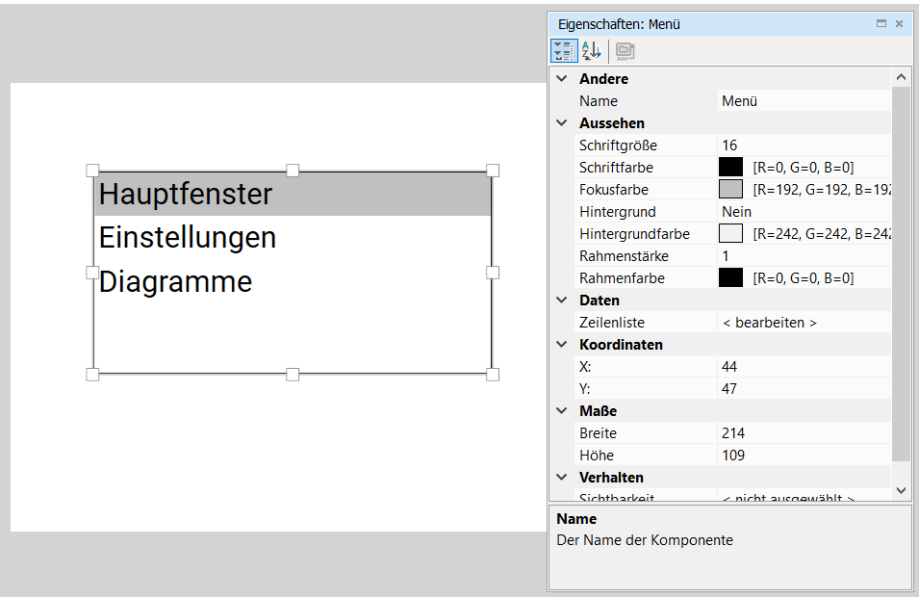
7.7.15 Menue

Das Element **Menü** dient der Navigation zwischen Displayformen. Jeder Menüpunkt ist mit einer Displayform verknüpft.

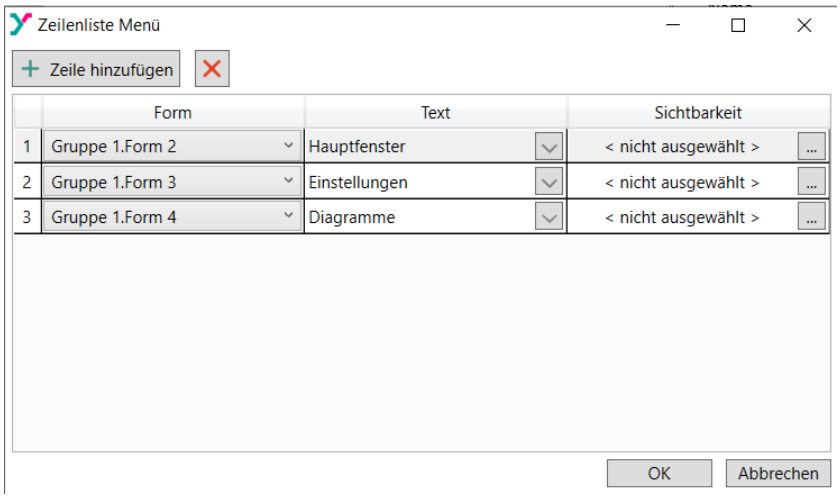
Ein Menü zu einem Projekt hinzufügen

Fügen Sie der Displayform ein Menü-Objekt wie folgt hinzu:

1. Platzieren Sie ein **Menü**-Objekt im Arbeitsbereich..



2. Doppelklicken Sie mit der linken Maustaste auf die platzierte Figur. Das Fenster zur Linienbearbeitung wird geöffnet.



3. Fügen Sie die erforderliche Anzahl Zeilen hinzu.

HINWEIS
Maximale Zeilenanzahl: 128

4. Geben Sie in der Spalte „Form“ an, zu welchem Displayform bei Auswahl eines Menüpunkts gewechselt werden soll.

HINWEIS
Zur Auswahl stehen alle Displayforme des Projekts außer dem aktuellen.

5. Der Name des ausgewählten Bildschirms wird automatisch in der Spalte **Text** angezeigt. Um den Text zu ändern, klicken Sie auf die Markierung am Ende der Zeile, und geben Sie den Text, der für den Menüpunkt angezeigt werden soll, in das sich öffnende Feld ein.
6. Legen Sie die Sichtbarkeit der hinzugefügten Zeilen fest, indem Sie Variablen in der Spalte „Sichtbarkeit“ binden.
7. Klicken Sie auf die Schaltfläche **OK**.

Liste der Eigenschaften

Folgende Eigenschaften stehen zur Konfiguration des Menüpunkts auf dem Gerätebildschirm zur Verfügung:

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
Aussehen	**Schriftgröße**	Legt die Größe des im Element angezeigten Textes fest.	
	Schriftfarbe	Legt die Farbe des im Element angezeigten Textes fest.	
	Fokusfarbe	Legt die Farbe des ausgewählten Menüpunkts fest.	
	Hintergrund	Legt den Hintergrund des im Element angezeigten Textes fest.	
	Hintergrund-farbe	Legt die Hintergrundfarbe des im Element angezeigten Textes fest.	
	Rahmenstär-ke	Legt die Stärke des Elementrahmens fest.	
	Rahmenfar-be	Legt die Farbe des Elementrahmens fest.	

Gruppe	Parameter	Beschreibung	Zusätzliche Funktionen
Daten	**Zeilenliste**	Öffnet das Bearbeitungsfenster für die Zeilenliste.	
Koordinaten	**X**	Legt die X-Position des Elements fest (von links nach rechts; begrenzt durch Elementgröße und Bildschirmbreite).	Die Koordinaten können durch Verschieben des Objekts über den Bildschirm festgelegt werden.
	Y	Legt die Y-Position des Elements fest (von oben nach unten; begrenzt durch Elementgröße und Bildschirmhöhe).	
Verhalten	**Sichtbarkeit**	Steuert die Sichtbarkeit des Elements über eine gebundene BOOL-Variable.	
	Standardmäßig fokussiert	Legt fest, ob der Menüpunkt nach dem Bildschirmwechsel standardmäßig fokussiert ist.	
Andere	**Name**	Zeigt den Namen in der Liste der verwendeten Elemente im Visualisierungseditor an.	
Maße	**Breite**	Legt die Breite des Elements entlang der X-Achse fest.	Die Größe eines Elements kann durch Ändern der Grenzen des Objekts im Bildschirmbereich festgelegt werden.
	Höhe	Legt die Höhe des Elements entlang der Y-Achse fest.	

7.7.16 Diagramm

Das Element **Diagramm** dient dazu, die Änderung von Variablenwerten grafisch auf dem Gerätedisplay anzuzeigen:

- Bis zu vier Variablen können gleichzeitig angezeigt werden (Datentypen – int und float).
- Das Abfragezeitintervall zum Zeichnen des Diagramms kann konfiguriert werden (der maximal zulässige Wert beträgt 23h59m59s).
- Der Abfragezeitraum wird automatisch berechnet, hängt vom Maximalwert der X-Achse, der Breite des **Diagramm**-Elements und der Rahmenstärke ab und darf nicht weniger als 1 Sekunde betragen.
- Die Anzahl der Slices wird durch die folgende Formel bestimmt: *Anzahl der Slices = max. X / t poll*, wobei:
  - max. X – Zeitbereich der Anzeige
  - t poll – Zeit zwischen zwei Messungen

HINWEIS
Die Diagrammdarstellung ist nur für Geräte mit Farb-LCD verfügbar. Die Diagrammdarstellung im Simulationsmodus wird nicht unterstützt.

Um der Displayform ein Diagramm hinzuzufügen, wählen Sie das **Diagramm**-Element in der Bibliothek aus und platzieren Sie es im Arbeitsbereich.
Die Koordinatenachsen werden mit angezeigter Skala und Raster angezeigt. Die X-Achse ist die Zeitachse, die Y-Achse ist die Achse der Variablenwerte (udint oder float).

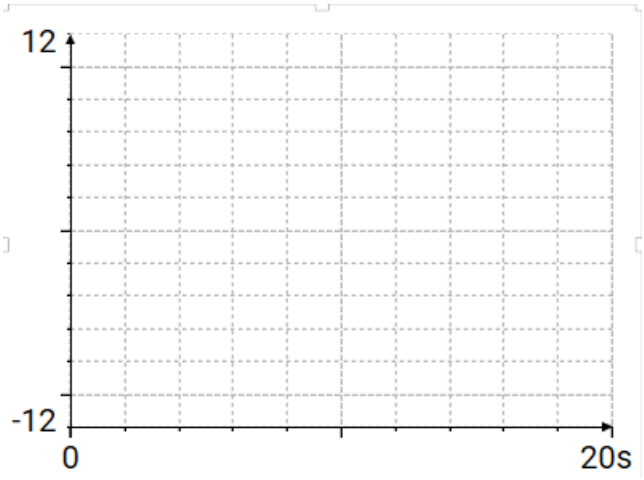


Abb. 7.5 Koordinatenachsen mit Standardwerten

Um Parameter zu konfigurieren und Variablen hinzuzufügen, verwenden Sie das **Eigenschaftenfeld**:

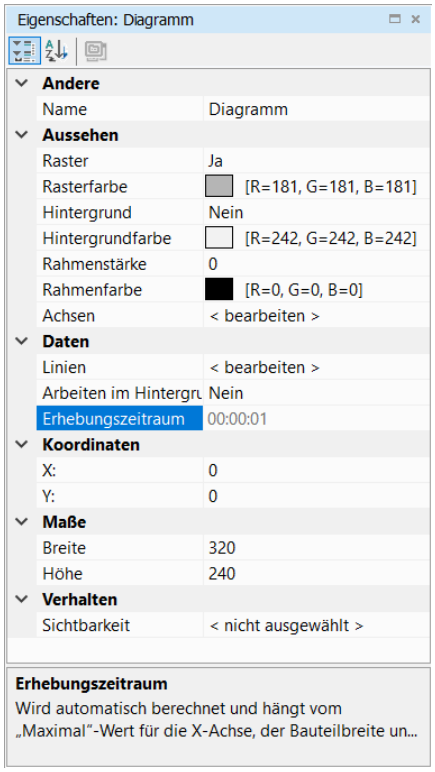


Abb. 7.6 Diagrammparameter

Passen Sie das **Aussehen** des Diagramms an:

– Vorhandensein und Farbe des Rasters
– Vorhandensein und Farbe des Hintergrunds
– Achsen:

  Wählen Sie die Achsenzeile aus und klicken Sie mit der linken Maustaste auf <*bearbeiten*> und dann auf die angezeigte **…** Schaltfläche. Das Fenster „Achsendiagramm" wird geöffnet:

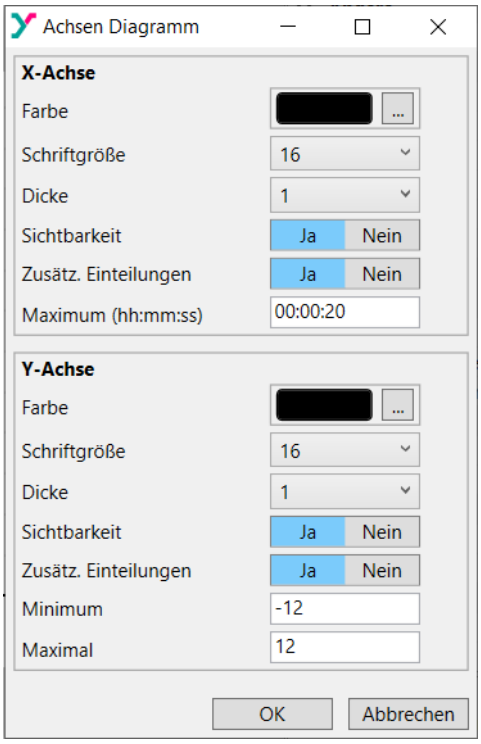


Abb. 7.7 Achsenfenster

Passen Sie bei Bedarf die Farbe und Breite der Achsen, die Schriftgröße, die Sichtbarkeit der Achsen und zusätzliche Teilstriche an. Geben Sie den gewünschten Maximalwert der X-Achse im Format hh:mm:ss ein. Der maximal zulässige Wert ist 23:59:59. Der Standardwert beträgt 20 Sekunden.

HINWEIS
Minimalwert der X-Achse = 0 – nicht editierbarer Parameter

Geben Sie die Minimal- und Maximalwerte für die Y-Achse ein. Standardwert: -12, 12.

Wenn Sie einen falschen Wert eingeben, wird eine Fehlermeldung angezeigt und die Schaltfläche „OK“ wird inaktiv:

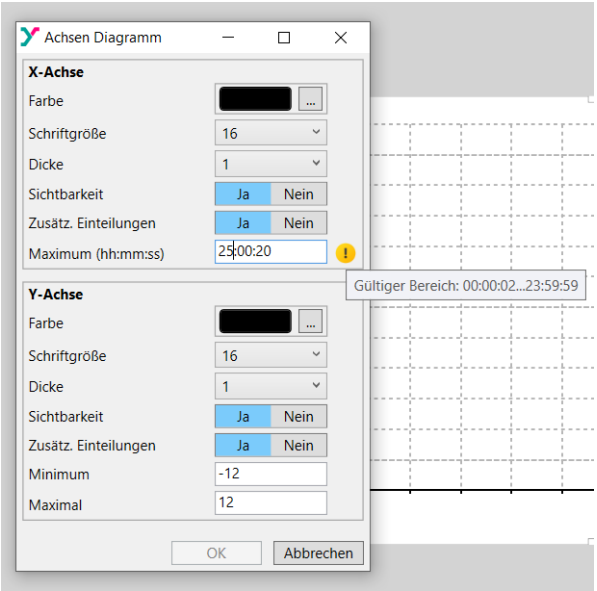


Abb. 7.8 Dateneingabefehler

Um Variablen zum Diagramm hinzuzufügen, wählen Sie im Bereich **Daten** des Fensters „Diagrammeigenschaften“ die Linie „Linien“ aus und klicken Sie mit der linken Maustaste auf <*bearbeiten*> und dann auf die angezeigte Schaltfläche ... oder doppelklicken Sie auf das Diagrammfeld. Das Fenster „Linienliste“ wird geöffnet:

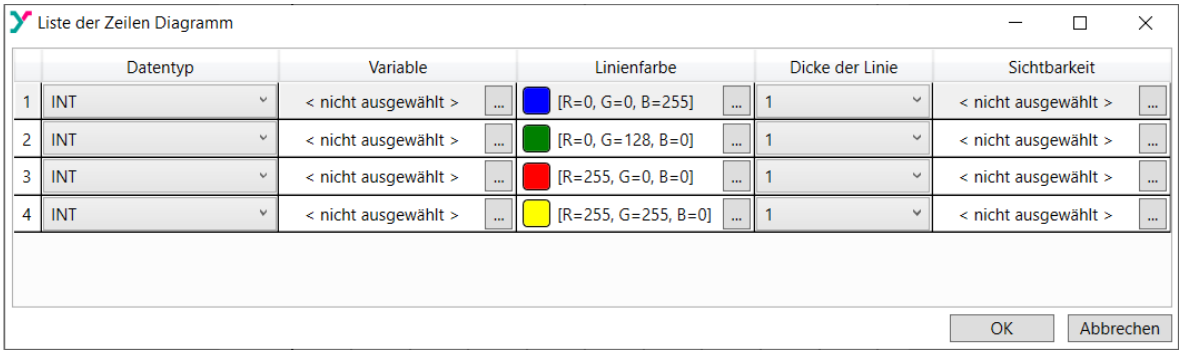


Abb. 7.9 Linienliste – Diagrammfenster

Wählen Sie im sich öffnenden Fenster Folgendes aus:

– Variablentyp – Ganzzahl/Gleitkommazahl
– Variable
– Farbe und Stärke der Linie
– Sichtbarkeit – Sie können eine BOOL-Variable aus der Kategorie „Standardvariablentabellen“ auswählen.

HINWEIS
Maximale Anzahl von Variablen in einem Diagramm: 4

Um die Anzahl der hinzugefügten Variablen sowie die Einstellungen für die Darstellung der Linien im Diagramm leichter verständlich zu machen, werden die Linien mit einem Versatz entlang der Y-Achse relativ zueinander angezeigt.
Die Option „Arbeiten in Hintergrund (Ja/Nein)“ im Abschnitt **Daten** beeinflusst die Datenspeicherung beim Wechsel zu einer anderen Displayform .
Der Erhebungszeitraum ist ein nicht editierbarer Parameter. Der Zeitraum hängt vom Maximalwert der X-Achse, der Breite des Graph-Elements und der Rahmenstärke ab.

HINWEIS
Der Erhebungszeitraum darf nicht weniger als 1 Sekunde betragen.

Die Position des Diagramms wird durch die Koordinaten der oberen linken Ecke entlang der X- und Y-Achse bestimmt. Die Koordinaten werden durch Verschieben des Objekts über den Bildschirm festgelegt. Wenn Sie eine genaue Position des Diagramms auf dem Gerätedisplay benötigen, legen Sie die Koordinaten in der Propery Box im Bereich **Koordinaten** fest.
Konfigurieren Sie bei Bedarf im Abschnitt **Verhalten** die Sichtbarkeit des gesamten Diagramms in Abhängigkeit von der BOOL-Variable.
Sie können den Namen eines Diagramms im Abschnitt **Andere** des Eigenschaftenfensters ändern.
Um das Diagramm zu skalieren, bewegen Sie die Maus über die Eck- oder Seitenmarkierung, halten Sie die linke Maustaste gedrückt und stellen Sie die gewünschte Diagrammgröße auf dem Bildschirm ein. Sie können die Größe auch im Bereich **Maße** ändern: Breite und Höhe werden in Pixeln eingestellt.
Nach dem *Übertragen des Programms* wird auf dem Gerätedisplay ein Diagramm mit den hinzugefügten Variablen angezeigt:

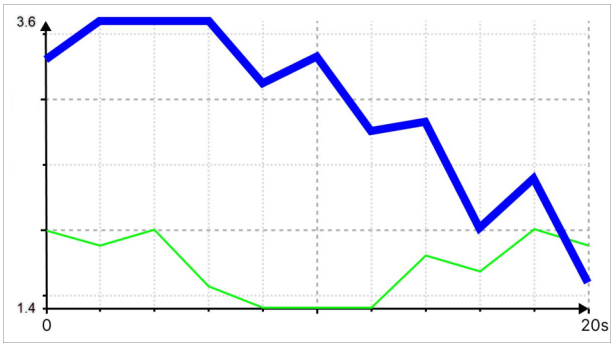


Abb. 7.10 Diagramm auf dem Gerätedisplay

HINWEIS
Wenn beim Anzeigen eines Diagramms auf dem Gerätebildschirm der Parameterwert den Minimal- oder Maximalwert der Y-Achse überschreitet, bewegt sich die Linie im Diagramm weiter entlang der entsprechenden Grenze.

Wird eine dem Diagramm hinzugefügte Variable aus dem Projekt gelöscht, erscheint ein Informationsfenster:

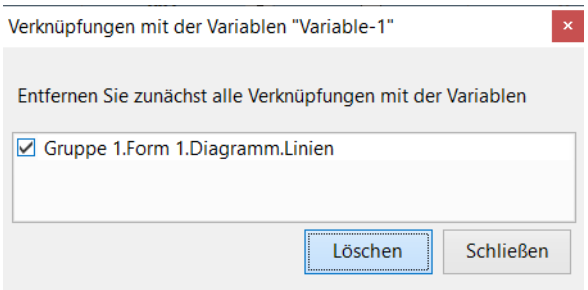


Abb. 7.11 Informationsfenster

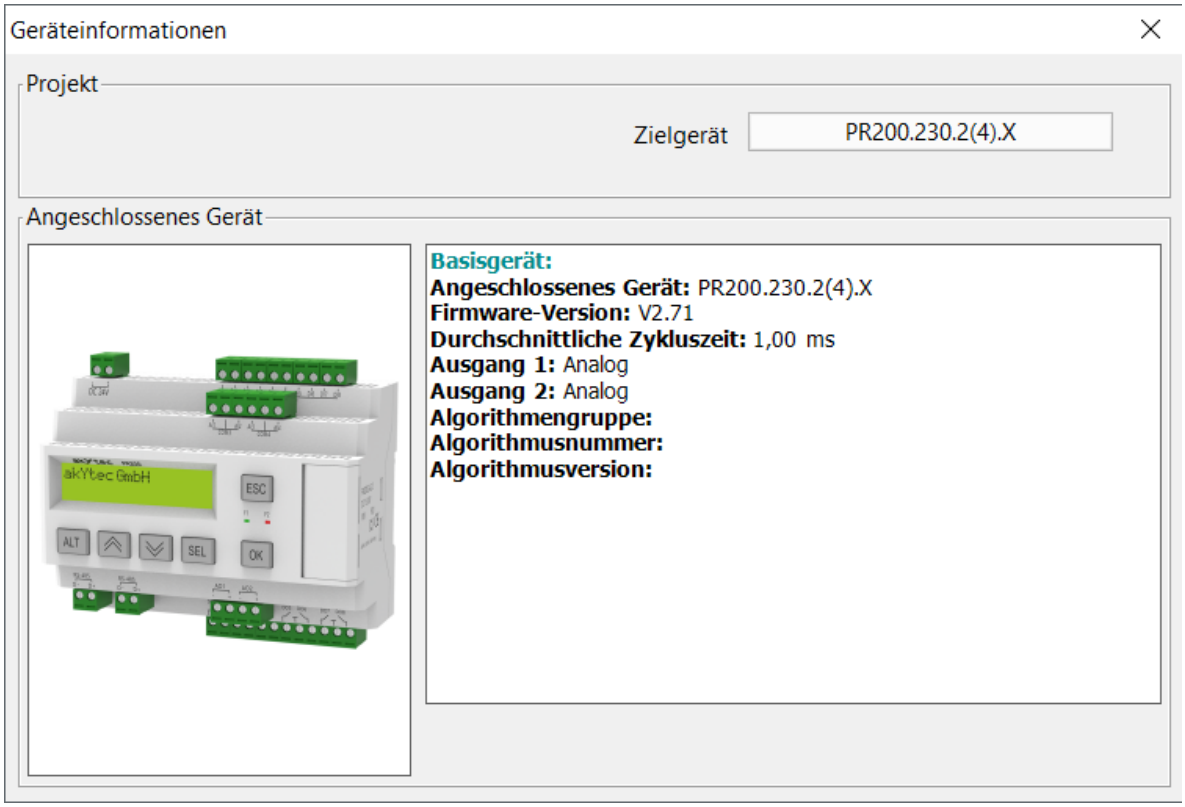
Wenn Sie im Liniendiagrammfenster **Löschen** auswählen, wird die Variablenverknüpfen gelöscht.

8 Geraet

In diesem Abschnitt werden die Betriebsfunktionen und die Konfiguration des Geräts beschrieben:
– *Geräteinformationen*
– *Zykluszeit*
– *Firmware-Update*
– *Kalibrierung*

8.1 Geraeteinformationen

Um Informationen zur Software, zum Zielgerät und zum angeschlossenen Gerät anzuzeigen, verwenden Sie den Menüpunkt ***Gerät*** → ***Geräteinformationen…*** oder das Symbol in der Symbolleiste.
Es erscheint ein Fenster mit Informationen zum angeschlossenen Gerät:



Das Fenster ***Geräteinformationen*** enthält folgende Informationen:
Zielgerät – Zeigt das Gerät an, für das das Projekt erstellt wurde.
Angeschlossenes Gerät – Zeigt Informationen über das Gerät an, das mit dem PC verbunden ist. Alternativ kann der Typ jedes Ausgangs entsprechend der Hardware manuell im ***Eigenschaftenfeld*** geändert werden.

Informationen zum Gerät auf der neuen Hardwareplattform

Bei Geräten auf der neuen Plattform unterscheiden sich die im Fenster angezeigten Informationen. Die Projektinformationen sind wie folgt:
– ***Ausgewähltes Gerätemodell*** – Zeigt das Gerätemodell und die Gerätemodifikation an, die beim Erstellen des Projekts ausgewählt wurden.

Informationen zum verbundenen Gerät:
– ***Gerätename*** – Zeigt das Modell und die Modifikation des angeschlossenen Geräts an.
– ***Firmware-Version*** – Zeigt die Firmware-Version des angeschlossenen Geräts an.
– ***S/N*** – Zeigt die eindeutige Gerätekennung an.

– **PRM-Slot. Modulname** – Zeigt das Modell des *Erweiterungsmoduls* an, das an das Gerät angeschlossen ist.
– **PRM-Slot. Modul-Firmware-Version** – Zeigt die Firmware-Version des *Erweiterungsmoduls* an, das an das Gerät angeschlossen ist.

8.2 Zykluszeit

Die Zykluszeit ist die Zeit, die zum Abschließen des Betriebszyklus des Geräts benötigt wird, und zwar:
– Abfragen des Zustands der physischen Eingänge und Kopieren der Werte in Speicherzellen
– Programmverarbeitung
– Lesen und Schreiben von Programmnetzwerkvariablen
– Ausgabe der Programmergebnisse auf die physischen Ausgänge

Die Standardzykluszeit beträgt **1 ms**. Das Gerät passt die Zykluszeit je nach Komplexität des Programms an.
Bedingungen zur Erhöhung der Zykluszeit:
– Steigende Komplexität des Algorithmus, zum Beispiel durch viele Funktionsbausteine und Makros
– Verwendung einer großen Anzahl von Netzwerkvariablen im Programm
– Nutzung zahlreicher Datenkontrollen über die Geräteanzeige

Die Zykluszeit kann vom Benutzer nicht eingestellt werden. Ist das Gerät mit einem Display ausgestattet, kann die aktuelle Zykluszeit im Systemmenü eingesehen werden. Ist das Gerät an einen PC angeschlossen, kann die Zykluszeit im Fenster *Geräteinformationen* eingesehen werden.

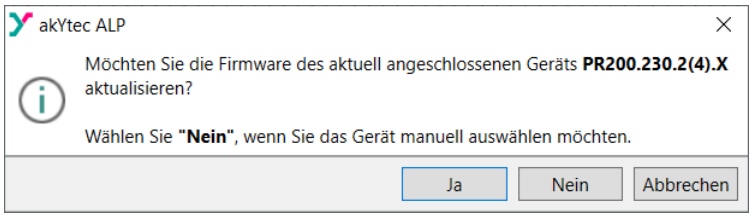
8.3 Firmware-Update/-Reparatur

Eine neue ALP-Version kann eine neue Firmware-Version für das angeschlossene Gerät oder Erweiterungsmodul enthalten. In diesem Fall 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.

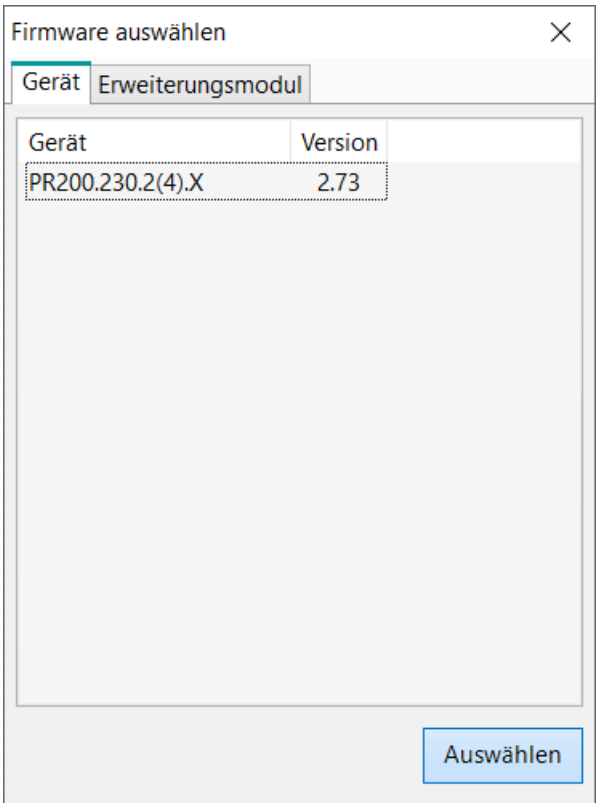
HINWEIS
Stellen Sie während des Update-Vorgangs sicher, dass das Gerät und die Erweiterungsmodule mit Strom versorgt werden. Stellen Sie außerdem sicher, dass die Verbindung zwischen PC, Gerät und Erweiterungsmodulen stabil ist.

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

HINWEIS
Das Benutzerprogramm wird durch das Firmware-Update nicht beeinträchtigt.



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. Das geöffnete Fenster hat zwei Registerkarten: ***Gerät*** und ***Erweiterungsmodul***. Auf diese Weise kann ein erzwungenes Firmware-Update durchgeführt werden.
Klicken Sie auf Auswählen, um die Auswahl zu bestätigen und den Update- (Reparatur-)Prozess zu starten. Nach Abschluss des Updates wird eine Meldung über das Update-Ergebnis angezeigt.



Erzwungenes Firmware-Update/-Reparatur

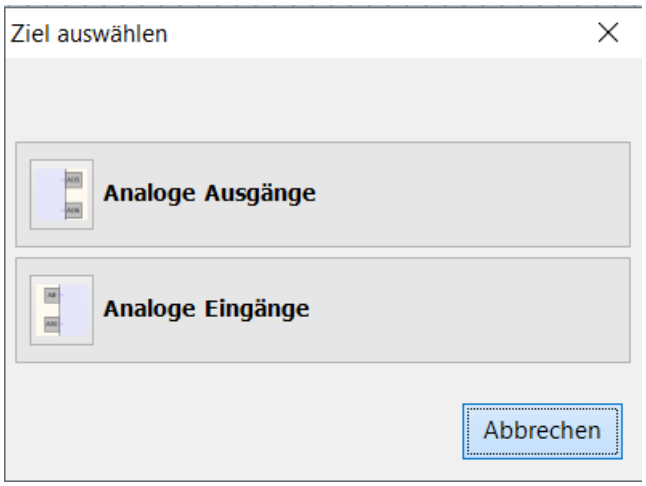
Wenn die Firmware beschädigt ist (siehe jeweilige Bedienungsanleitung, Tabelle „Fehleranzeige“) und keine automatische Geräteerkennung möglich ist, sollte ein erzwungenes Firmware-Update durchgeführt werden. Gehen Sie dazu wie folgt vor:

1. Stellen Sie das Gerät in den erzwungenen Download-Modus (siehe Benutzerhandbuch des Geräts).
2. Wählen Sie den Menüpunkt ***Gerät > Firmware-Update***. Anschließend werden Listen mit verfügbaren Geräten und Erweiterungsmodulen angezeigt.
3. Wählen Sie das gewünschte Gerät oder Erweiterungsmodul aus.
4. Klicken Sie auf ***Auswählen***, um die Auswahl zu bestätigen und den Aktualisierungsvorgang (Reparaturvorgang) zu starten.

Nach Abschluss der Aktualisierung wird die Meldung zum Update-Ergebnis angezeigt.
Wenn das Gerät und das Erweiterungsmodul inkompatible Firmware-Versionen haben und das Anwenderprogramm auf das Gerät geladen wird, ohne dass das Erweiterungsmodul angeschlossen ist, kann dies dazu führen, dass ein Fehler am Erweiterungsmodul angezeigt wird. Um den Fehler zu beheben, führen Sie wie beschrieben ein erzwungenes Firmware-Update für das Erweiterungsmodul durch und überspringen Sie dabei Schritt 1.

8.4 Kalibrierung

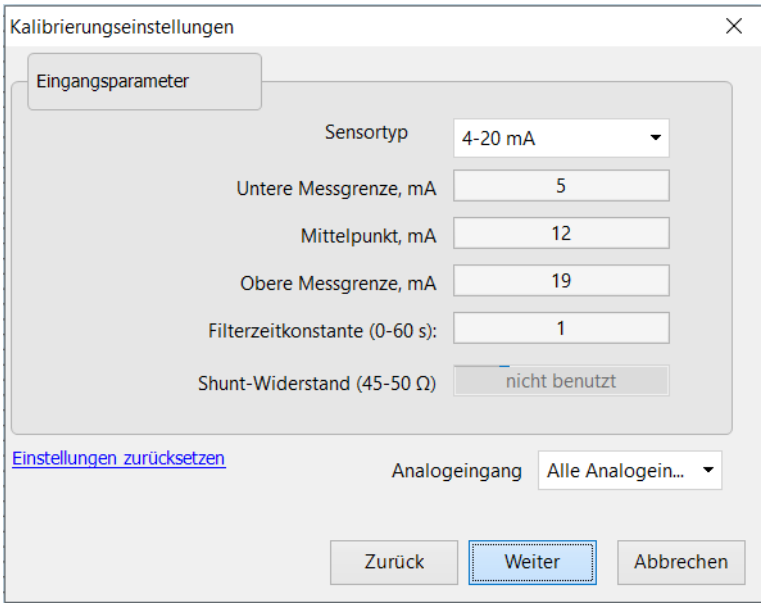
In diesem Abschnitt werden nur allgemeine Informationen zur Kalibrierung analoger Ein- und Ausgänge gegeben. Detaillierte Informationen zur Kalibrierung finden Sie im Benutzerhandbuch des jeweiligen Geräts.
Wenn eine Kalibrierung der analogen Ein- oder Ausgänge notwendig ist, nutzen Sie den Menüpunkt ***Gerät → Kalibrierung…***.
Der Punkt ist nur aktiv, wenn ein Gerät angeschlossen ist. Wählen Sie im geöffneten Dialog das Kalibrierungsziel (Eingänge oder Ausgänge) aus.

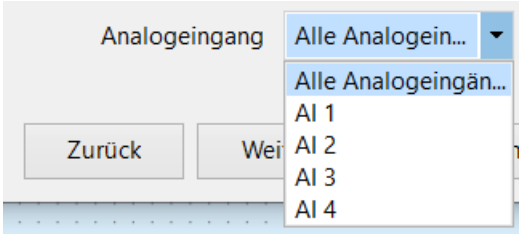


Nach der Auswahl des Kalibrierziels wird die Ausführung des Programms im Gerät gestoppt. Nach erfolgreichem Abschluss der Kalibrierung startet das Programm erneut.

8.4.1 Eingangskalibrierung

Um Eingänge zu kalibrieren, schließen Sie eine Referenzsignalquelle an sie an. Starten Sie die Kalibrierung, wählen Sie den Signaltyp aus, der an den Eingang angeschlossen ist, und legen Sie die Kalibrierungsparameter im geöffneten Dialogfeld fest.

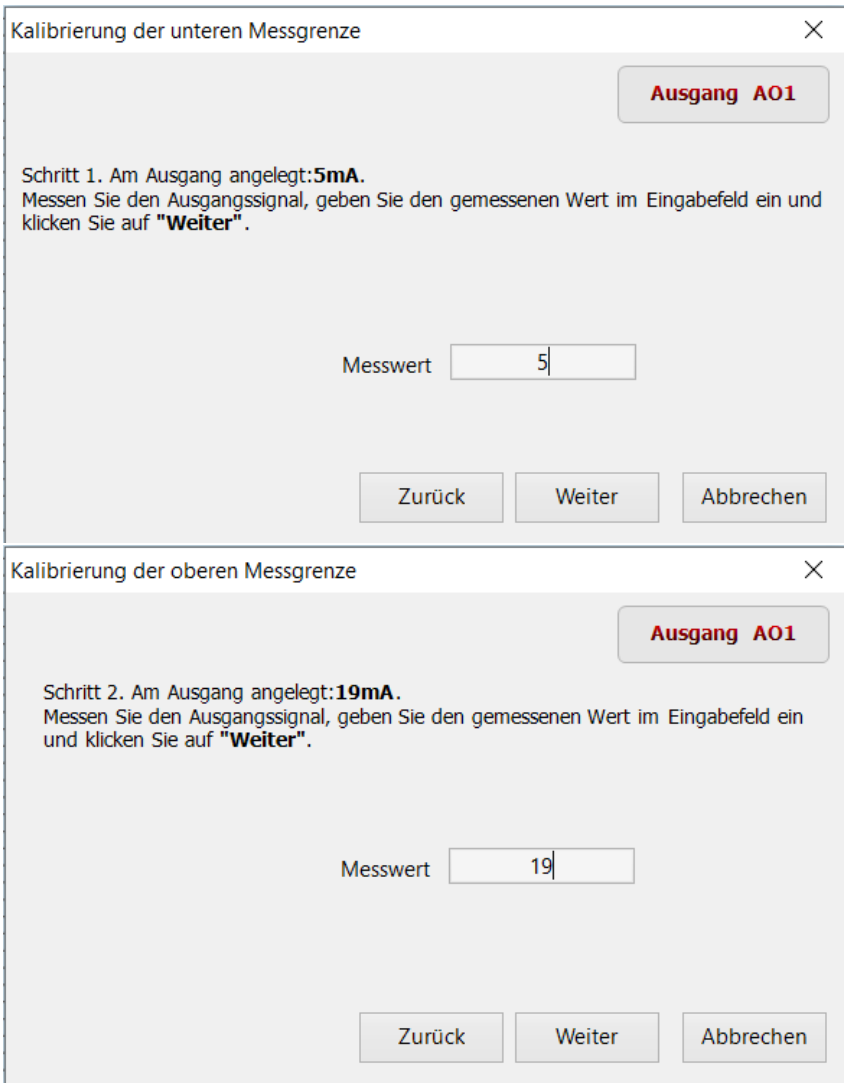


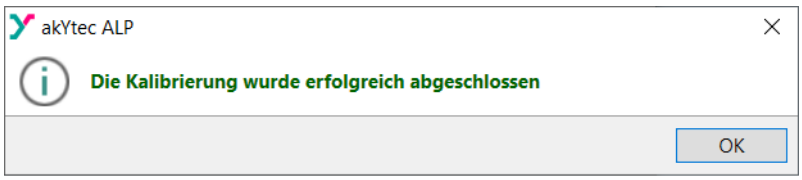


Über den Punkt ***Einstellungen zurücksetzen*** werden die Standardeinstellungen für die Kalibrierung übernommen.
Wählen Sie über die Liste ***Eingang auswählen*** den zu kalibrierenden Eingang aus, klicken Sie auf die Schaltfläche ***Weiter*** und folgen Sie den Anweisungen.

8.4.2 Ausgangskalibrierung

Bevor Sie einen analogen Ausgang kalibrieren, legen Sie das entsprechende Messgerät bereit, starten Sie die Kalibrierung und folgen Sie den Anweisungen. Messen Sie das Signal am Ausgang, der oben rechts im Fenster angezeigt wird, und geben Sie den Wert in das Eingabefeld ein. Verfahren Sie bei Bedarf mit den anderen Ausgängen genauso. Nach Abschluss der Kalibrierung erscheint die Meldung über das Kalibrierergebnis.





8.5 Zielgeraet aendern

Die Funktion **Zielgerät ändern** dient dazu, ein Projekt auf eine andere Ausführung eines Geräts aus derselben Generation zu übertragen.
Die Änderung des Zielgeräts erfolgt nach folgenden Regeln:

– Der Arbeitsbereich passt sich automatisch an Änderungen in der Anzahl der Eingänge/Ausgänge an.
– Das vom Benutzer konfigurierte Ein-/Ausgabe-Layout bleibt erhalten. Neue E/A werden nach den bereits vorhandenen E/A des Originalprojekts hinzugefügt.
– Verbindungen von Ein-/Ausgängen, deren Datentyp sich geändert hat, werden aufgehoben.

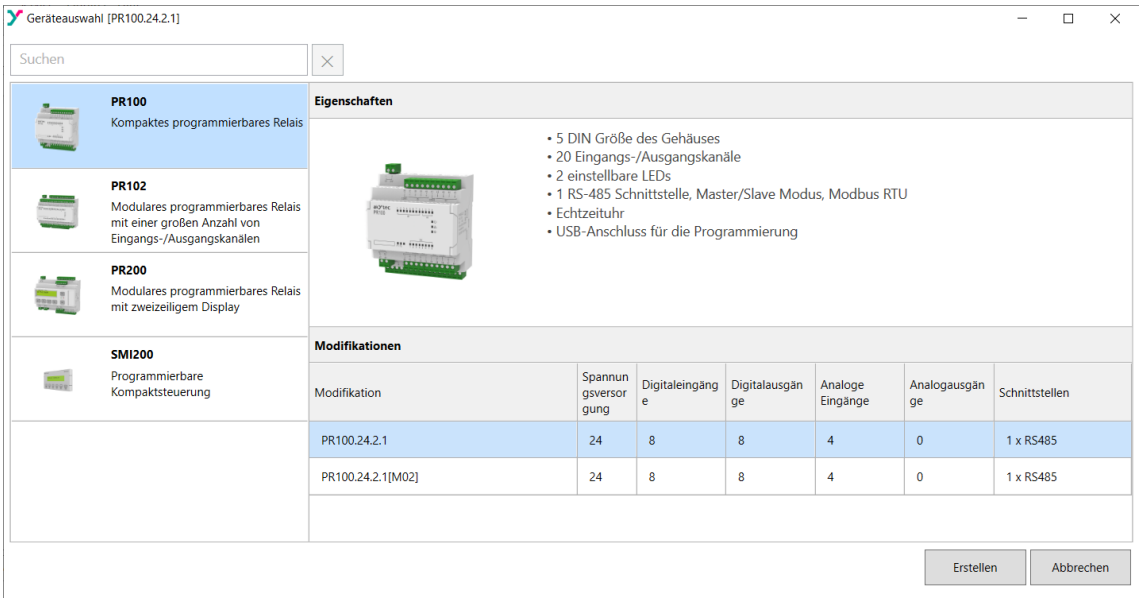
– Wenn die Anzahl der Ein-/Ausgänge zunimmt, sich der Datentyp des ursprünglichen E/A-Satzes jedoch nicht ändert, bleiben die Verbindungen bestehen.
– Wenn die Anzahl der Ein-/Ausgänge im Vergleich zum Original abnimmt, werden die Verbindungen der entfernten E/A aufgehoben.
– Wenn im Originalprojekt Erweiterungsmodule hinzugefügt wurden, werden diese auf die neue Plattform übertragen – die dafür bestehenden Verbindungen bleiben erhalten.
– Die Einstellungen für analoge Ein-/Ausgänge werden übertragen (sofern das Zielgerät über entsprechende analoge Ausgänge verfügt).
– Kommunikationsschnittstellen werden unverändert übertragen. Falls das Zielgerät für die Projektübertragung eine Schnittstelle nicht unterstützt, öffnet sich ein Fenster mit der Information, dass die Schnittstelle, Netzwerkvariablen und Verbindungen zu Variablen gelöscht werden.
– Alle Bildschirme und ihre Einstellungen werden von einer Zielplattform auf eine andere übertragen. Falls das Zielgerät für die Projektübertragung kein Display besitzt, öffnet sich ein Fenster mit Informationen über die Entfernung der Verbindungen zu Variablen, die mit den Visualisierungselementen verknüpft sind.

HINWEIS
Beim Wechsel der Zielplattform von PR225 auf PR205 werden die Visualisierungselemente proportional zur Änderung der Bildschirmgröße skaliert. Beim Wechsel der Zielplattform von PR205 auf PR225 werden die Visualisierungselemente ohne Skalierung übertragen.

– Alle Variablen werden übertragen. Falls im Projekt Variablen an Parameter gebunden sind, die im Zielgerät nicht vorhanden sind, öffnet sich ein Fenster mit Informationen über die Entfernung der Verbindungen zu Variablen, die mit den Parameter verknüpft sind.
– Wenn die Zielplattform einen im Quellprojekt enthaltenen Funktionsbaustein (FB) nicht unterstützt, erscheint ein Hinweisfenster, das über die Notwendigkeit zum Löschen dieses FBs informiert.

Um das Zielgerät zu ändern, gehen Sie wie folgt vor:

1. Öffnen Sie das Projekt, das auf ein anderes Gerät oder Gerätemodel übertragen werden soll.
2. Wählen Sie im Menü **Datei** → **Zielgerät ändern**. ALP wird Sie auffordern, das Projekt vor dem Plattformwechsel zu speichern. Anschließend öffnet sich ein Menü mit einer Liste der für den Plattformwechsel verfügbaren Gerätemodellen:



3. Wählen Sie ein neues Gerät aus. Klicken Sie auf die Schaltfläche **OK**.
4. Überprüfen und stellen Sie gegebenenfalls aufgehobene Verbindungen wieder her. Die Programmfunktionalität kann im *Simulator 4.5* geprüft werden.
5. Speichern Sie das geänderte Projekt.

9 Plugins

Plugins werden verwendet, um komplexe Projekte zu erstellen und sie in andere akYtec-Dienste zu integrieren.
Plugins in ALP:

– *Replikationsmaster*. Mit dem **Replikationsmaster** können Sie ein Programm ohne ALP auf das Gerät laden und es vor dem Kopieren schützen.
– *Export des Geräts in die akYtec Cloud*. Ermöglicht den Export der Netzwerkvariablenkonfiguration für die Abfrage im akYtec Cloud-Dienst.

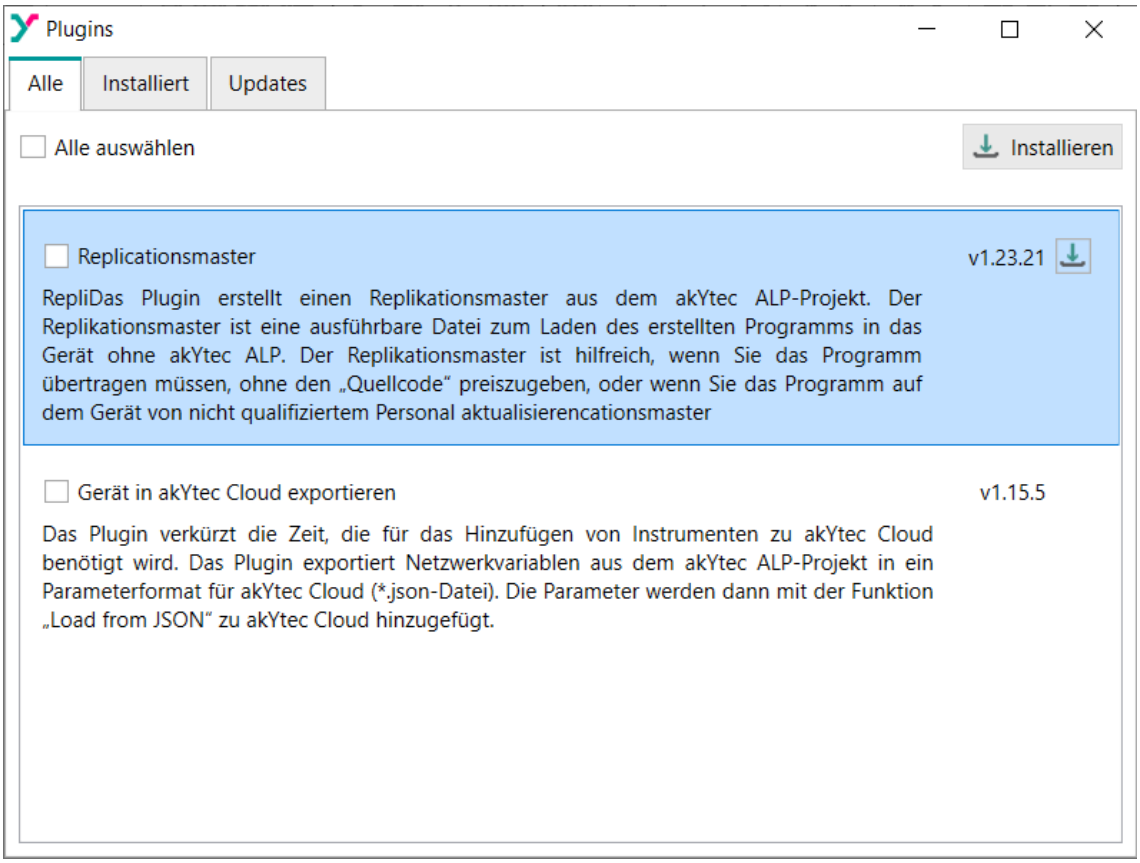
Plugins installieren

So installieren Sie Plugins:

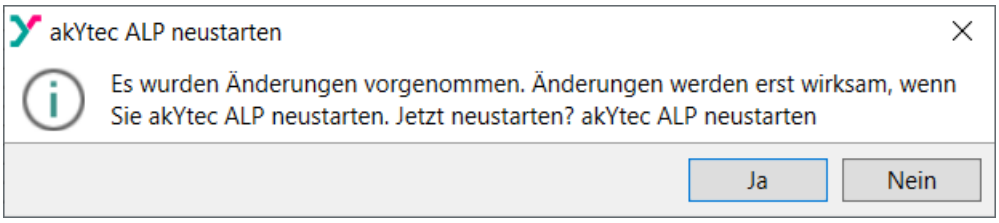
1. Wählen Sie im Hauptmenü **Plugin-Manager** aus.

.

2. Aktivieren Sie im Fenster **Plugins** die erforderlichen Erweiterungen.



3. Klicken Sie auf die Schaltfläche **Installieren**.
4. Nach einer Weile wird der Nachweis der erfolgreichen Installation in Form eines grünen Häkchens neben den ausgewählten Erweiterungen angezeigt.
5. Nachdem das Fenster **Plugins** geschlossen wurde, wird das Dialogfeld zum Neustarten der ALP angezeigt.



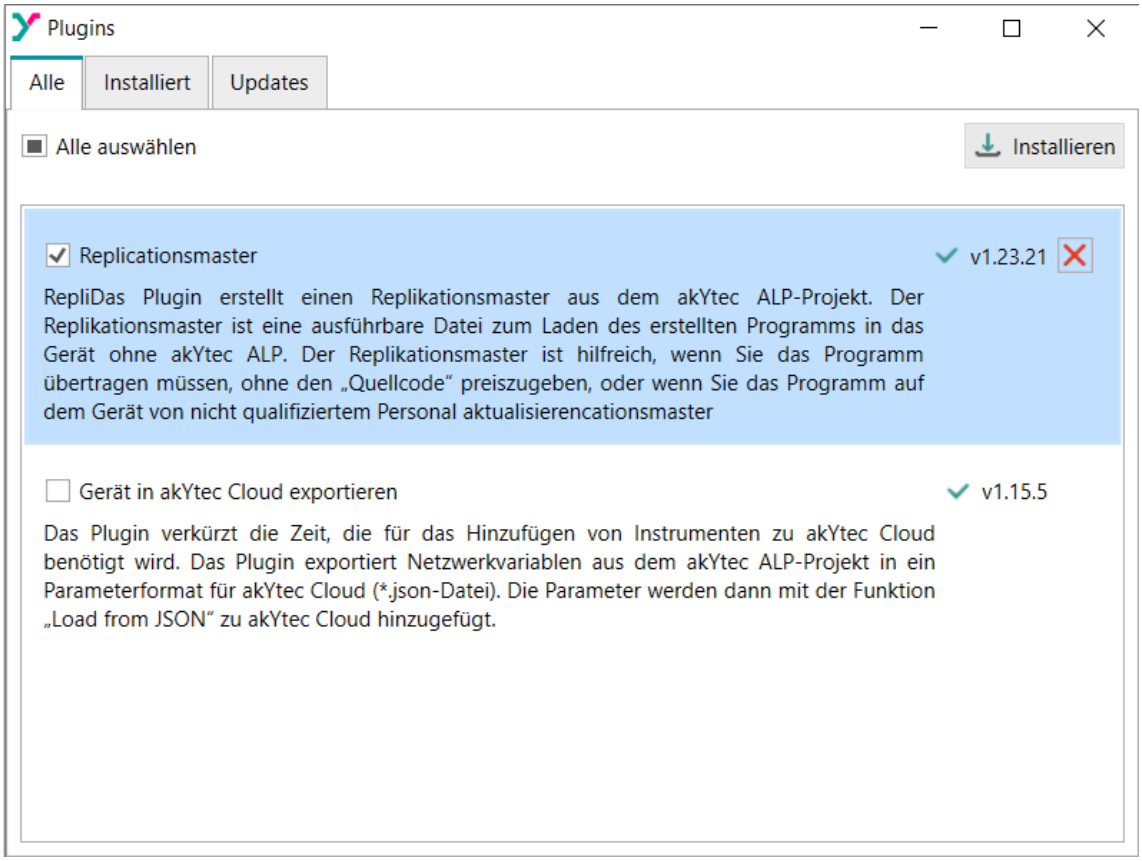
Nach dem Neustart der ALP sind die Plugins einsatzbereit.

HINWEIS
Wenn keine Plugins installiert sind, sollten Sie überprüfen, ob Port 8084 geöffnet ist.

Plugins löschen

So löschen Sie ein Plugin:

1. Wählen Sie im Hauptmenü **Plugin-Manager** aus.
2. Klicken Sie im Fenster **Plugin** neben dem gewünschten Plugin auf die Schaltfläche ✕.



3. Das Löschsymbol ändert sich in das Installationssymbol.
4. Nach dem Schließen des **Plugins**-Fensters wird das Dialogfeld zum Neustart von ALP angezeigt. Nach dem Neustart von ALP sind die gelöschten Erweiterungen nicht mehr im Hauptmenü verfügbar.

9.1 Replikationsmaster

Mit dem Replikationsmaster können Sie aus einem ALP-Projekt eine Programmdatei mit der folgenden Erweiterung erstellen:
**.exe* – Windows
**.dll* – Linux
Mit der Datei kann das Programm ohne Verwendung der ALP in das Gerät geladen werden.

VORSICHT
Das Programm kann nur zwischen identischen Gerätemodifikationen repliziert werden.

Das Ergebnis des Schreibens eines Programms auf das Gerät mithilfe des Replikationsmasters hängt davon ab, ob der Schlüssel im Gerät und im Replikationsmaster verfügbar ist. Der Schlüssel kann mithilfe von ALP auf das Gerät übertragen werden.

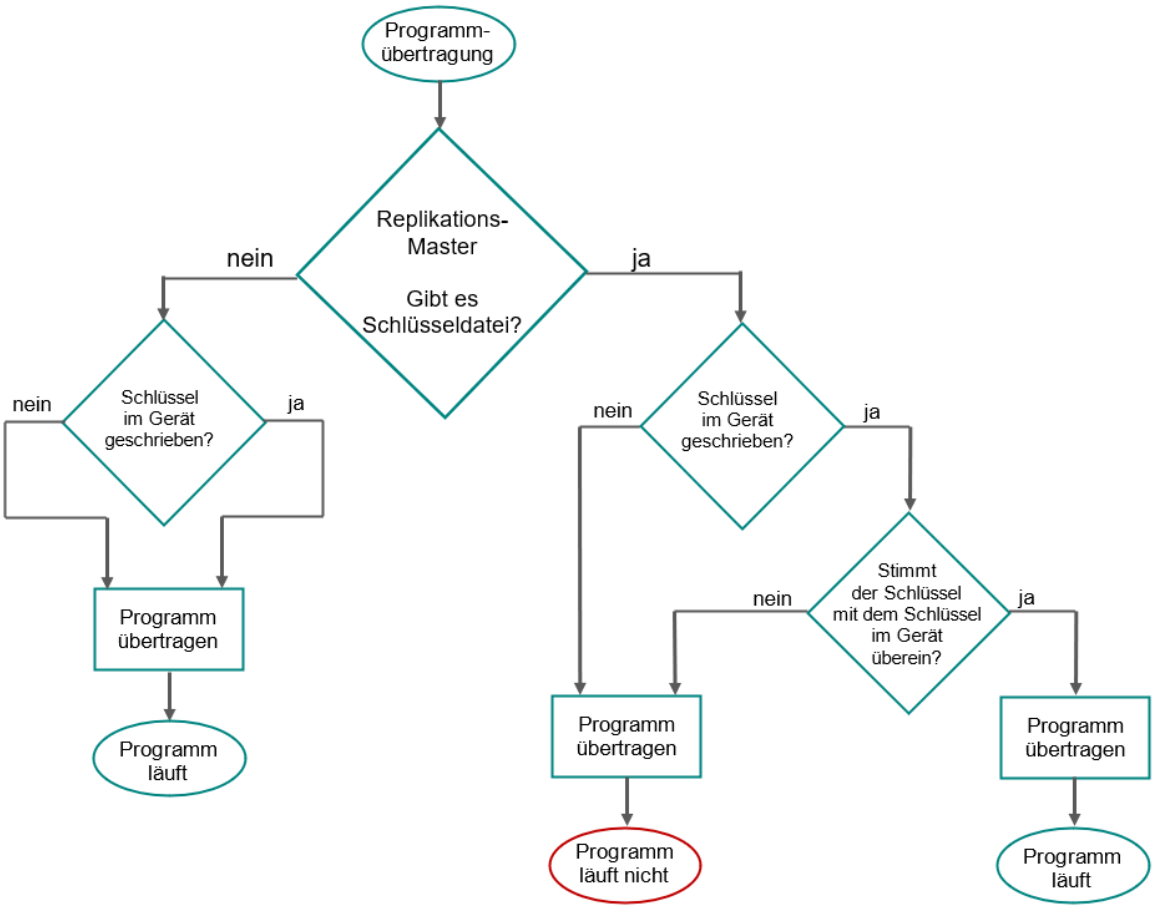


Abb. 9.1 Programmübertragung mit dem Replikationsmaster

Replikationsmaster erstellen

Vor dem Erstellen des Replikationsmasters für Windows empfiehlt es sich, eine *.rtf-Textdatei mit der Projektbeschreibung zu erstellen. Die Projektbeschreibung wird beim Start des Replikationsmasters angezeigt, bevor das Programm auf das Gerät geladen wird.
Um einen Replikationsmaster für Linux zu erstellen, müssen Sie das .NET 6 SDK installieren. Dies kann bei der Installation von ALP erfolgen. Aktivieren Sie dazu im Installationsfenster das Kontrollkästchen **.NET6 SDK installieren**:

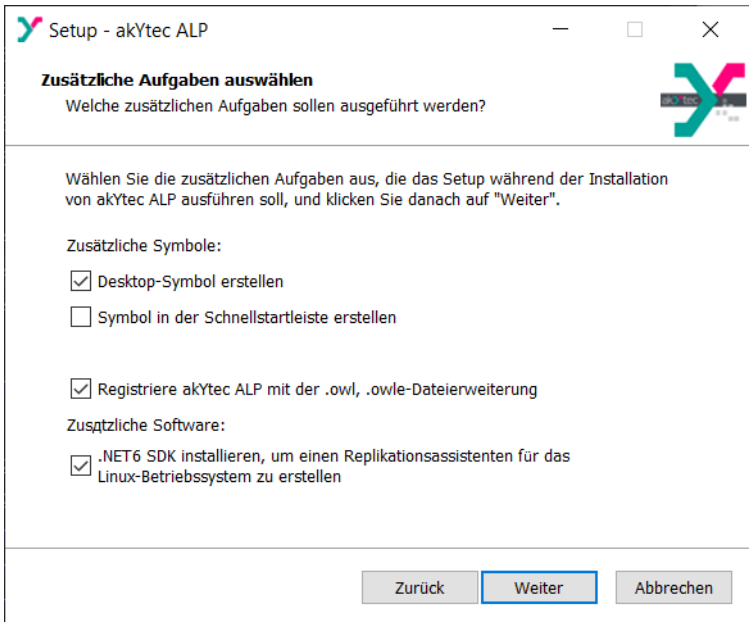


Abb. 9.2 ALP-Installationsfenster

Wenn das .NET 6 SDK nicht installiert ist, tritt beim Erstellen des Replikationsmasters für Linux ein Fehler auf und die Datei wird nicht erstellt. Unten im Fenster sehen Sie einen Hinweis auf die erforderliche Installation und einen Link zum Download des Pakets:

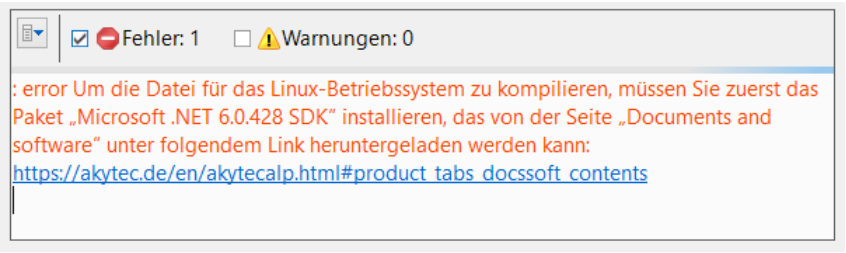
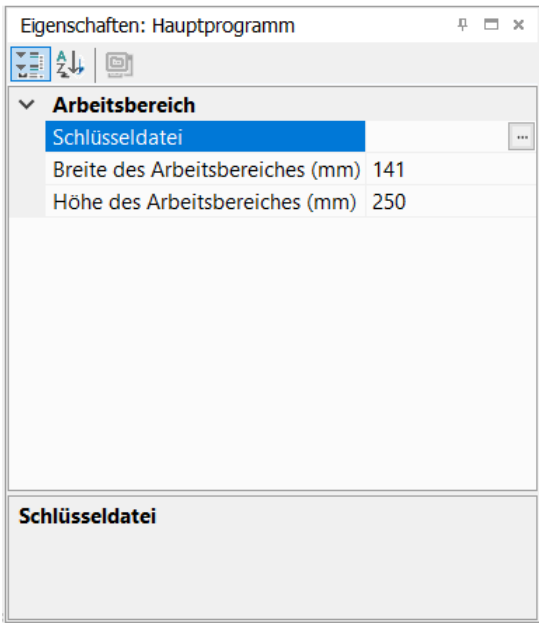


Abb. 9.3 Fehler beim Erstellen der Datei

So erstellen Sie einen Replikationsmaster:

1. Geben Sie im Feld **Schlüsseldateiname** im Projekteigenschaftenfenster die auf dem PC gespeicherte *Schlüsseldatei* an. Um das Programm ohne Schlüssel herunterzuladen, lassen Sie dieses Feld leer.



Wenn der Schlüssel nach der Angabe des Pfads zur Schlüsseldatei gelöscht wird, wird der Replikationsmaster nicht erstellt und unten im Fenster werden Fehlerinformationen angezeigt:

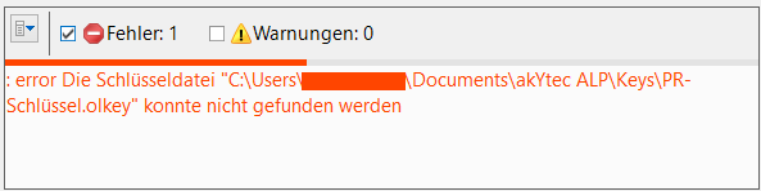


Abb. 9.4 Fehler beim Erstellen der Datei

2. Wählen Sie **Replikationsmaster erstellen** aus dem Hauptmenü **Plugins**.
3. Es öffnet sich ein Fenster zur Erstellung des Replikationsmasters. Wenn ein Schlüssel mit dem Projekt verknüpft ist, wird die Meldung **Benutzerprogramm ist durch einen Schlüssel geschützt!** angezeigt. Andernfalls wird die Meldung **Gerät unterstützt Benutzerprogrammschutz, aber das Projekt wurde nicht verschlüsselt** angezeigt.

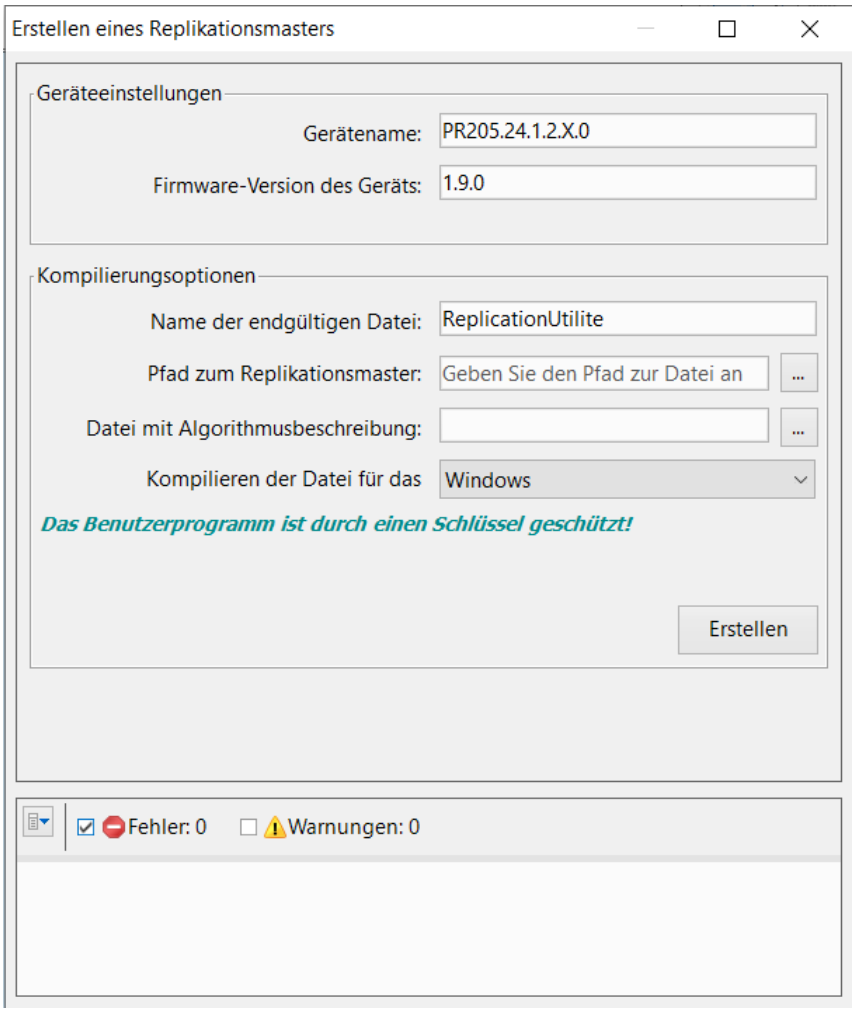


Abb. 9.5 Erstellen des Replikationsmasters

Die **Geräteparameter** werden vom Gerät gelesen und automatisch eingetragen.

Füllen Sie im Fenster „Kompilierungsparameter“ die Felder aus:

- *Name der endgültiger Datei* – Legt den Namen der zu erstellenden Replikationsmasterdatei fest.
- *Pfad zum Replikationsmaster* – Legt den Speicherort der zu erstellenden Datei fest.
- *Datei mit Algorithmusbeschreibung (Windows)* – Legt den Pfad zur Textdatei mit der Projektbeschreibung im Format *.rtf fest (optional).
- *Kompilieren der Datei für das* – Legt das Zielbetriebssystem fest.

4. Klicken Sie auf die Schaltfläche **Erstellen**. Wenn keine Fehler auftreten, wird eine Datei mit dem angegebenen Namen und der Erweiterung **.exe* oder **.dll* erstellt.

HINWEIS
Die Erstellung eines Replikationsmasters für Linux kann längere Zeit dauern.

HINWEIS
Die Dateierweiterung **.dll* wird in Windows möglicherweise nicht angezeigt, es sei denn, die Option zur Anzeige von Systemdateien ist aktiviert:

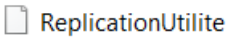


*Abb. 9.6 Anzeige einer Datei mit der Erweiterung ***.dll***

Der Dateierstellungsbericht wird unten im Fenster angezeigt.

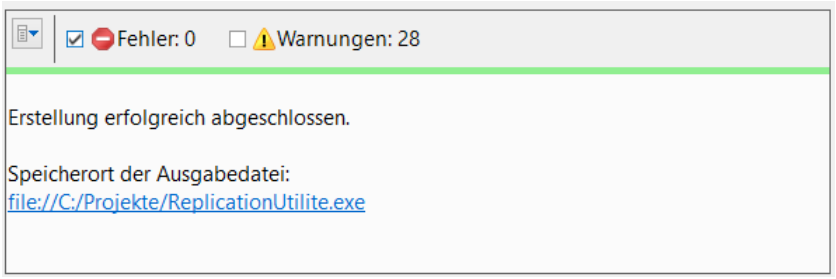


Abb. 9.7 Dateierstellungsbericht

Um den Berichtsanzeigebereich ein- oder auszublenden, klicken Sie auf die Schaltfläche . Informationen zu Warnungen und Fehlern werden angezeigt, wenn die entsprechenden Kontrollkästchen aktiviert sind.

Abb. 9.8 Beispiel für Warnungen und Fehler

9.1.1 Programmuebertragung ueber Replikationsmaster unter Windows

Um mithilfe des Replikationsmasters ein Programm auf das Gerät zu schreiben, sollten Sie:

1. Führen Sie die erstellte Replikationsmasterdatei aus.

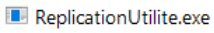


Abb. 9.9 Replikationsmasterdatei

HINWEIS
Beim Öffnen des Masters wird die Projektbeschreibung aus der angehängten Datei mit der Erweiterung **.rtf* angezeigt. War die Datei beim Erstellen des Masters nicht angehängt, ist die Projektbeschreibung leer.

2. Folgen Sie den Anweisungen des Replikationsmasters.

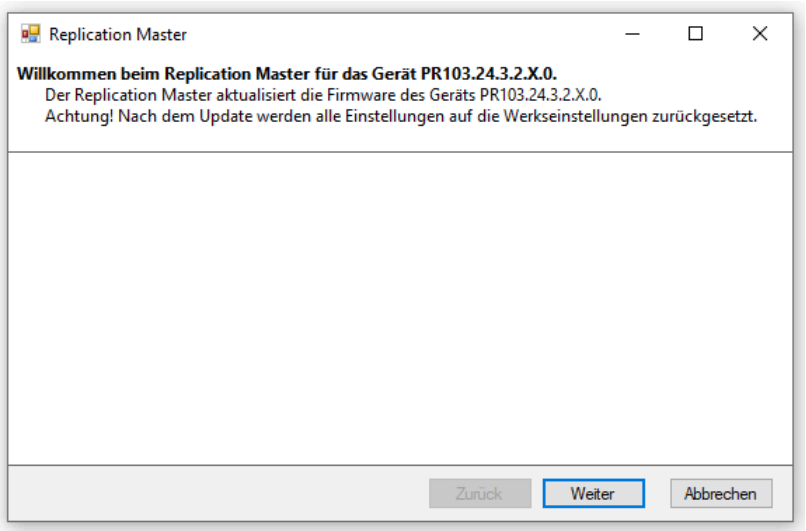


Abb. 9.10 Replikationsmasterfenster

3. Im Abschnitt **Porteinstellungen** wird die Nummer des COM-Ports angezeigt, an den das Gerät angeschlossen ist. Wie Sie die Portnummer herausfinden, erfahren Sie im Abschnitt *Verbindung zum Gerät 4.6*.
4. Wenn das Gerät passwortgeschützt ist, geben Sie das Passwort im Fenster „Replikationsmaster“ ein. Fehlt das Passwort oder wird es falsch eingegeben, ist die Schaltfläche **Weiter** inaktiv. Nach Auswahl des Ports wird der Verbindungsstatus zum Gerät unten rechts angezeigt.

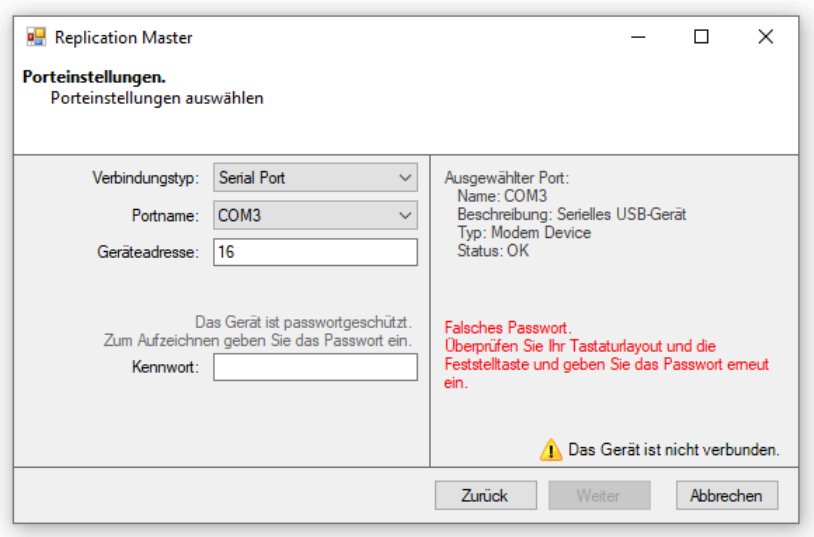
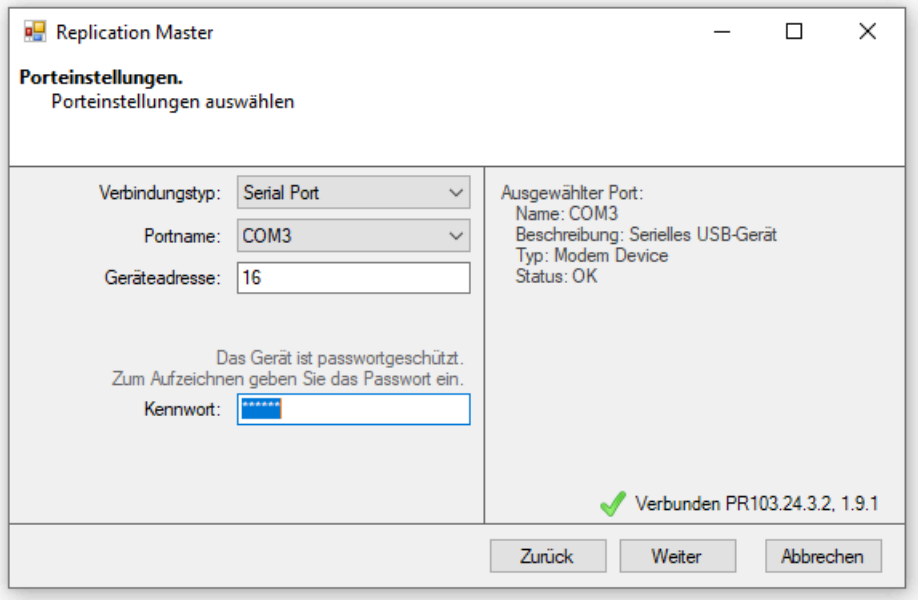


Abb. 9.11 Geräteverbindungsfehler

Bei einem Verbindungsfehler werden das Fehlersymbol ⚠ und ein entsprechender Text angezeigt. Wenn Sie mit der Maus über das Symbol ⚠ fahren, wird ein Tooltip mit einer Fehlerbeschreibung angezeigt.

Fehler	Mögliche Gründe
Das Gerät ist nicht verbunden.	– Falscher COM-Port – Fehlender USB-Treiber – Verbindungsabbruch
Ein anderes Gerät ist erforderlich.	Weist darauf hin, dass das angeschlossene Gerät nicht zum Projekt passt und entweder im Projekt ersetzt oder durch ein geeignetes Gerät ausgetauscht werden muss.

Liegen keine Fehler vor, wird ein grünes Häkchen mit dem Namen des Geräts angezeigt.



Wenn die Verbindung zum Gerät ständig unterbrochen ist (das Häkchen blinkt), belegt möglicherweise eine andere Anwendung den USB-Anschluss. Dies kann passieren, wenn ALP läuft und darin eine Verbindung zum gleichen Anschluss konfiguriert ist. In diesem Fall muss die andere Anwendung geschlossen oder *OFFLINE* geschaltet werden, um das Projekt auf das Gerät zu laden.

5. Wenn die Verbindung stabil ist, sollten Sie auf die Schaltfläche **Weiter** klicken.

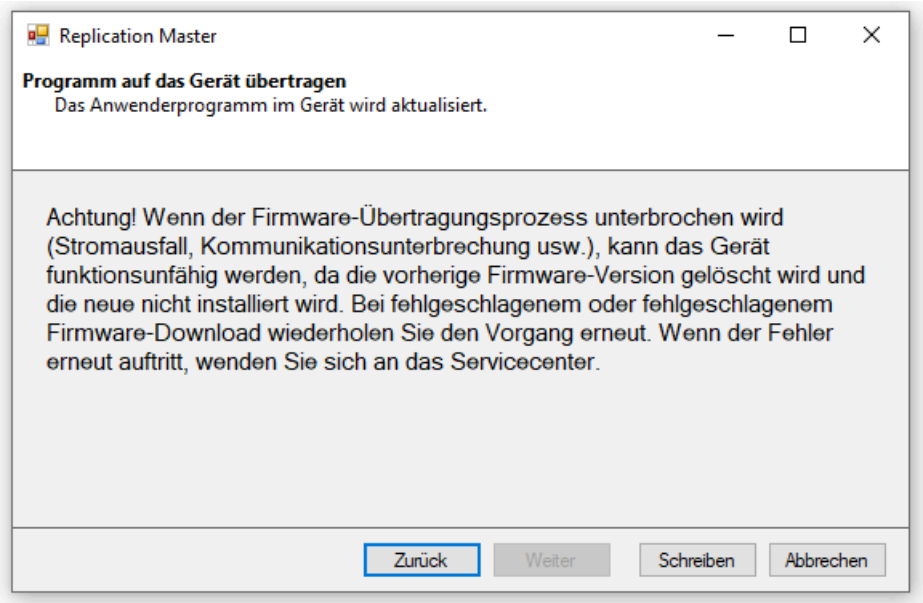


Abb. 9.12 Übertragen des Programms

6. Um das Programm auf dem Gerät aufzunehmen, klicken Sie auf die Schaltfläche **Aufnehmen**. Die Statuszeile der Aufnahme wird unten im Fenster angezeigt. Während des Programmladevorgangs darf die Verbindung zwischen PC und Gerät nicht unterbrochen werden. Nach Abschluss des Ladevorgangs wird die Meldung angezeigt.

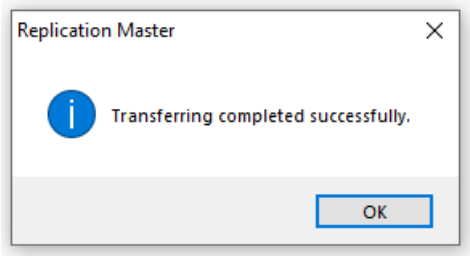


Abb. 9.13 Übertragung abgeschlossen

„Stiller“ Installationsmodus

HINWEIS
Der „stille“ Installationsmodus ist verfügbar, wenn Sie mit dem Replikationsmaster ab Version 1.21.14 arbeiten.

Der Replikationsmaster unterstützt den stillen Installationsmodus für Geräte. Um ihn zu starten, verwenden Sie die folgenden Befehlszeilenargumente:

– ***/silent*** – Start der stillen Installation
– ***/password:*** – Passwort für passwortgeschützte Geräte
– **/ip:** – IP-Adresse des angeschlossenen Geräts bei Ethernet-Verbindung
– **/portname:** – COM-Portnummer bei USB-Verbindung
– **1>> Dateiname.txt** – Textdatei mit Fortschrittsangaben zum Firmware-Vorgang
– **2>> Dateiname.txt** – Textdatei mit Fehlermeldungen während des Firmware-Vorgangs

Ein Beispiel für einen Befehl zum Ausführen der Replikationsmasterdatei ReplicationUtilite.exe, die sich auf dem Desktop befindet, wenn eine Verbindung über eine Ethernet-Schnittstelle besteht:
C:\Benutzer\Benutzername\Desktop\ReplicationUtilite.exe /silent /IP:10.2.3.204
Beispiel für einen Befehl zum Ausführen der Replikationsmasterdatei ReplicationUtilite.exe, die sich auf dem Desktop befindet, wenn eine Verbindung über USB zu einem Gerät mit dem Kennwort *1234* besteht:

C:\Benutzer\Benutzername\Desktop\ReplicationUtilite.exe /silent /portname:COM7 /password:1234
Beispiel für einen Befehl zum Ausführen der Replikationsmasterdatei ReplicationUtilite.exe, die sich bei einer Verbindung über USB auf dem Desktop befindet, um eine Textdatei zu erstellen, die Fehler während des Firmware-Update-Prozesses anzeigt:
C:\Benutzer\Benutzername\Desktop\ReplicationUtilite.exe /silent /portname:COM7 2>> Dateiname.txt

9.1.2 Programmuebertragung ueber Replikationsmaster unter Linux

PC-Anforderungen:

– Ubuntu 22.04 oder 24.04, x86/32- oder x64-Prozessorarchitektur
– .NET 6

Um ein Programm auf ein Gerät unter Linux zu schreiben, öffnen Sie eine Befehlszeile und geben Sie den folgenden Befehl ein:

– **./ReplicationUtilite -ip:192.168.1.100** – Führt die Programmübertragung über eine Ethernet-Verbindung mit Angabe der IP-Adresse des Geräts durch.
– **./ReplicationUtilite -portname:/dev/ttyACM0** – Führt die Programmübertragung über eine serielle Schnittstelle mit Angabe des Portnamens durch.

HINWEIS
Um den Geräteanschluss zu ermitteln, verwenden Sie die folgenden Befehle:
– **lsusb** – für USB
– **dmesg | grep tty** – für den seriellen Anschluss

Ein Beispiel für einen Befehl zum Starten der Replikationsmasterdatei beim Anschluss des Geräts über einen seriellen Port mit der Adresse ACM0:

```
[redacted]:~$ ./ReplicationUtilite -portname:/dev/ttyACM0
```

Abb. 9.14 Befehl zum Schreiben eines Programms

Wenn die Aktionen korrekt ausgeführt werden, zeigt die Befehlszeile die Statuszeile des in das Gerät geladenen Programms an:

```
[#######################################-----------]  77% /
```

Abb. 9.15 Statusleiste

Tabelle 9.1 Mögliche Fehler

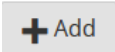
Fehler	Lösung
Sie haben keine Berechtigung zum Ausführen der Datei `Permission denied`	– Wenden Sie sich an den Administrator. – Verwenden Sie den Befehl **sudo ./ReplicationUtilite -portname:/dev/ttyACM0**, wobei ACM0 die Portadresse ist.
Eingabe eines ungültigen Befehls `Incorrect string arguments are specified`	Bitte geben Sie einen gültigen Befehl ein, um die Datei auszuführen.
Das Gerät ist passwortgeschützt `Incorrect string arguments are specified`	Geben Sie den Befehl **-password:** ein.

9.2 Exportieren des Geraets in die akYtec Cloud

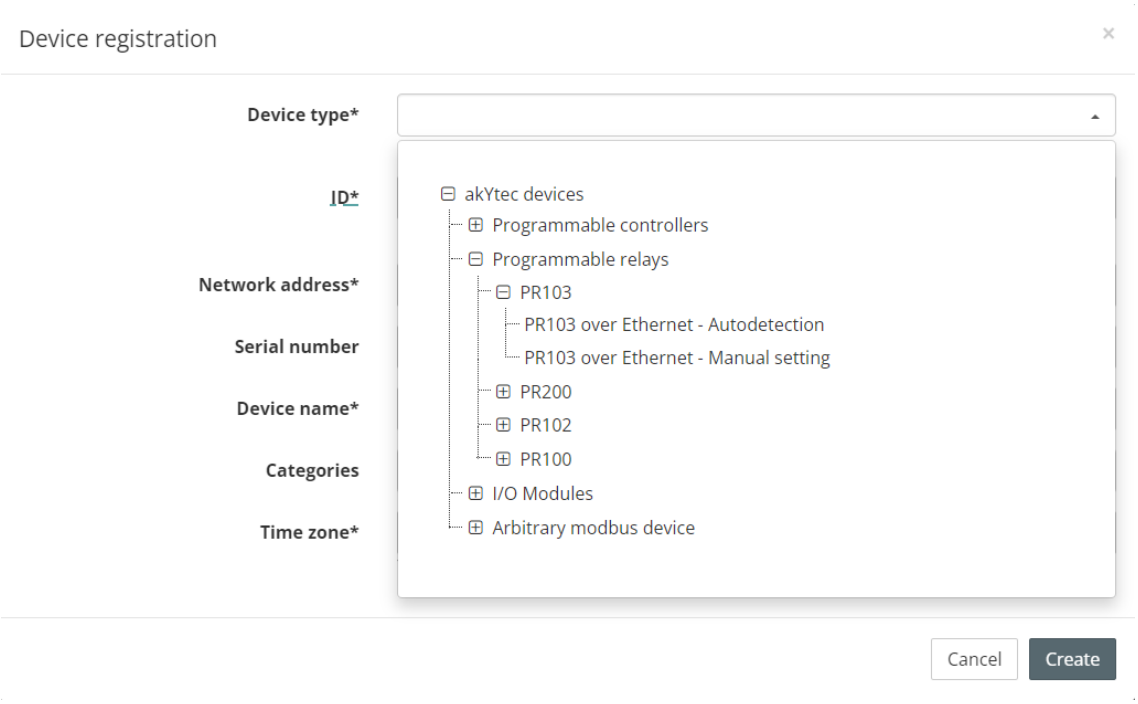
Durch das Exportieren eines Geräts in die akYtec Cloud wird die Zeit verkürzt, die zum Hinzufügen eines im Slave-Modus konfigurierten Geräts zum akYtec Cloud-Dienst benötigt wird.
Für die Arbeit mit der akYtec Cloud ist ein Gateway erforderlich, das mit dem Gerät verbunden werden muss. Vorausgesetzt, das Gateway ist bereits konfiguriert und mit der Zentrale verbunden. Die Einstellungen finden Sie in der *Bedienungsanleitung* des Gateways.
Um die Konfiguration der Modbus-Parameter des Geräts zu exportieren, gehen Sie wie folgt vor:

1. Erstellen Sie ein Projekt mit Einstellungen im Slave-Modus und Netzwerkvariablen.
2. Wählen Sie im Hauptmenü **Plugins** die Option **Gerät in akYtec Cloud exportieren**.
3. Wählen Sie im sich öffnenden Menü den Speicherort und geben Sie den Dateinamen im Format **.json* an.

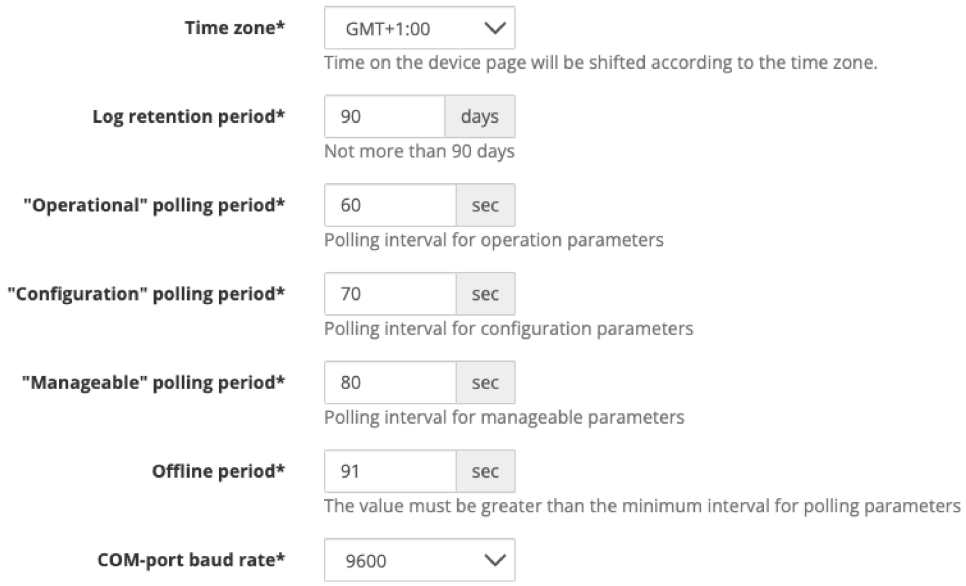
Für die weiteren Schritte ist ein persönliches Konto in der akYtec Cloud erforderlich.
Nach der Anmeldung in Ihrem persönlichen Konto:

1. Klicken Sie auf die Schaltfläche .
2. Klicken Sie auf der sich öffnenden Seite auf die Schaltfläche 

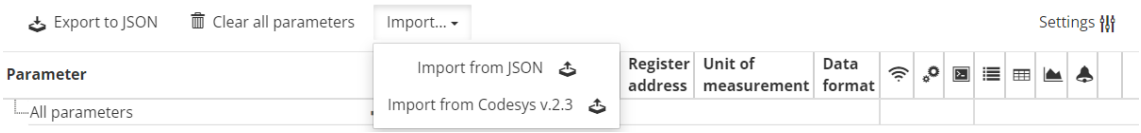
.
3. Geben Sie im angezeigten Menü die **ID** (IMEI/Seriennummer) des Netzwerk-Gateways ein (siehe Gehäuse). Wählen Sie in der Dropdown-Liste **Gerätetyp** in der Liste **Programmierbare Relais (Programmable relays)**das Gerät aus. Geben Sie die in ALP festgelegte Geräteadresse ein. Füllen Sie die restlichen Felder aus. Klicken Sie anschließend auf **Erstellen**.



4. Geben Sie auf der Registerkarte **Allgemein/Allgemeine Einstellungen (General/General Settings)** die Abfragerate und die COM-Port-Einstellungen des Geräts entsprechend den Einstellungen in ALP an. Klicken Sie auf die Schaltfläche **Speichern (Save)**, um die neuen Einstellungen zu übernehmen.



5. Klicken Sie auf der angezeigten Seite auf die Schaltfläche **Parametereinstellungen (Parameter Settings)**. Klicken Sie auf die Dropdown-Liste **Importieren (Import)** und wählen Sie die Option **Aus JSON laden (Load from JSON)**. Wählen Sie im sich öffnenden Menü die zuvor erstellte Datei im Format **.json* aus und klicken Sie auf **Parameter laden (Load Parameters)**.



6. Modbus-Variablen des Geräts werden zur akYtec Cloud hinzugefügt.

10 Tastaturkuerzel

Tastaturkürzel	Aktion
Menü/Datei	
Strg + N	Neues Projekt erstellen
Strg + O	Vorhandenes Projekts öffnen
Strg + Alt + S	Geöffnetes Projekt unter anderem Namen speichern
Strg + S	Geöffnetes Projekt speichern
Strg + P	Drucken
Strg + Umschalt + C	Komponenten-Manager öffnen
Menü/Ansicht	
Strg + Z	Letzte Änderung rückgängig machen
Strg + Y	Abgebrochene Aktion zurückgeben (wiederherstellen)
Menü/Gerät	
Strg + F7	Programm auf das Gerät übertragen
Strg + Umschalt + V	Variablentabelle öffnen
Strg + Umschalt + S	Gerätekonfiguration öffnen
Menü/Service	
Umschalt + F5	Zum Simulationsmodus
F6	Simulation starten
F7	Simulation stoppen
F8	Simulation anhalten
F10	Einzelzyklus
Strg + F5	Zum Debug-Modus
Menü/Plugins	
Strg + Umschalt + P	Plugin-Manager öffnen
Menü/Hilfe	
F1	Hilfe öffnen
Symbolleiste Einfügen	
Strg + Umschalt + F	ST-Funktion erstellen
Strg + Umschalt + M	Makro erstellen
Strg + Umschalt + B	ST-Funktionsbaustein erstellen
Strg + M	Makro aus einer Auswahl erstellen
Verwendungsstellen	
F5	Verwendungsstellen aktualisieren
Tasten zum Arbeiten mit Elementen	
Strg + C	Ein Element kopieren

Tastaturkürzel	Aktion
Strg + V	Einfügen aus der Zwischenablage
Entf	Ausgewähltes Elements löschen
Tasten zur Größenänderung von Elementen	
Strg + →	Breite eines ausgewählten Elements erhöhen
Strg + ←	Breite eines ausgewählten Elements verringern
Strg + ↓	Höhe eines ausgewählten Elements erhöhen
Strg + ↑	Höhe eines ausgewählten Elements verringern
Skalieren des Arbeitsbereichs	
Strg + Mausrad	Wenn Sie das Mausrad von sich weg drehen, vergrößert sich der Maßstab des Arbeitsbereichs. Wenn Sie das Mausrad auf sich zu drehen, verkleinert sich der Maßstab des Arbeitsbereichs
Strg + «+»	Maßstab erhöhen
Strg + «–»	Maßstab verringern
Zwischen Registerkarten wechseln	
Tab + →	Zwischen Registerkarten wechseln
Tab + ←	

11 Programmbeispiele

Die folgenden Beispiele mit einfachen Aufgaben veranschaulichen die Erstellung eines Schaltprogramms in der ALP-Software.

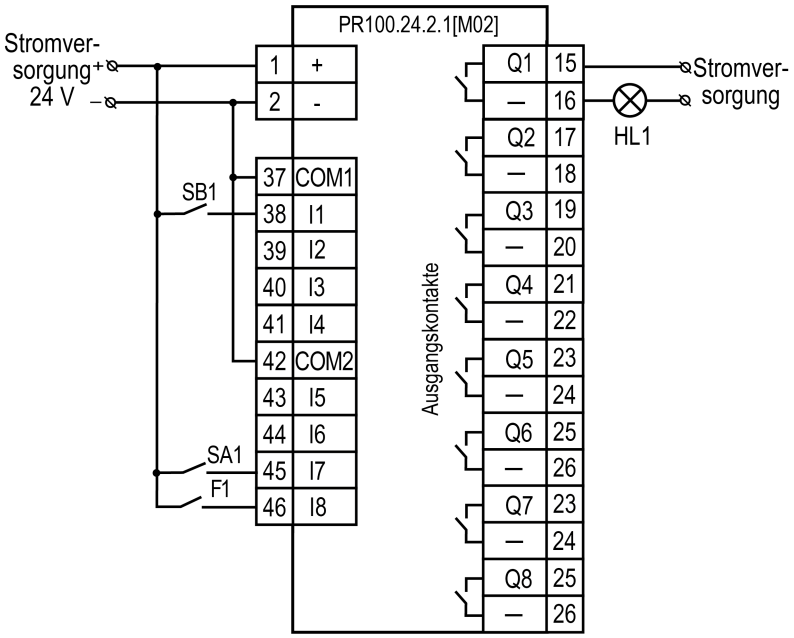
– *Lichtschalter mit Abschaltautomatik*
– *Mixersteuerung*
– *Direkte Verbindung eines PR103 zur akYtec Cloud*

11.1 Aufgabe 1: Lichtschalter mit Abschaltautomatik

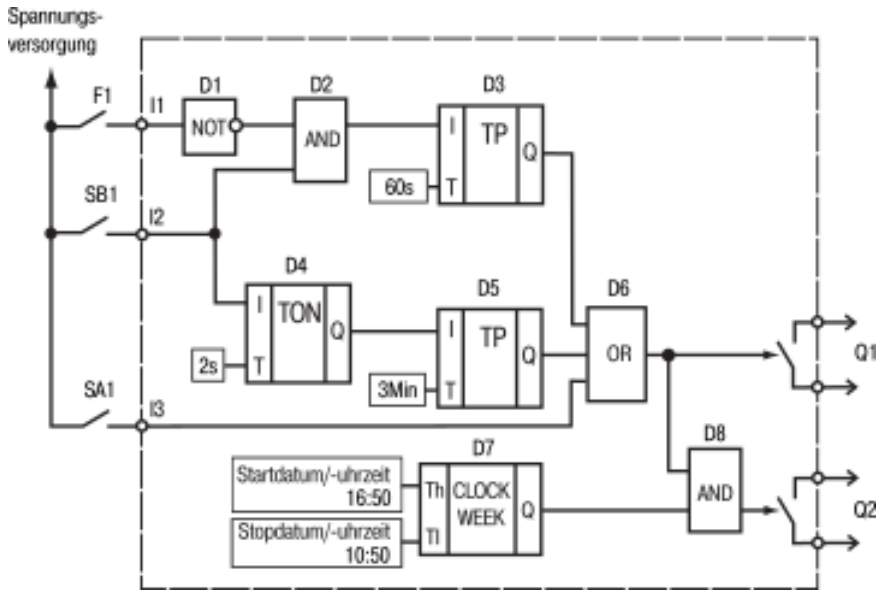
Die Aufgabe besteht darin, das Licht für eine bestimmte Zeit einzuschalten, z. B. beim Hauseingang.
Aufgabenstellung:

1. Der Lichtsensor F1 und der Lichttaster SB1 „ZEIT“ werden vor der Eingangstür montiert.
2. Wird die Taste SB1 kurz gedrückt und reicht das Umgebungslicht nicht aus, soll das Licht für 1 Minute eingeschaltet werden – diese Zeit sollte ausreichen um ein Schlüsselloch zu finden und die Tür zu öffnen.
3. Wird die Taste SB1 für 2 Sekunden gedrückt, soll das Licht unabhängig vom Umgebungslicht für 3 Minuten eingeschaltet werden – dieser Modus kann bei der Eingangsreinigung sinnvoll sein.
4. Bietet die Möglichkeit, das Licht unabhängig vom Umgebungslicht durch Befehle von externen Geräten oder mit dem Schalter SA1 „CONST“ zu steuern. Dieser Modus kann beim Empfang von Gästen oder für die weitere Automatisierung der Wohnung im Rahmen des Programms „Smart House“ nützlich sein.
5. Bieten Sie die Möglichkeit, das Licht nur zu einer bestimmten Zeit einzuschalten.

Geräteauswahl:
Zur Umsetzung dieser Aufgabe muss das Steuergerät mindestens über zwei digitale Eingänge, einen digitalen Ausgang und eine integrierte Echtzeituhr verfügen. Diese Funktionen können Geräte der PR100-Serie bieten.
Die Aufgabenrealisierung mit dem Gerät PR100.24.2.1(M02):



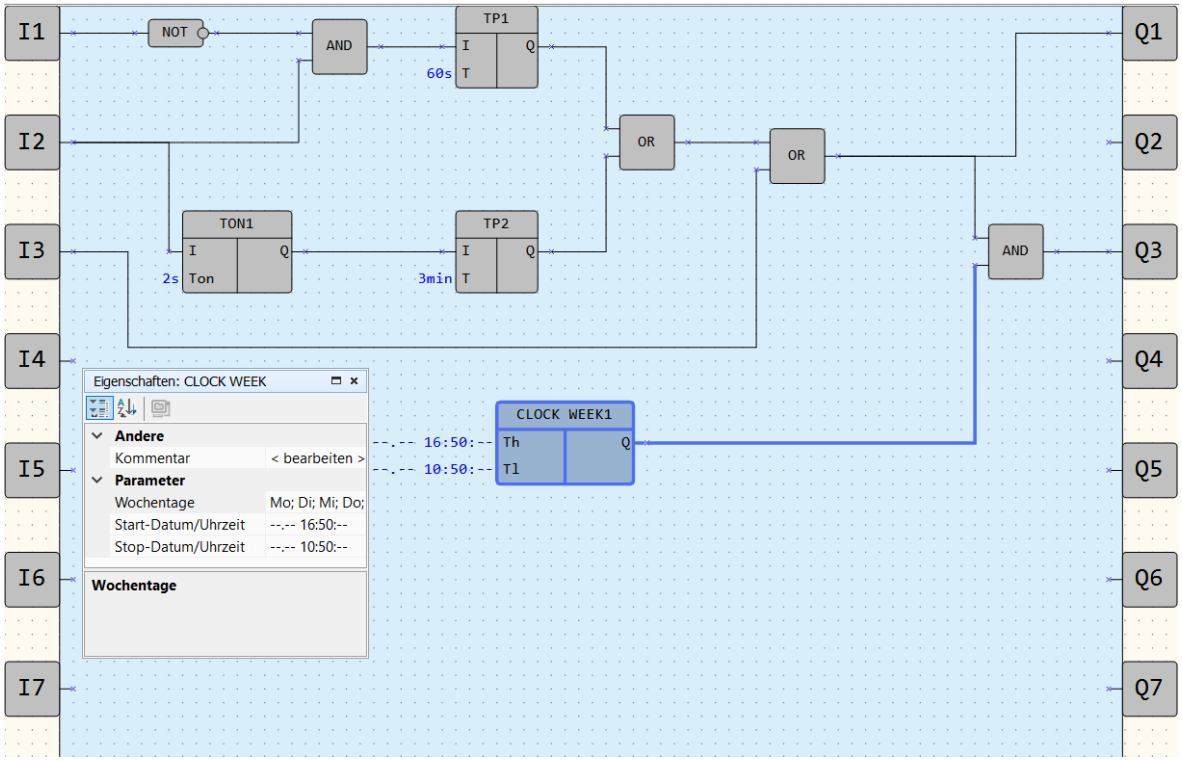
Schaltungsprogramm
Das Schaltprogramm kann wie in der folgenden Abbildung gezeigt implementiert werden.



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 Umsetzung der Aufgabenpunkte 1-4
Ausgang Q2 – Ausgang zur Umsetzung des Aufgabenpunkts 5
Programmbeschreibung:

1. Wird der Taster SB1 kurz (< 2 s) gedrückt, wird das logische UND (D2) aktiviert. Reicht das Umgebungslicht nicht aus, ist auch der erste Eingang von D2 **True** und der Timer TP „Pulse“ (D3) bildet einen Impuls mit 1 Minute Dauer. Dieser Impuls aktiviert über das logische ODER (D6) den Ausgang Q1 und das Licht wird für 1 Minute eingeschaltet.
2. Wird die Taste SB1 > 2 s gedrückt, aktiviert der Einschaltverzögerungstimer TON (D4) den Timer TP „Puls“ (D5), ein Impuls mit der Dauer von 3 Minuten aktiviert über logisches ODER (D6) den Ausgang Q1 und das Licht wird für 3 Minuten eingeschaltet.
3. Bei ausreichender Umgebungshelligkeit wird der Kontakt des Sensors F1 geschlossen, das logische UND (D2) deaktiviert und der Zeitgeber TP „Puls“ (D3) blockiert.
4. Wird der Schalter SA1 „CONST“ geschlossen, wird über das logische ODER (D6) der Ausgang Q1 aktiviert und das Licht dauerhaft eingeschaltet.
5. Wenn Sie das Licht nur an bestimmten Wochentagen zu bestimmten Uhrzeiten nutzen möchten, können Sie den Ausgang Q2 verwenden. Mit der Wochenzeitschaltuhr CLOCKW (D7) können Sie Start- und Stoppzeit sowie die Wochentage für die Beleuchtung einstellen.

Das in ALP erstellte Schaltprogramm ist in der folgenden Abbildung dargestellt.

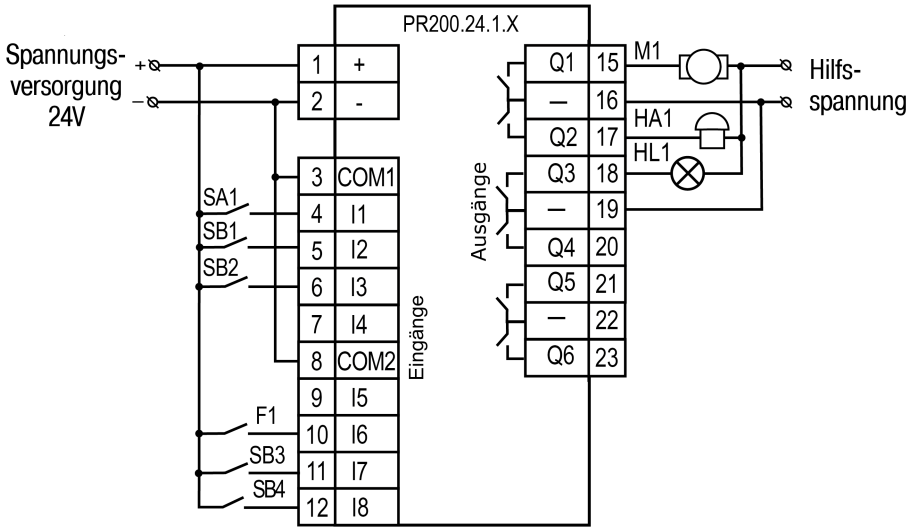


11.2 Aufgabe 2: Mixersteuerung

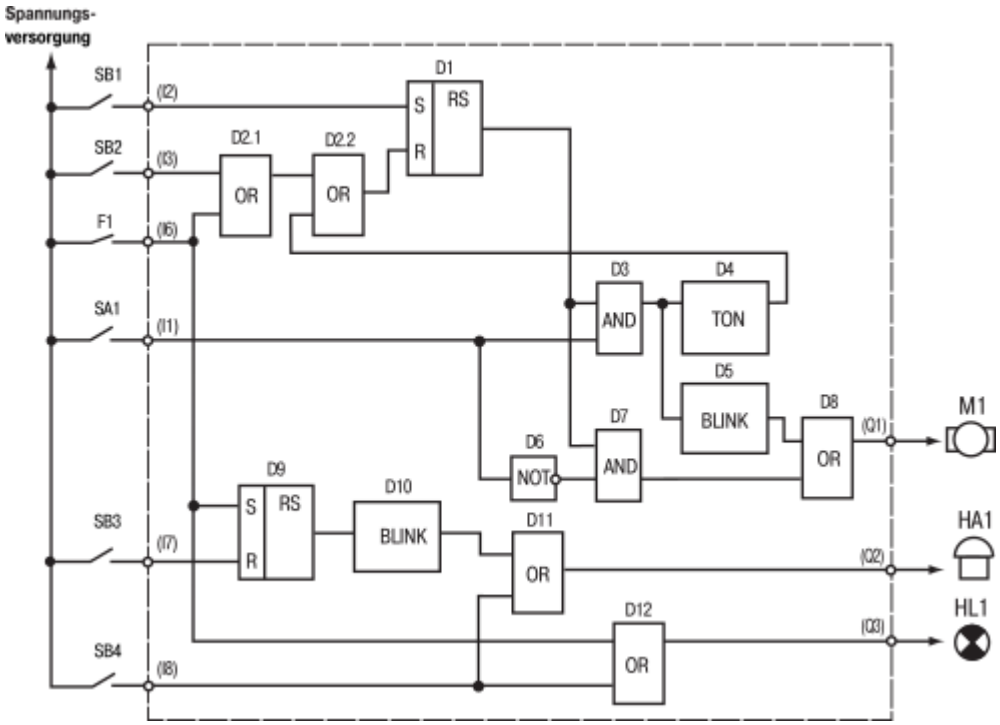
Die Aufgabe besteht darin, einen Industriemixer mit einfachen Steuerungsfunktionen zu realisieren.
Aufgabenstellung:

1. Es müssen die Betriebsarten Automatik und Manuell gewählt werden. Zum Umschalten zwischen den Betriebsarten dient der Schalter SA1 „MODE“.
2. Im Automatikbetrieb kann der Betriebszyklus mit der Taste SB1 „START“ gestartet und mit Zyklusende automatisch oder manuell mit der Taste SB2 „STOP“ gestoppt werden. Die Zyklusdauer beträgt 5 Minuten. Während des Zyklus ist der Motor des Mixers abwechselnd 15 Sekunden lang an und 10 Sekunden lang aus. Alle Einstellungen können im Programm geändert werden.
3. Im Handbetrieb kann der Motor mit der Taste SB1 „START“ gestartet und mit der Taste SB2 „STOP“ gestoppt werden.
4. Bei Überlastung des Motors (Überlastschalter F1) soll dieser automatisch abgeschaltet werden, ein intermittierendes akustisches Warnsignal (HA1) im 3-Sekunden-Takt erklingen und eine Betriebsstörung durch die Signalleuchte HL1 „Überlast“ angezeigt werden.
5. Das akustische Signal kann mit der Taste SB3 „RESET“ abgeschaltet werden.
6. Die Taste SB4 „CONTROL“ dient der Funktionsprüfung der Lampe HL1 und des akustischen Signals HA1.

Geräteauswahl:
Zur Umsetzung dieser Aufgabe muss das Steuergerät über mindestens 6 digitale Eingänge und 3 digitale Ausgänge verfügen. Diese Funktionen können Geräte der PR200-Serie bieten.
Die Aufgabenrealisierung mit dem Gerät PR200.230.1.X:



Schaltprogramm
Das Schaltprogramm kann wie in der folgenden Abbildung gezeigt implementiert werden.



Eingang I1 – Schalter SA1 „MODE“
Eingang I2 – Taster SB1 „START“
Eingang I3 – Taster SB2 „STOP“
Eingang I4 – Überlastschalter F1
Eingang I5 – Taster SB3 „RESET“
Eingang I6 – Taster SB4 „TEST“
Eingang I8 – verbunden mit dem Taster SB4 „TEST“
Ausgang Q1 – Motor
Ausgang Q2 – akustisches Signal HA1
Ausgang Q3 – Signallampe HL1
Programmbeschreibung:

1. Eingang I2 (SB1 „START“)

   Wird der Taster SB1 gedrückt, wird der RS-Trigger D1 **True**, solange kein Reset-Signal am Eingang R anliegt. Der weitere Signalverlauf hängt vom Zustand des Schalters SA1 „MODE“ ab:

– Wenn SA1 geöffnet ist (Handbetrieb), werden das logische UND (D7) und das logische ODER (D8) aktiviert und der Motor M1 (Ausgang Q1) eingeschaltet.
– Wenn SA1 geschlossen ist (Automatikbetrieb), ist das logische UND (D7) deaktiviert und das Startsignal kann nur den Impulsgeber BLINK (D5) aktivieren, um den Betriebszyklus zu starten (15 s an / 10 s aus) und den Einschaltverzögerungstimer TON (D4), um ihn zu stoppen (in 5 Min.).

2. Eingang I3 (SB2 „STOP“)

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

3. Eingang I1 (SA1 “MODE”)
   – Wenn der Schalter SA1 geöffnet ist (Handbetrieb), ist das logische UND D3 deaktiviert und D7 aktiviert, der Zeitgeber D4 und der Impulsgeber D5 sind deaktiviert und der Motor M1 kann nur mit SB1 gestartet und mit SB2 gestoppt werden.
   – Wenn der Schalter SA1 geschlossen ist (Automatikbetrieb), wird das logische UND D3 aktiviert und D7 deaktiviert, somit kann der Motor M1 nur durch den Impulsgeber D5 gestartet werden (15 s an / 10 s aus Zyklus) und durch den Zeitgeber D4 in 5 Minuten gestoppt werden.

4. Eingang I6 (Überlastschalter F1)

   Bei Überlastung des Motors wird der Kontakt F1 geschlossen, der RS-Trigger D1 zurückgesetzt und der Motor gestoppt.

   Gleichzeitig wird über das logische ODER (D12) die Signallampe HL1 eingeschaltet und über den RS-Trigger D9 das akustische Signal HA1 aktiviert. Der Impulsgeber D10 gibt ein intermittierendes akustisches Signal mit dem Takt 3 s ein / 3 s aus ab.

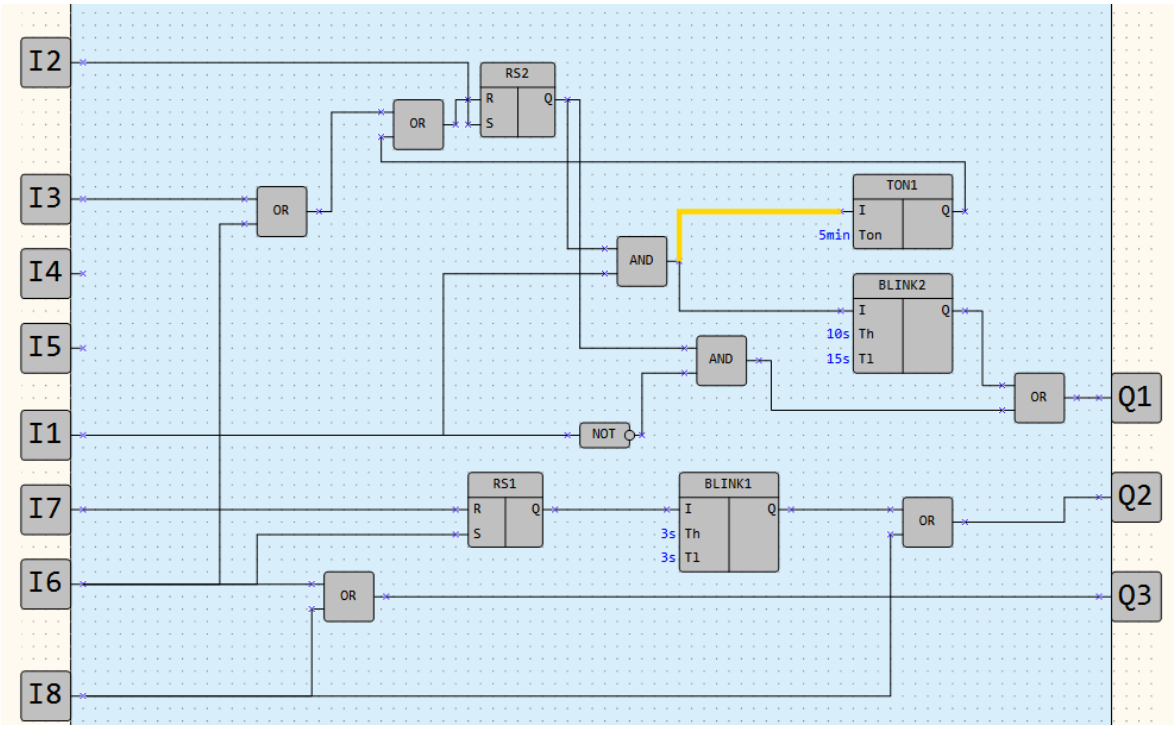
5. Eingang I7 (SB3 “RESET”)

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

6. Eingang I8 (SB4 “TEST”)

   Mit der Taste TEST können das akustische Signal HA1 und die Signallampe HL1 getestet werden. Wird die Taste SB4 gedrückt, werden die logischen ODER-Verknüpfungen D11 und D12 aktiviert, die Ausgänge Q2 und Q3 aktiviert, das akustische Signal und die Lampe eingeschaltet.

Das Schaltprogramm ist in der folgenden Abbildung dargestellt.



HINWEIS

1. *Die verbleibenden zwei ungenutzten Eingänge und ein Ausgang können zur Realisierung von Zusatzfunktionen genutzt werden. Beispielsweise um zwischen verschiedenen Zeiteinstellungen für den automatischen Motorbetrieb umzuschalten oder um andere Betriebsparameter des Mixers umzuschalten.*
2. *Der technologische Betriebszyklus kann durch Implementierung eines Inkrementalzählers (CT) zum Abschalten des RS-Triggers D1 vollständig automatisiert werden.*

11.3 Aufgabe 3: Direkte Verbindung eines PR103 zur akYtec Cloud

Direkte Verbindung zur akYtec Cloud ist für Geräte der zweiten Generation verfügbar.

HINWEIS
Eine Liste der Geräteder zweiten Generation finden Sie im Abschnitt *Einführung*.

Das Gerät muss für den Betrieb konfiguriert und mit dem Netzwerk verbunden sein.
Um eine Verbindung zur akYtec Cloud herzustellen, sollten Sie:

1. Verbinden Sie das Gerät mit dem PC und erstellen Sie ein Projekt mit Netzwerkvariablen.
2. Legen Sie ein *Passwort* fest, um auf das Gerät zuzugreifen.
3. Erlauben Sie im *Geräteeinstellungenfenster* den Zugriff auf die akYtec Cloud.

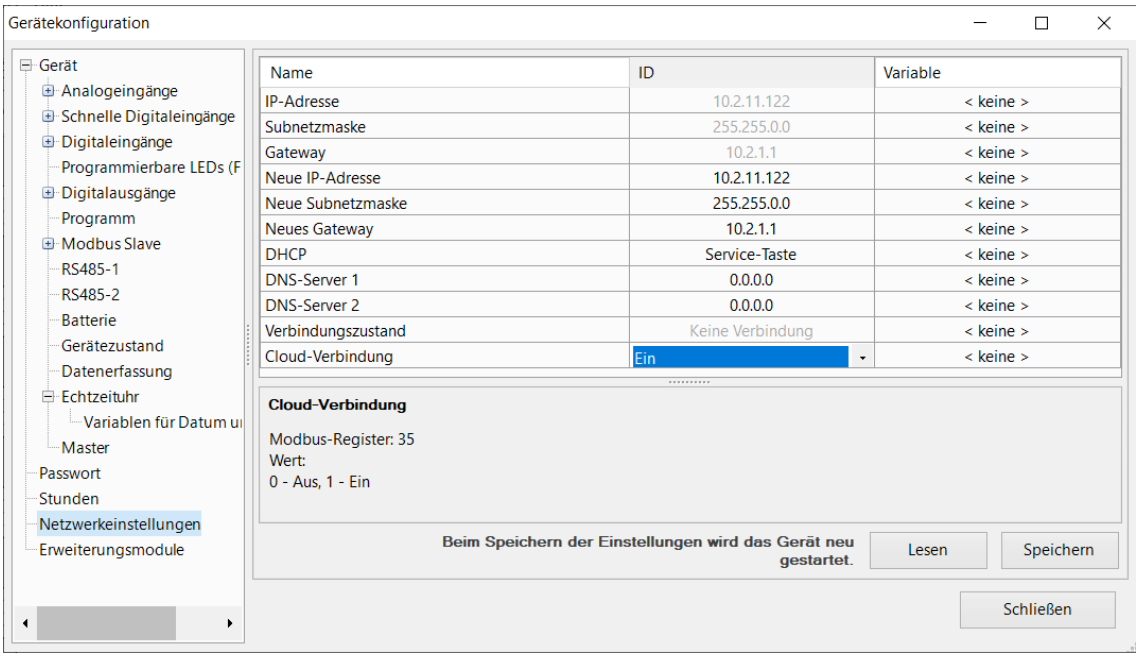


Abb. 11.1 Gerätekonfigurationsfenster

HINWEIS
Der Wert des Statusparameters **Cloud-Verbindung** kann über Modbus ausgelesen oder mit einer Benutzerprogrammvariable verknüpft werden. Die Parameterbeschreibung finden Sie in der folgenden Tabelle..

Tabelle 11.1 Mögliche Zustände des Cloud-Verbindungsparameters

Zustand	Wert	Beschreibung
keine Verbindung	0	Cloud-Freigabe ist deaktiviert.
Identifikation	1	Das Gerät verbindet sich mit der Cloud.
Normalbetrieb	2	Die Cloud steuert das Gerät. Es liegen keine Fehler vor.
Netzwerkfehler	3	Das Gerät kann keine Verbindung zum Cloud-Server herstellen.
kein Passwort	4	Für das Gerät ist kein Passwort festgelegt.

4. Erlauben Sie den Fernzugriff auf Modbus-Register.

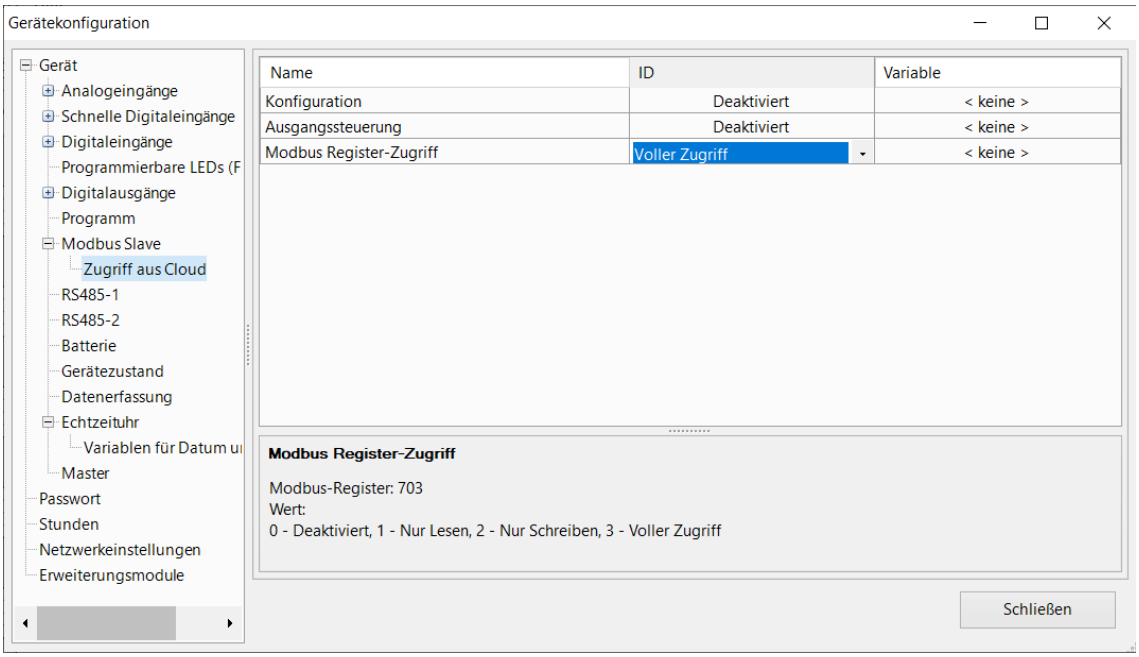
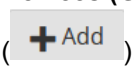


Abb. 11.2 Gerätekonfigurationsfenster

5. Laden Sie das Programm in das Gerät (**Gerät → Programm auf Gerät übertragen**).
6. Rufen Sie die Hauptseite des *Cloud*-Dienstes auf. Falls keine Registrierung vorhanden ist, müssen Sie den Registrierungsprozess durchlaufen.
7. Gehen Sie zur Registerkarte **Administration (Verwaltung)**, öffnen Sie die Registerkarte **Devices (Geräte)** und klicken Sie auf die Schaltfläche **Add device (Gerät hinzufügen)** (

).
8. Es erscheint ein Fenster zur Auswahl des Gerätetyps.

HINWEIS
Die direkte Verbindung zur Cloud wird am Beispiel des PR103 erläutert. Um weitere Geräte auf der neuen Plattform zu verbinden, folgen Sie ähnlichen Schritten.

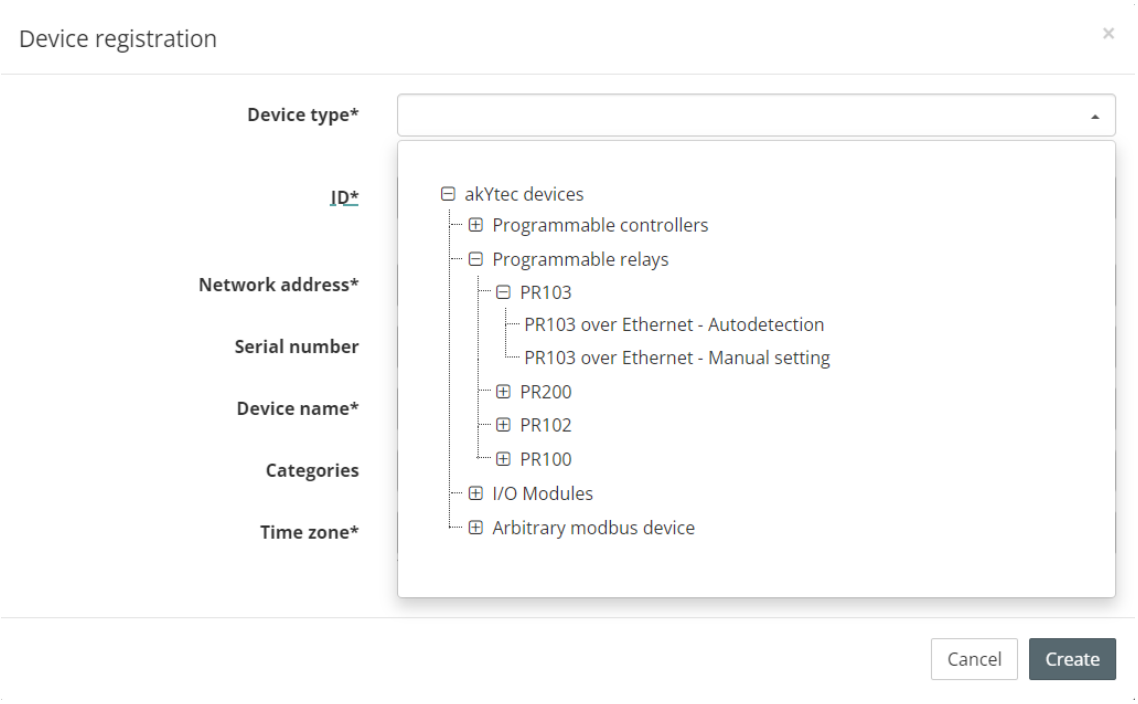


Abb. 11.3 Auswählen des Gerätetyps

Option **PR103 über Ethernet – Automatische Erkennung**:

HINWEIS
Bei dieser Methode zum Hinzufügen eines Geräts werden nicht nur Netzwerkvariablen angezeigt, sondern auch der gesamte Baum der Geräteparameter.

a. Im angezeigten Fenster in den Feldern:
   – ID (ID): Geben Sie die Seriennummer des Geräts ein.
   – Network address (Netzwerkadresse): Lassen Sie die Adresse 1 unverändert.
   – Device name (Gerätename): Geben Sie den Namen des Geräts ein.
   – Time zone (Zeitzone): Geben Sie die Zeitzone an, in der sich das Gerät befindet.

b. Klicken Sie auf die Schaltfläche „Add“ („Hinzufügen“). Die Oberfläche mit den Grundeinstellungen des Geräts wird geöffnet.

c. Geben Sie das Passwort für das Gerät ein. Bei Bedarf können Sie weitere Einstellungen (z. B. den Abfragezeitraum) ändern.

d. Klicken Sie auf die Schaltfläche „Save“ („Speichern“), um die neuen Einstellungen anzuwenden.

e. Die Cloud stellt eine Verbindung zum Gerät her und liest alle Parameter daraus.

Option **PR103 über Ethernet – Manuelle Einrichtung**:

a. Im angezeigten Fenster in den Feldern:
   – ID (ID): Geben Sie die Seriennummer des Geräts ein.
   – Network address (Netzwerkadresse): Lassen Sie die Adresse 1 unverändert.
   – Device name (Gerätename): Geben Sie den Namen des Geräts ein.
   – Time zone (Zeitzone): Geben Sie die Zeitzone an, in der sich das Gerät befindet.

b. Klicken Sie auf „Add“ („Hinzufügen“). Die allgemeine Geräteeinstellungsoberfläche wird geöffnet.

c. Geben Sie das Passwort für das Gerät ein. Bei Bedarf können Sie weitere Einstellungen ändern. Klicken Sie auf die Schaltfläche **Erstellen (Create)**, um die neuen Einstellungen zu übernehmen.

d. Fügen Sie anschließend auf der Parameter/Parameter settings (Registerkarte Parameter/ Parametereinstellungen) die Netzwerkparameter des Geräts mit einer der verfügbaren Methoden hinzu:

– Manuell, entsprechend den in PR103 angegebenen Netzwerkvariablen. Es ist zu beachten, dass die Registeradressen hexadezimal angegeben sind – daher unterscheiden sich die verwendeten Werte von den in ALP angegebenen – 16384 (DEC) = 4000 (HEX).
– Als *.json-Datei, wenn Sie die Erweiterung „Device Export“ in akYtec Cloud ALP verwenden. Um Parameter hinzuzufügen, klicken Sie auf die Dropdown-Liste „Import“ und wählen Sie die Option „Loading from JSON“. Wählen Sie im sich öffnenden Menü die zuvor erstellte *.json-Datei aus und klicken Sie auf die Schaltfläche „Download parameters“.

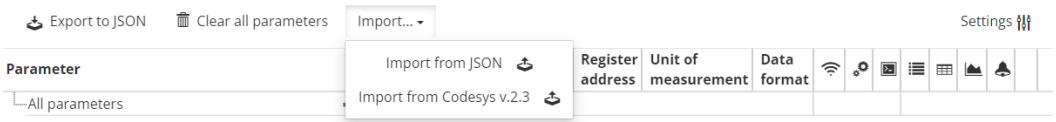


Abb. 11.4 Parameter importieren

Wenn alle Einstellungen korrekt waren, werden auf der Registerkarte **Allgemeine Daten** die Gerätedaten angezeigt. Wird das Gerät nicht von der Cloud abgefragt, sollten Sie die Netzwerkeinstellungen und den Verbindungsstatus zur Cloud überprüfen.

Abb. 11.5 Beispielanzeige einer Verbindung zur Cloud

Wenn die Werte nicht aus der Cloud auf das Gerät geschrieben werden, sollten Sie überprüfen, ob das Kontrollkästchen in der Spalte **Managed parameter (Verwaltete Parameter)** aktiviert ist:

Abb. 11.6 Verwaltete Parameter

12 ST-Sprache

ST (Structured Text) ist eine höhere Textsprache. Sie ist eine der fünf Sprachen, die vom IEC-Standard 61131-3 unterstützt werden.
Mit ALP können Sie Funktionen und Funktionsbausteine in der Sprache ST erstellen.

- *Syntax*
- *Datentypen*
- *Sprachstruktur*

12.1 Syntax

Schlüsselwörter können in Groß- und Kleinbuchstaben eingegeben werden. Leerzeichen und Tabulatoren haben keinen Einfluss auf die Syntax und können überall verwendet werden.
Für die Namen von Variablen, Funktionen und Funktionsblöcken gelten folgende Regeln:

- Der Name darf keine Leerzeichen oder Sonderzeichen enthalten (z. B. !, @ usw.). Die Ausnahme ist der Unterstrich (_).
- Der Name muss mit einem Buchstaben beginnen.
- Der Variablenname darf nur Buchstaben des lateinischen Alphabets enthalten.
- Der Name darf nicht mehrere Unterstriche (_) hintereinander enthalten (d. h. der Name **i__Test** ist ungültig, aber der Name **i_Te_st** ist gültig).
- Bei Objektnamen wird die Groß-/Kleinschreibung nicht beachtet (**ITest** und **iTEST** werden als derselbe Name interpretiert).
- Es gibt keine Beschränkungen hinsichtlich der Namenslänge.
- Der Name darf nicht mit einem der reservierten Schlüsselwörter übereinstimmen (z. B. VAR, INT usw.).
- Es wird empfohlen, für Variablennamen die ungarische Notation und den LowerCamelCase-Stil zu verwenden.

12.1.1 Bitweiser Zugriff auf Variablen

Für Variablen vom Typ UDINT ist ein bitweiser Zugriff möglich. Um auf ein Bit zuzugreifen, verwenden Sie folgendes Format:

```
Variable.Bit_Nummer
```

Beispiel

```
Variable.1 := TRUE;
Variable.31 := FALSE;
```

Die Bit-Nummer muss als Ganzzahl von 0 bis 31 angegeben werden. Die Verwendung von Variablen als Bit-Nummern ist nicht zulässig.

12.1.2 ST-Funktionen in anderen ST-Programmelementen verwenden

Eine Funktion kann innerhalb einer anderen Funktion oder eines Funktionsbausteins aufgerufen werden. Dazu müssen Sie das folgende Format verwenden (informeller Aufruf):
Funktionsname (durch Komma getrennte Liste der Funktionseingaben)
Beispiel

```
FUNCTION rFun1: REAL;

    VAR_INPUT
        rIn1 : REAL;
        rIn2 : REAL;
    END_VAR

    rFun1 := rIn1 + rIn2;
```

```
END_FUNCTION
```

Es gibt eine Funktion **rFun1**:
Sein Aufruf in der Funktion **rFun2** sieht folgendermaßen aus:

```
FUNCTION rFun2: REAL;

    VAR_INPUT
        rIn1_0 : REAL;
        rInf1_1 : REAL;
        rInf1_2 : REAL;
    END_VAR

    rFun2 := rIn1_0 * rFun1(rInf1_1, rInf1_2);

END_FUNCTION
```

Die Funktion **rFun2** gibt eine Zahl zurück, die dem Produkt von **rIn1_0** und der Summe von **rInf1_1** und **rInf1_2** entspricht, die am Eingang dieser Funktion angegeben sind.

12.1.3 Verwendung eines Funktionsbausteins in einem anderen Funktionsbaustein

Instanzen eines Funktionsbausteins können in einem anderen Funktionsbausteinen erstellt werden, und zwar im Deklarationsbereich lokaler Variablen im Format:
(Kurzbezeichnung):(Funktionsbausteinname)
Sobald ein Funktionsblock instanziiert wurde, können Sie mit seinen Daten arbeiten. Die Eingänge einer Blockinstanz sind extern beschreibbar. Ausgänge sind zum Lesen vorgesehen.
Sie können eine Instanz auf verschiedene Arten aufrufen. Als Beispiel betrachten wir den Aufruf eines Zählers zum Vorwärtszählen, der beim Erstellen eines Funktionsbausteins als Vorlage angelegt wird:

1. Durch einen formellen Aufruf:

```
FUNCTION_BLOCK fb2

    VAR_INPUT
        xIn : BOOL;
    END_VAR

    VAR_OUTPUT
        xAlarmMax : BOOL;
        udiQ : UDINT;
    END_VAR

    VAR
        fb1 : functionblock1; //Deklaration einer Funktionsbausteininstanz
    END_VAR

    fb1 (U := xIn, Q => udiQ); //Aufrufen einer Funktionsbausteininstanz
    IF fb1.Q > 10 THEN
        xAlarmMax := TRUE;
    END_IF
```

```
END_FUNCTION_BLOCK
```

HINWEIS
Achten Sie auf den speziellen Operator zum Kopieren eines Wertes aus den Ausgabevariablen des Blocks („=>“).

2. Durch Zugriff auf die Ein-/Ausgänge eines Funktionsblocks (Variablendeklarationen sind ähnlich):

```
fb1.U := xIn; //Daten in den FB-Eingang schreiben
fb1 (); //eine FB-Instanz aufrufen
udiQ := fb1.Q; //Ausgabe der FB-Instanz lesen
IF fb1.Q > 10 THEN
    xAlarmMax := TRUE;
END_IF
```

Der **fb2**-Ausgang zeigt an, dass 10 Impulse überschritten wurden, und den kumulierten Zählerwert.

ACHTUNG
Sie können im Hauptteil einer Funktion keine Instanz eines Funktionsbausteins deklarieren.

ACHTUNG
In einem ST-Funktionsbaustein beträgt die maximale Verschachtelung von Bausteinen nicht mehr als 8.

ACHTUNG
ST-Funktionsbausteine unterstützen keine Variablen vom Typ RETAIN.

ACHTUNG
ST-Funktionsbausteine reservieren Speicherplatz im ROM-Speicher, nachdem sie zur Projektbibliothek hinzugefügt wurden, unabhängig davon, ob sie im Projekt verwendet werden oder nicht.

12.1.4 Kommentare im ST-Editor

Der ST-Editor in ALP unterstützt die Kommentarfunktion. Es stehen zwei Arten von Kommentaren zur Verfügung:

1. Einzeilig. Als Markierung dient ein doppelter Schrägstrich – //
2. Mehrzeilig. Die Markierung lautet:
   – (* – Anfang des Kommentars
   – *) – Ende des Kommentars

12.1.5 ST-Elemente zwischen Projekten kopieren

So kopieren Sie ST-Elemente zwischen Projekten:

1. Wählen Sie die ST-Elemente aus, die in das ursprüngliche Projektdiagramm kopiert werden müssen.
2. Kopieren Sie die ausgewählten Elemente mit der Tastenkombination **Strg+C** oder über das Kontextmenü.
3. Öffnen Sie das Projekt, um die kopierten Elemente einzufügen.
4. Fügen Sie Elemente mit der Tastenkombination **Strg + V** oder über das Kontextmenü in das zweite Projektdiagramm ein.

Als Ergebnis werden Vorlagen aller kopierten Elemente zum entsprechenden Abschnitt der Projektkomponentenbibliothek hinzugefügt.

HINWEIS

1. Wenn die Komponentenbibliothek die für die kopierte Komponente angegebene Gruppe nicht enthält, wird sie erstellt.
2. Wenn die kopierte Komponentengruppe nicht angegeben ist, wird die Vorlage dem Ordner **Andere** hinzugefügt.

Beim Kopieren bleiben sämtliche Verbindungen zwischen allen Komponenten erhalten, die beim Kopiervorgang berücksichtigt wurden.

ACHTUNG

Sollte eines der kopierten Elemente einen Fehler enthalten, wird das Einfügen der Elemente in das Projektdiagramm abgebrochen.
Es wird eine Fehlermeldung mit dem Namen des falschen Elements angezeigt.
Andere Komponenten werden in der Komponentenbibliothek in den entsprechenden Abschnitten angezeigt.

12.2 Dokumentation im ST-Editor

Der ST-Editor unterstützt die Dokumentationsfunktion.
Der Dokumentationsmarker ist der dreifache Schrägstrich „///“.
Über den Objekten wird Dokumentation hinzugefügt: Funktions-/Funktionsbausteindeklaration, Eingabe-/Ausgabevariablendeklaration.
Tags zur Dokumentation sind in der folgenden Tabelle aufgeführt.

Tag	Beschreibung
<Description>… </Description>	Beschreibung des Programmelements (Funktion, Funktionsbausteine, Eingangs- und Ausgangsvariable (nur Funktionsblock))
<Author>…</Author>	Name des Erstellers der Funktion oder des Funktionsbausteins
<GroupName>… </GroupName>	Gruppenname zum Gruppieren einer Funktion oder eines Funktionsblocks in der Bibliothek
<OutputDescription>… </OutputDescriotion>	Beschreibung der Funktionsausgabe

Dokumentieren der Funktion

```
///<Description>Widerstandstemperaturfühler (Pt1000)</Description>
///<OuptutDescription>Temperatur</OutputDescription>
///<Author>akYtec</Author>
///<GroupName>Temperatursensor</GroupName>

FUNCTION f_PT1000: REAL; // Funktion für PT1000 RTD-Sensor

    VAR_INPUT
        ///<Description>Widerstand</Description>
        R : REAL;
    END_VAR

    VAR_OUTPUT
        ///<OutputDescription>Temperatur</OutputDescription>
        Q : REAL;
    END_VAR
```

Dokumentieren des Funktionsbausteins

```
///<Description>Zähler zum direkten Zählen</Description>
///<Author>akYtec</Author>
///<GroupName>Timer und Zähler</GroupName>

FUNCTION_BLOCK fb_Counter
    VAR_INPUT
        ///<Description>Pulsdetektor</Description>
        U : BOOL; //boolesche Eingabevariable

        ///<Description>Zähler-Reset-Flag</Description>
        Res : BOOL; //boolesche Eingabevariable

        ///<Description>Voreingestellter Zählerwert</Description>
        N : UDINT; //boolesche Eingabevariable
    END_VAR

    VAR_OUTPUT
        ///<Description>Zählerwert</Description>
        Q : UDINT; //UDINT-Ausgabewert
    END_VAR
```

12.3 Datentypen

In der akYtec ALP unterstützte Datentypen:

Datentyp	Beschreibung	Gültiger Bereich	Größe
BOOL	Boolescher Wert	FALSE, TRUE	4 Byte
UDINT	Vorzeichenlose doppelte Ganzzahl	0…4294967295	4 Byte
REAL	Gleitkommazahl	$-1{,}2\times10^{-38}\ldots3{,}4\times10^{38}$	4 Byte
TIME	Zeitintervall	T#0…4294967295ms T#0...4294967s T#0...71582m T#0...1193h T#0...49d T#0...49d17h02m47s295ms	4 Byte
DT	Datum und Zeit	DT#2000-01-01-00:00:00..2136-02-07-6:28:15	4 Byte
ARRAY	Array	—	—

Der Datentyp einer Variablen bestimmt die Art der Informationen, den Darstellungsbereich und die Menge der zulässigen Operationen.
Die Variable kann erst nach ihrer Deklaration verwendet werden. Den Wert einer Variable einer anderen zuzuweisen ist nur möglich, wenn sie vom gleichen Typ sind. Andernfalls muss ein Typkonverter verwendet werden.

HINWEIS
Die Konvertierung eines größeren Typs in einen kleineren kann zu Informationsverlust führen.

12.3.1 Reservierte Schluesselwoerter

HINWEIS
In der Tabelle sind die verwendbaren Wörter **fett** hervorgehoben.

ABS	END_TRANSITION	**RETURN**	**SYS.TOF**
ACTION	END_TYPE	SEL	**SYS.TON**
AND	**END_VAR**	**SHL**	**SYS.TP**
ARRAY	**END_WHILE**	**SHR**	TASK
AT	ENO	SINT	**THEN**
BEGIN	**EXIT**	STEP	**TIME**
BOOL	F_EDGE	STRING	TIME_OF_DAY
BOOL_TO_REAL	**FALSE**	STRUCT	**TIME_TO_UDINT**
BOOL_TO_UDINT	**FOR**	**SYS.ACOS**	**TO**
BY	**FROM**	**SYS.ASIN**	TOD
CASE	**FUNCTION**	**SYS.ATAN**	TRANSITION
CD32	**FUNCTION_BLOCK**	**SYS.BLINK**	**TRUE**
CONFIGURATION	**GET_DATE_TIME**	**SYS.CLOCK**	TYPE
CONTINUE	**GET_TIME**	**SYS.CLOCKWEEK**	**UDINT**
DATE	**IF**	**SYS.COMPARE_ DATE_TIME**	**UDINT_TO_BOOL**
DATE_AND_TIME	INITIAL_STEP	**SYS.COS**	**UDINT_TO_DT**
DC32	INT	**SYS.CT**	**UDINT_TO_REAL**
DINT	LINT	**SYS.CTN**	**UDINT_TO_TIME**
DO	LREAL	**SYS.CTU**	UINT
DT	LWORD	**SYS.DTRIG**	ULINT
DT_TO_UDINT	**MOD**	**SYS.EXP**	**UNTIL**
DWORD	NON_RETAIN	**SYS.FACT**	USINT
ELSE	**NOT**	**SYS.FMAX**	**VAR**
ELSIF	**OF**	**SYS.FMIN**	VAR_ACCESS
EN	ON	**SYS.FTRIG**	VAR_CONFIG
END	**OR**	**SYS.IS_LEAP_YEAR**	VAR_EXTERNAL
END_ACTION	**POW**	**SYS.LG**	VAR_GLOBAL
END_CASE	PROGRAM	**SYS.LN**	VAR_IN_OUT
END_ CONFIGURATION	R_EDGE	**SYS.LOG**	**VAR_INPUT**
END_FOR	READ_ONLY	**SYS.MAX**	**VAR_OUTPUT**
END_FUNCTION	READ_WRITE	**SYS.MIN**	VAR_TEMP
END_FUNCTION_ BLOCK	**REAL**	**SYS.RS**	**WHILE**

END_IF	**REAL_TO_BOOL**	**SYS.RTRIG**	WITH
END_REPEAT	**REAL_TO_UDINT**	**SYS.SIN**	WORD
END_RESOURCE	**REPEAT**	**SYS.SQRT**	WSTRING
END_STEP	RESOURCE	**SYS.SR**	**XOR**
END_STRUCT	RETAIN	**SYS.TAN**	

12.3.2 Arrays

Syntax der Array-Deklaration

Zum Deklarieren eines Arrays wird die folgende Syntax verwendet:

```
<Array name>:[Index1..IndexN]OF <ELEMENT TYPE>:=[Default value of the element1,
Default value of the element2, Default value of the elementN];
```

Wobei:
Index1 - Nummer des ersten Elements des Arrays
IndexN - Nummer des letzten Elements des Arrays
Regel für Indizes:

– Die IndexN-Nummer darf nicht kleiner oder gleich der Index1-Nummer sein.
– Indizes dürfen nur ganzzahlig und nur positiv sein.

Beispiele für Array-Deklarationen:

```
Mass1: ARRAY[0..1] OF UDINT;
//Deklaration eines Arrays bestehend aus zwei Elementen vom Typ UDINT

Mass1: ARRAY[0..1] OF UDINT :=[1, 5];
//Initialisierung von Array-Elementen beim Deklarieren eines Arrays

Mass1: ARRAY[0] OF UDINT :=[1, 5];
//Fehler: Array-Dimension muss als <udint>...<udint> angegeben werden

Mass1: ARRAY[] OF UDINT;
//Fehler: Array-Dimension muss als <udint>...<udint> angegeben werden
```

Fehler beim Deklarieren eines Arrays:

```
Mass1: ARRAY[0] OF UDINT :=[1, 5]; //Fehler: Die Array-Dimension muss
angegeben werden als <integer>...<integer>
Mass1: ARRAY[] OF UDINT; //Fehler: Die Array-Dimension
muss angegeben werden als <integer>...<integer>
```

Syntax für den Zugriff auf Array-Elemente

Um auf Array-Elemente zuzugreifen, verwenden Sie die folgende Syntax:

```
<Array name>[Index1];
```

Beispiele für die Arbeit mit Array-Elementen:

```
VAR:=Mass1[5];
//Zuweisen des Wertes aus dem fünften Array-Element zur Variable VAR
```

```
VAR:=Mass1[5]+5*2;
VAR:=UDINT_TO_BOOL(Mass1[5]);
    //Zuweisen des konvertierten Wertes zur Variable VAR aus dem fünften Element

VAR:=(Mass1[i]);
    //Zuweisen des Wertes aus dem i-Element des Arrays
    //zur Variable VAR, wobei I eine Variable (lokal/Eingabe) ist
```

Beim Arbeiten mit Arrays gelten folgende Regeln:

1. Ein Array kann nur eindimensional sein.
2. Die Anzahl der Array-Elemente ist auf 32768 Elemente begrenzt.
3. Array-Elemente können folgende Typen haben: BOOL, UDINT, FLOAT, TIME und DT.
4. Array-Unterstützung für Funktionen:
   – Eine Funktion kann eine Variable vom Datentyp Array zurückgeben.

     Beispiel einer Funktion, die ein Array zurückgibt:

```
//Die Funktion MyFunction gibt eine Variable vom Typ Array zurück
function MyFunction: ARRAY [0..1] OF  UDINT ;
   var_input
      //Eingabevariablen vom Array-Typ
       Mass1 : ARRAY [0..1] OF  UDINT ;
   end_var
      //Das Rückgabearray wird im Funktionskörper erstellt
MyFunction [0]:= Mass1 [0];
MyFunction [1]:= Mass1 [1];
end_function
```

   – Die Deklaration einer Variable vom Array-Typ innerhalb einer Funktion erfolgt in den Abschnitten VAR...END_VAR und/oder VAR_INPUT...END_VAR.

     Beispiel für die Deklaration von Array-Variablen in verschiedenen Abschnitten einer Funktion:

```
//Die Funktion MyFunction gibt eine Variable vom Typ Array zurück.
function MyFunction: ARRAY [0..1] OF  UDINT ;
   var_input
      //Eingabevariablen vom Array-Typ
       Mass1 : ARRAY [0..1] OF  UDINT ;
   end_var
   var
      //llokale Variablen vom Array-Typ
       Mass2 : ARRAY [0..1] OF  UDINT ;
   end_var
end_function
```

   Für Funktionsbausteine:

   – Die Deklaration einer Variable vom Array-Typ in einem Funktionsbaustein erfolgt in den Abschnitten VAR...END_VAR, VAR_INPUT...END_VAR und/oder VAR_OUTPUT...END_VAR.

```
function_block MyFunctionblock
   var_input
      //Deklaration einer Eingabevariablen vom Typ Array
```

```
        Mass1 : ARRAY [0..1] OF  UDINT ;
    end_var
    var_output
       //Deklaration einer Ausgabevariablen vom Typ Array
        Mass2 : ARRAY [0..1] OF  UDINT ;
    end_var
       //Deklaration einer lokalen Variable vom Typ Array
        Mass3 : ARRAY [0..1] OF  UDINT ;
    end_var
  end_function_block
```

5. Die Folge der Standardwerte für Array-Elemente ist in eckige Klammern eingeschlossen.
6. Beim Initialisieren von Array-Elementen muss die Anzahl der Standardwerte mit der Anzahl der Array-Elemente übereinstimmen.

   Beispiel:

```
VAR
    Mass1: ARRAY[0..1] OF UDINT:=[1,5]; //keine Fehler
    Mass1: ARRAY[0..1] OF UDINT:=[1,5,7]; //Fehler
    Mass1: ARRAY[0..1] OF UDINT:=[1]; //Fehler
END_VAR
```

Einschränkungen bei der Arbeit mit Arrays:

– Der bitweise Zugriff auf Array-Elemente wird nicht unterstützt.

  Beispiel:

```
Mass1: ARRAY[0..1] OF UDINT:=[1,5];
Mass1[1].1 : = TRUE; //Fehler
```

– Das Wiederholen einer Sequenz von Array-Elementen wird nicht unterstützt.

  Beispiel:

```
Mass1: ARRAY[0..1] OF UDINT:=[a, b, c(N)];
//wobei (N) für Wiederholungssequenzen a, b, c (N-mal) steht)
Mass1: ARRAY[0..1] OF UDINT:=[1,5,7(2)]; //Fehler
```

– Das Wiederholen einer Sequenz von Array-Elementen wird nicht unterstützt.

  Beispiel:

```
Mass1[1, 5]:=1;
//Fehler: Zuweisen des Werts 1 zum ersten und fünften Array-Element
```

Beispiel für die *Verwendung von Arrays*.

12.4 Sprachstrukturen

Zu den Sprachstrukturen von ST gehören:

– *Rechenoperationen*
– *Bitoperationen*
– *Datentypkonvertierung*
– *logische Operationen*
– *relationale Operationen*
– *Zuweisung*

– ***IF – ELSIF – ELSE**-Anweisung*
– ***CASE**-Anweisung*
– ***RETURN**-Anweisung*
– ***FOR**-Anweisung*;
– ***WHILE**-Anweisung*;
– ***REPEAT – UNTIL**-Anweisung*.

HINWEIS
Beim Schreiben von Ausdrücken ist die Verwendung von Variablen (Eingabe-, Ausgabe- und lokale Variablen) sowie Konstanten zulässig.

12.4.1 Operationen

12.4.1.1 Rechenoperationen

Operation	Operator	Datentyp	Beispiel
Addition	+	IN, OUT: UDINT/ REAL	OUT := IN1 + IN2 + …
Multiplikation	*		OUT := IN1 * IN2 * …
Subtraktion	-		OUT := IN1 - IN2
Division	/		OUT := IN1 / IN2
Modulo	MOD		OUT := IN1 MOD IN2
Absoluter Wert	ABS (IN)	IN, OUT: REAL	OUT := ABS (IN1)
Potenzierung	POW (IN, N) IN – Basis N – Exponent	IN, N, OUT: REAL	OUT := POW (IN1, N)

Das Ergebnis der Rechenoperation ist das mathematische Ergebnis des Ausdrucks.
Die Priorität einer Operation bestimmt die Reihenfolge ihrer Ausführung im Ausdruck. Klammern sind zulässig, um die Berechnungsreihenfolge in arithmetischen Ausdrücken zu definieren.

12.4.1.2 Bitoperationen

Operation	Operator	Datentyp	Beispiel
Bitweises Verschieben nach links	SHL (IN, N)	IN, OUT: UDINT N: 1..32	OUT := SHL (IN1, N)
Bitweises Verschieben nach rechts	SHR (IN, N)		OUT := SHR (IN1, N)
Decoder: Konvertiert Binärcode in Positionscode	DC32 (IN)	IN, OUT: UDINT	OUT := DC32 (IN1)
Encoder: Konvertiert Positionscode in Binärcode	CD32 (IN)	IN, OUT: UDINT	OUT := CD32 (IN1)

12.4.1.3 Datentypkonvertierung

Operation	Operator	Datentyp	Beispiel
UDINT in REAL	UDINT_TO_REAL (IN)	IN: UDINT OUT: REAL	OUT := UDINT_TO_ REAL (IN)
UDINT in BOOL	UDINT_TO_BOOL (IN)	IN: UDINT OUT: BOOL	OUT := UDINT_TO_ BOOL (IN)

Operation	Operator	Datentyp	Beispiel
UDINT in TIME	UDINT_TO_TIME (IN)	IN: UDINT OUT: TIME	OUT := UDINT_TO_ TIME (IN)
UDIMT in DT	UDINT_TO_DT (IN)	IN: UDINT OUT: DT	OUT := UDINT_TO_ DT (IN)
REAL in UDINT	REAL_TO_UDINT (IN)	IN: REAL OUT: UDINT	OUT := REAL_TO_ UDINT (IN)
REAL in BOOL	REAL_TO_BOOL (IN)	IN: REAL OUT: BOOL	OUT := REAL_TO_ BOOL (IN)
BOOL in REAL	BOOL_TO_REAL (IN)	IN: BOOL OUT: REAL	OUT := BOOL_TO_ REAL (IN)
BOOL in UDINT	BOOL_TO_UDINT (IN)	IN: BOOL OUT: UDINT	OUT := BOOL_TO_ UDINT (IN)
TIME in UDINT	TIME_TO_UDINT (IN)	IN: TIME OUT: UDINT	OUT := TIME_TO_ UDINT (IN)
DT in UDINT	DT_TO_UDINT (IN)	IN: DT OUT: UDINT	OUT := DT_TO_ UDINT (IN)

12.4.1.4 Logische Operationen

Operation	Operator	Datentyp	Beispiel
Logische Negation	NOT	IN, OUT: BOOL	OUT := NOT IN1
Boolesche Multiplikation	AND &		OUT := IN1 AND IN2 OUT := IN1 & IN2
Boolesche Addition	OR		OUT := IN1 OR IN2
Logisches (bitweises) „exklusives OR“	XOR		OUT := IN1 XOR IN2

12.4.1.5 Relationale Operationen

Operation	Operator	Datentyp	Beispiel
größer als	>	IN: UDINT/REAL OUT: BOOL	OUT := IN1 > IN2
größer oder gleich	>=		OUT := IN1 >= IN2
gleich	=		OUT := IN1 = IN2
kleiner oder gleich	<=		OUT := IN1 <= IN2
kleiner als	<		OUT := IN1 < IN2
ungleich	<>		OUT := IN1 <> IN2

12.4.1.6 Prioritaeten der Operationen

HINWEIS
Die Operationen in der Tabelle sind **von der höchsten zur niedrigsten Priorität** sortiert. Je höher die Priorität einer Operation, desto schneller wird sie ausgeführt.

Operation	Operator
Klammern	(Ausdruck)
Aufrufen einer Funktion und eines Funktionsbausteins	Beispiel: fb1(); function1 := … ;
Bitoperationen	
Unäres Minus	–
Logische Negation	NOT
Potenzierung	POW
Multiplikation	*
Division	/
Modul	MOD
Addition	+
Subtraktion	–
Relationale Operationen	>, <, <=, >=
Gleich	=
Ungleich	<>
Konjunktion Logische Multiplikation „AND“	& AND
Exklusives OR	XOR
Disjunktion Logische Addition „OR“	OR

12.4.2 Zuweisungsvorgang

Das gepaarte Symbol „**:=**“ wird verwendet, um eine Zuweisung anzuzeigen. Die rechte und linke Seite des Ausdrucks müssen Operanden desselben Typs enthalten (eine automatische Typumwandlung ist nicht vorgesehen). Auf der linken Seite des Ausdrucks (Empfängerseite) kann nur eine Variable verwendet werden. Die rechte Seite kann einen Ausdruck oder eine Konstante enthalten.

12.4.3 IF-Anweisung

Mit dem Operator **WENN** können Sie eine oder mehrere Bedingungen prüfen und, wenn mindestens eine der Bedingungen erfüllt ist, die angegebenen Ausdrucksbedingungen ausführen. Nach der Ausführung der Ausdrücke verlässt der Operator die Anweisung, d. h. die übrigen Bedingungen werden nicht mehr geprüft.
Betrachten wir die Arbeit des Bedieners am Beispiel der Signalisierung, dass der Temperaturwert die zulässigen Grenzen überschreitet:

```
FUNCTION_BLOCK fb //Name des Funktionsbaustein

    VAR_INPUT //Deklaration von Eingabevariablen
        rTemp : REAL;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        xHigh : BOOL;
        xLow : BOOL;
    END_VAR
```

```
    VAR //Deklaration lokaler Variablen
        rHighTemp : REAL := 20;
        rLowTemp : REAL := 10;
    END_VAR

    //Codebereich

    IF rTemp > rHighTemp THEN
        xHigh := TRUE;
    ELSIF rTemp < rLowTemp THEN
        xLow := TRUE;
    ELSE
        xHigh := FALSE;
        xLow := FALSE;
    END_IF

END_FUNCTION_BLOCK
```

Wenn die Bedingung in der **IF**-Anweisung erfüllt ist (der Wert der Variable **rTemp** ist größer als **rHighTemp**), wird der Variable **xHigh** der Wert **TRUE** zugewiesen und die Anweisung wird beendet (die nächste Bedingung wird nicht geprüft). Wenn die Bedingung nicht erfüllt ist, wird die nächste Bedingung in der verschachtelten **ELSIF**-Anweisung geprüft. Wenn die Bedingung in **ELSIF** erfüllt ist (der Wert der Variable **rTemp** ist kleiner als **rLowTemp**), wird der Variable **xLow** der Wert **TRUE** zugewiesen und die Anweisung wird beendet (die nächste Bedingung wird nicht geprüft). Wenn keine der Bedingungen in **IF** und **ELSIF** erfüllt ist (d. h., der Temperaturwert liegt innerhalb akzeptabler Grenzen), werden die in der verschachtelten **ELSE-Anweisung platzierten Ausdrücke ausgeführt – die Zuweisung des Wertes FALSE an die Variablen xHigh und xLow.**
Die Verwendung verschachtelter **ELSIF**- und **ELSE**-Anweisungen ist optional. Innerhalb einer **IF**-Anweisung können beliebig viele **ELSIF**-Anweisungen platziert werden.
Die Konstruktion erlaubt eine Verschachtelung, d. h. innerhalb eines **IF** kann ein weiteres stehen usw. Außerdem können innerhalb des **IF**-Operators Schleifen und der ***CASE 12.4.4***-Operator verwendet werden.

12.4.4 CASE-Anweisung

Mit dem Operator CASE können Sie den Wert einer Integer-Variable (Selektor) mit mehreren Konstanten oder Integer-Werten (Labels) vergleichen. Bei Übereinstimmung werden die dem Label zugeordneten Ausdrücke ausgeführt.
Beispiel:

```
FUNCTION_BLOCK fb1 //Name des Funktionsbaustein

    VAR_INPUT //Deklaration von Eingabevariablen
        udiSel : UDINT;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        xOut1 : BOOL;
        xOut2 : BOOL;
        xOut3 : BOOL;
        xOut4 : BOOL
```

```
    END_VAR

    //Codebereich

    xOut1 := FALSE;
    xOut2 := FALSE;
    xOut3 := FALSE;
    xOut4 := FALSE;

    CASE udiSel OF
        0:
            xOut1 := TRUE;
        1..3:
            xOut2 := TRUE;
        4, 6:
            xOut3 := TRUE;
    ELSE
        xOut4 := TRUE;
    END_CASE

END_FUNCTION_BLOCK
```

Wenn der **udiSel**-Wert:

- Gleich 0 ist, dann nimmt **xOut1** den Wert **TRUE** an
- Im Bereich 1..3 liegt, nimmt **xOut2** den Wert **TRUE** an
- Gleich 4 oder 6 ist, dann nimmt **xOut3** den Wert **TRUE** an
- Unter keinen der angegebenen Werte fällt, dann nimmt **xOut4** den Wert **TRUE** an

Wie aus dem Beispiel ersichtlich, kann ein Label mehrere Werte enthalten, die durch Kommas getrennt aufgelistet sind: „4, 6“ oder den Bereich „1..3“. In diesem Fall dürfen die Werte eines der Labels nicht mit den Werten der anderen übereinstimmen. Außerdem muss bei der Angabe eines Wertebereichs der Anfang des Bereichs kleiner sein als sein Ende.
Die verschachtelte **ELSE**-Anweisung ist optional; die darin platzierten Ausdrücke werden ausgeführt, wenn der Selektorwert mit keinem der Labels übereinstimmt.
Die zum Behandeln der einzelnen **CASE**-Anweisungsfälle bereitgestellten Aktionen können Schleifen, **IF**-Anweisungen und **CASE**-Anweisungen verwenden.

12.4.5 RETURN-Anweisung

Mit der **RETURN**-Anweisung können Sie ein Programmobjekt verlassen.
Anwendungsbeispiel:

```
IF xDone THEN
    RETURN;
END_IF;

udiCounter := udiCounter + 1;
```

Wenn die Variable **xDone** den Wert **TRUE** annimmt, wird der Ausdruck „**udiCounter** := **udiCounter** + 1“ nicht ausgeführt (wie alles, was weiter unten im Programmkörper steht).

12.4.6 FOR-Anweisung

Der FOR-Operator wird verwendet, um eine Schleife mit einer vorgegebenen Anzahl von Iterationen auszuführen. Er wird häufig für Operationen über Datenstrukturen wie Arrays verwendet.
Die Operatoren **IF** und **CASE** sowie andere Schleifenoperatoren können innerhalb einer Schleife verwendet werden.
Betrachten Sie als Beispiel die Implementierung der Bubblesortierung vom Kleinsten zum Größten:

```
FUNCTION_BLOCK MaxI_MinI
//Maximale und minimale Zahlen mithilfe der Bubblesortierung
//vom Kleinsten zum Größten

    VAR_INPUT //Deklaration von Eingabevariablen
        udiX1, udiX2, udiX3, udiX4, udiX5 : UDINT;
//Hinzufügen der erforderlichen Anzahl von Eingabevariablen
//und Definieren des Datentyps
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        udiMaxI, udiMinI : UDINT;
    END_VAR

    VAR //Deklaration lokaler Variablen
        udiI, udiJ, udiN, udiK : UDINT;
        audiX : ARRAY [1..5] OF UDINT;
//Gibt den Bereich des Arrays und den Datentyp des Arrays an
    END_VAR

    //Codebereich
    //Array-Deklaration
    audiX[1] := udiX1;
    audiX[2] := udiX2;
    audiX[3] := udiX3;
    audiX[4] := udiX4;
    audiX[5] := udiX5;
    udiN := 5; // die Anzahl der zu sortierenden Zahlen wird
               //angegeben (Array-Größe)

    FOR udiI := 1 TO udiN-1 DO
        FOR udiJ := 1 TO udiN-udiI DO
            IF audiX[udiJ] > audiX[udiJ+1] THEN
                udiK := audiX[udiJ];
                audiX[udiJ] := audiX[udiJ+1];
                audiX[udiJ+1] := udiK;
            END_IF;
        END_FOR;
```

```
    END_FOR;

    udiMaxI := audiX[udiN];
    //die letzte (maximale) Nummer des Arrays wird angezeigt
    udiMinI := audiX[1];
    //die erste (minimale) Nummer des Arrays wird angezeigt

END_FUNCTION_BLOCK
```

Die Variable **udiI** wird als Iterator (Zähler) der Schleife bezeichnet. Diese Variable muss vom Typ einer vorzeichenbehafteten Ganzzahl (**UDINT**) sein. Nach jeder Ausführung des Schleifenkörpers ändert sich der Iteratorwert – standardmäßig auf **+1**. Der Benutzer kann den Iterator-„Schritt“ mithilfe des verschachtelten Operators **BY** festlegen. Danach erfolgt der Übergang sofort zur nächsten Iteration – das heißt, der gesamte Zyklus wird „von Anfang bis Ende“ ausgeführt. Die Struktur sieht folgendermaßen aus:

```
// udiI nimmt die Werte 1, 4, 7, 10 usw. an.
FOR udiI := 1 TO udiN-1 BY 3 DO
    .. // Schleifencode
END_FOR
```

Der Schleifeniterator (Zähler), seine Start- und Endwerte sowie sein Inkrement sind ganzzahlige Werte. Sie werden vor dem Eintritt in die Schleife berechnet, und das Ändern der Werte der in einem dieser Ausdrücke enthaltenen Variablen ändert nicht die Anzahl der Iterationen.
Sie können eine Schleife vorzeitig mit dem Operator **EXIT** verlassen. Beispielstruktur:

```
FOR udiI := 1 TO udiN-1 DO
    IF audiX[2] > 100 THEN
        EXIT;
    ELSE
        .. // Schleifencode
    END_IF
END_FOR
```

Mit dem Operator **CONTINUE** können Sie einen Schleifenschritt überspringen.

```
FOR udiI := 1 TO udiN-1 DO
    IF udiI = 3 THEN
        CONTINUE;
    ELSE
        .. // Schleifencode
        END_IF
END_FOR
```

12.4.7 WHILE-Anweisung

Der Operator **WHILE** wird verwendet, um eine Schleife mit einer unbekannten Anzahl von Iterationen zu erstellen. Die Schleife wird beendet, wenn die getestete Bedingung **FALSE** zurückgibt. In diesem Fall wird die Bedingung überprüft, BEVOR der Ausdruck ausgeführt wird (eine Schleife mit einer Vorbedingung). Wenn die Bedingung also sofort **FALSE** zurückgibt, wird die Schleife nicht einmal ausgeführt.
Die Operatoren **IF** und **CASE** sowie andere Schleifenoperatoren können innerhalb einer Schleife verwendet werden.
Beispiel

```
WHILE rVar < 100 DO
    rVar := rVar + 1;
```

```
END_WHILE
```

Das Ergebnis der Ausführung dieser Schleife (mit dem Anfangswert der Variablen **rVar** := 10) ist die Zahl 100.
Sie können eine Schleife vorzeitig mit der Anweisung **EXIT** verlassen.
Mit dem Operator **CONTINUE** können Sie einen Schleifenschritt überspringen.

12.4.8 REPEAT UNTIL-Anweisung

Die Anweisung **REPEAT** wird verwendet, um eine Schleife mit einer unbekannten Anzahl von Iterationen zu erstellen. Die Schleife wird beendet, wenn die getestete Bedingung **TRUE** zurückgibt. In diesem Fall wird die Bedingung überprüft, NACHDEM der Ausdruck ausgeführt wurde (eine Schleife mit einer Nachbedingung). Wenn die Bedingung also sofort **TRUE** zurückgibt, wird die Schleife einmal ausgeführt.
Die Operatoren **IF** und **CASE** sowie andere Schleifenoperatoren können innerhalb einer Schleife verwendet werden.
Beispiel:

```
REPEAT
    IF rVar > 100 THEN
        EXIT;
    END_IF;
    rVar := rVar + 1;
UNTIL rVar > 180
END_REPEAT;
```

Das Ergebnis der Ausführung dieser Schleife (mit dem Anfangswert der Variablen **rVar** := 10) ist die Zahl 101.
Um eine Schleife vorzeitig zu verlassen, können Sie den Operator **EXIT** verwenden.
Mit dem Operator **CONTINUE** können Sie einen Schleifenschritt überspringen.

12.5 Systemfunktionen

Der ST-Editor unterstützt die folgenden Systemfunktionen:

- *Datums- und Uhrzeitfunktionen*
- *Mathematische Funktionen*
- *Logarithmische Funktionen*
- *Trigonometrische Funktionen*

12.5.1 Datums- und Uhrzeitfunktionen

- *GET_TIME-Funktion*
- *GET_DATE_TIME-Funktion*
- *SYS.COMPARE_DATE_TIME-Funktion*
- *SYS.IS_LEAP_YEAR-Funktion*

12.5.1.1 GET_TIME-Funktion

Die Funktion **GET_TIME** gibt einen Wert vom Typ **TIME** (4 Bytes) zurück, der die seit dem letzten Einschalten des Geräts vergangene Zeit in Millisekunden enthält.
Beispiel

```
VAR
    Time_1 : TIME := T#0ms;
    Time_2 : TIME := T#0ms;
    Q :      BOOL := FALSE;
END_VAR
```

```
IF Time_1 = T#0ms THEN
   Time_1 := GET_TIME();
END_IF

Time_2 := GET_TIME();

IF (Time_2 - Time_1) >= T#1000ms THEN
    Q      := NOT Q;
    Time_1 := T#0ms;
    Time_2 := T#0ms;
END_IF
```

12.5.1.2 GET_DATE_TIME-Funktion

Die Funktion **GET_DATE_TIME** gibt einen Wert vom Typ **DT** (4 Bytes) zurück, der Echtzeituhrdaten in Sekunden seit 01.01.2000, 00:00:00, enthält und dabei die im Gerät eingestellte Zeitzone berücksichtigt.
Beispiel

```
VAR
    Ton_UDINT : UDINT;
    Ton_DT : DT;
END_VAR

Ton_DT := GET_DATE_TIME();

Ton_UDINT := DT_TO_UDINT(Ton_DT);
```

12.5.1.3 SYS.COMPARE_DATE_TIME-Funktion

Die Funktion **SYS.COMPARE_DATE_TIME** vergleicht zwei als Eingabe angegebene UDINT-Werte unter Verwendung einer angegebenen Datums-/Zeitmaske und gibt einen UDINT-Wert zurück, der wie folgt ausgewertet wird:

- 1 – Wert 1 ist größer als Wert 2
- 2 – Wert 1 ist kleiner als Wert 2
- 0 – Wert 1 ist gleich Wert 2

Der Vergleich erfolgt in Sekundenzahl ab 00:00:00 01.01.2000.
Beispiel

```
FUNCTION udiCompare: UDINT;

    VAR_INPUT
        udiValue1:  UDINT; //Wert 1
    END_VAR

    VAR
        udiValue2 : UDINT; //Wert 2
        udiMask :   UDINT := 63; //date/time mask
    END_VAR
```

```
    udiValue2  := DT_TO_UDINT (GET_DATE_TIME ());
    udiCompare := SYS.COMPARE_DATE_TIME (udiValue1, udiValue2, udiMask);

END_FUNCTION
```

Mask = 63 (0b111111) – Alle Bits werden verwendet:

- Bit 0: Sekunden
- Bit 1: Minuten
- Bit 2: Stunden
- Bit 3: Tage
- Bit 4: Monate
- Bit 5: Jahre

12.5.1.4 SYS.IS_LEAP_YEAR-Funktion

Die Funktion **SYS.IS_LEAP_YEAR** gibt einen BOOL-Wert zurück, der Daten darüber enthält, ob die am Funktionseingang bereitgestellte UDINT-Nummer einem Schaltjahr entspricht (1 – Schaltjahr, 0 – kein Schaltjahr).

Beispiel

```
FUNCTION xLeapYear: BOOL; //Funktionsname und Ausgabedatentyp

    VAR_INPUT //Deklaration von Eingabevariablen
        udiYear : UDINT; //Jahr für Prüfung
    END_VAR

    xLeapYear := SYS.IS_LEAP_YEAR (udiYear);

END_FUNCTION
```

12.5.2 Mathematische Funktionen

- *Mathematische Konstanten*
- *MIN/FMIN*
- *MAX/FMAX*
- *SQRT*
- *EXP*
- *FACT*

12.5.2.1 Mathematische Konstanten (CONST.*)

Name	Wert	Beschreibung
CONST.PI	3.1415927	Die Zahl Pi (π)
CONST.E	2.7182817	Euler-Zahl (e)
CONST.POS_INF	+ ∞	Positive Unendlichkeit
CONST.NEG_INF	– ∞	Negative Unendlichkeit

12.5.2.2 MIN/FMIN

Die Funktionen **MIN/FMIN** geben den kleinsten Wert von zwei Eingangsvariablen zurück. Eingabevariablen und ein Rückgabewert für die Funktion:

- **MIN** — für den UDINT-Typ (beide Argumente und Ergebnis)

– **FMIN** — für den REAL-Typ (beide Argumente und Ergebnis)

Beispiel 1

```
VAR
    Var_1: UDINT;
    Var_2: UDINT := 7;
    Var_3: UDINT := 5;
END_VAR
Var_1 := SYS.MIN(Var_2, Var_3); // Rückgabewert 5
```

Beispiel 2

```
VAR
    Var_1: REAL;
    Var_2: REAL := 7.1;
    Var_3: REAL := 8.2;
END_VAR
Var_1 := SYS.FMIN(Var_2, Var_3); // Rückgabewert 7.1
```

12.5.2.3 MAX/FMAX

Die **MAX/FMAX** Funktionen geben den größten Wert von zwei Eingangsvariablen zurück. Eingabevariablen und ein Rückgabewert für die Funktion:

– **MAX** — für den UDINT-Typ (beide Argumente und Ergebnis)

– **FMAX** — für den REAL-Typ (beide Argumente und Ergebnis)

Beispiel 1

```
VAR
    Var_1: UDINT;
    Var_2: UDINT := 7;
    Var_3: UDINT := 8;
END_VAR
Var_1 := SYS.MAX(Var_2, Var_3); // Rückgabewert 8
```

Beispiel 2

```
VAR
    Var_1: REAL;
    Var_2: REAL := 7.5;
    Var_3: REAL := 5.2;
END_VAR
Var_1 := SYS.FMAX(Var_2, Var_3); // Rückgabewert 7.5
```

12.5.2.4 SQRT

Die Funktion **SQRT** gibt die Quadratwurzel der eingegebenen Nummer zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.

Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 4.0;
END_VAR
Var_1 := SYS.SQRT(Var_2); // Rückgabewert 2.0
```

12.5.2.5 EXP

Die Funktion **EXP** gibt den Exponentenwert zurück, der in die Potenz der Eingabezahl gestiegen ist. Eine Eingabevariable und ein Rückgabewert vom Typ REAL.
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.0;
END_VAR
Var_1 := SYS.EXP(Var_2); // Rückgabewert 2.7182817
```

12.5.2.6 FACT

Die Funktion **FACT** gibt das Faktorium der Eingabezahl zurück. Die Funktion gibt die richtigen Werte zurück, wenn der Wert der Eingabevariablen im Intervall zwischen 0 und 12 liegt. In allen anderen Fällen gibt die Funktion 0 zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ UDINT.
Beispiel

```
VAR
    Var_1: UDINT;
    Var_2: UDINT := 5;
END_VAR
Var_1 := SYS.FACT(Var_2); // Rückgabewert 120
```

12.5.3 Logarithmische Funktionen

- *LOG*
- *LG*
- *LN*

12.5.3.1 LOG

Die Funktion **LOG** gibt den Logarithmus einer Eingabezahl mit einer beliebigen Basis zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.

Basis des Logarithmus	Zahl des Logarithmus	Ergebnis
< 0	beliebige	NaN*
NaN	1	0
+ inf**	0	NaN
+ inf	> 0	0
+ inf	+ inf	NaN
0	> 0, ≠ 1	0
0	0	NaN
0	+ inf	NaN
0	1	0
1	beliebige	NaN
NaN	NaN	NaN
> 0, ≠ 1	> 0	Zahl

Basis des Logarithmus	Zahl des Logarithmus	Ergebnis
> 0, ≠ 1	0	- inf***
> 0, ≠ 1	+ inf	+ inf

*NaN (Not-a-Number) — keine Zahl
**+ inf (infinity) — positive Unendlichkeit
***- inf (infinity) — negative Unendlichkeit
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 2; // Basis des Logarithmus
    Var_3: REAL := 8; // Zahl des Logarithmus
END_VAR
Var_1 := SYS.LOG(Var_2, Var_3); // Rückgabewert 3
```

12.5.3.2 LG

Die Funktion **LG** gibt den Dezimallogarithmus einer Eingabezahl zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.

Zahl des Logarithmus	Ergebnis
0	- inf*
< 0	NaN**
NaN	NaN
+ inf***	+ inf

*- inf (infinity) — negative Unendlichkeit
**NaN (Not-a-Number) — keine Zahl
**+ inf (infinity) — positive Unendlichkeit
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 10;
END_VAR
Var_1 := SYS.LG(Var_2); // Rückgabewert 1
```

12.5.3.3 LN

Die Funktion **LN** gibt den natürlichen Logarithmus einer Eingabezahl zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.

Zahl des Logarithmus	Ergebnis
0	- inf*
< 0	NaN**
NaN	NaN
+ inf***	+ inf

*- inf (infinity) — negative Unendlichkeit
**NaN (Not-a-Number) — keine Zahl
**+ inf (infinity) — positive Unendlichkeit
Beispiel

```
VAR
```

```
    Var_1: REAL;
    Var_2: REAL := 2.718281;
END_VAR
Var_1 := SYS.LN(Var_2); // Rückgabewert ~1
```

12.5.3.4 Trigonometrische Funktionen

- *SIN*
- *COS*
- *TAN*
- *ASIN*
- *ACOS*
- *ATAN*

12.5.3.4.1 SIN

Die Funktion **SIN** gibt den Sinus-Wert (sin) einer Eingabezahl zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.5708; // im Bogenmaßs
END_VAR
Var_1 := SYS.SIN(Var_2); Rückgabewert 1
```

12.5.3.4.2 COS

Die Funktion **COS** gibt den Kosinus-Wert (cos) einer Eingabezahl zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.0472; // im Bogenmaß
END_VAR
Var_1 := SYS.COS(Var_2); // Rückgabewert 0.49999794
```

12.5.3.4.3 TAN

Die Funktion **TAN** gibt den Tangens-Wert (tan) einer Eingabezahl zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.3734; // im Bogenmaß
END_VAR
Var_1 := SYS.TAN(Var_2); // Rückgabewert 4.999982
```

12.5.3.4.4 ASIN

Die Funktion **ASIN** gibt den Arkussinus-Wert (arcsin) einer Eingabezahl zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 0.5;
END_VAR
Var_1 := SYS.ASIN(Var_2); // Rückgabewert 0.52359825
```

HINWEIS
Der relative Fehler des Funktionsergebnisses ist 1e-5.

12.5.3.4.5 ACOS

Die Funktion **ACOS** gibt den Arkuskosinus-Wert (arccos) einer Eingabezahl zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 0.5;
END_VAR
Var_1 := SYS.ACOS(Var_2); // Rückgabewert 1.047198
```

12.5.3.4.6 ATAN

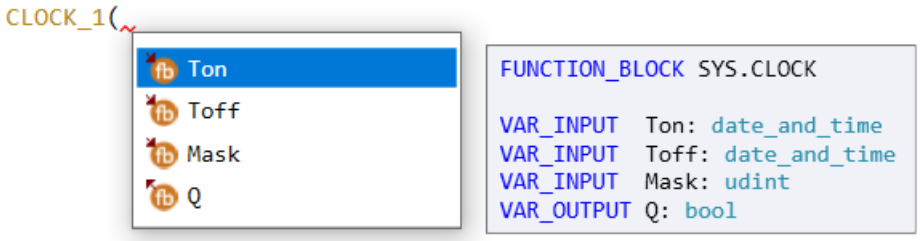
Die Funktion **ATAN** gibt den Arkustangens-Wert (arctg) einer Eingabezahl zurück.
Eine Eingabevariable und ein Rückgabewert vom Typ REAL.
Beispiel

```
VAR
    Var_1: REAL;
    Var_2: REAL := 1.73205; // 3^(1/2)
END_VAR
Var_1 := SYS.ATAN(Var_2); // Rückgabewert 1.0471973
```

12.6 Systemfunktionsbausteine

Der ST-Editor unterstützt die folgenden Systemfunktionsbausteine:
- *Trigger*
- *Timer*
- *Generatoren*
- *Zähler*

Die Verwendung von Systemfunktionsbausteine in anderen Programmelementen erfolgt ähnlich *unter Verwendung benutzerdefinierter Funktionsbausteine*.
Zu jedem Systemfunktionsbaustein gibt der Editor einen Hinweis zu seiner Verwendung.



12.6.1 Trigger

- *RS-Trigger mit RESET-Priorität (SYS.RS)*

- *SR-Trigger mit SET-Priorität (SYS.SR)*
- *Steigende Flanke (SYS.RTRIG)*
- *Fallende Flanke (SYS.FTRIG)*
- *D-Trigger (SYS.DTRIG)*

12.6.1.1 RS-Trigger mit RESET-Prioritaet (SYS.RS)

Der RS-Trigger mit RESET-Priorität (SYS.RS) dient zum Schalten mit Zustandsfixierung beim Empfang kurzer Impulse am entsprechenden Eingang. Am Ausgang Q erscheint an der Flanke des Signals am Eingang S ein Signal mit HIGH-Pegel.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
S	BOOL	SET-Eingang
R	BOOL	RESET-Eingang
Ausgänge		
Q	BOOL	Trigger-Ausgang

```
FUNCTION_BLOCK RS_trigger //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        R_in : BOOL;
        S_in : BOOL;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR //Deklaration lokaler Variablen
        RS_1: SYS.RS;
    END_VAR

    //Codebereich

    RS_1(R := R_in, S := S_in, Q => Q_out);

END_FUNCTION_BLOCK
```

Bei gleichzeitigem Signalempfang an beiden Eingängen hat das Signal vom Eingang R Vorrang.

12.6.1.2 SR-Trigger mit SET-Prioritaet (SYS.SR)

Der SR-Trigger mit SET-Priorität (SYS.SR) dient zum Schalten mit Zustandsfixierung beim Empfang kurzer Impulse am entsprechenden Eingang. Am Ausgang Q erscheint bei der Flanke des Signals am Eingang S ein HIGH-Pegel.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
S	BOOL	SET-Eingang

Bezeichnung	Datentyp	Beschreibung
R	BOOL	RESET-Eingang
Ausgänge		
Q	BOOL	Trigger-Ausgang

```
FUNCTION_BLOCK SR_trigger //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        S_in : BOOL;
        R_in : BOOL;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR //Deklaration lokaler Variablen
        SR_1: SYS.SR;
    END_VAR

    //Codebereich

    SR_1(S := S_in, R := R_in, Q => Q_out);

END_FUNCTION_BLOCK
```

Bei gleichzeitigem Empfang von Signalen an beiden Eingängen hat das Signal vom Eingang S Vorrang.

12.6.1.3 Steigende Flanke (SYS.RTRIG)

Die steigende Flanke (SYS.RTRIG) wird verwendet, wenn auf eine Zustandsänderung eines digitalen Eingangssignals reagiert werden muss. Bei der steigenden Flanke des Eingangs I wird am Ausgang Q ein einzelner Impuls erzeugt.

Bezeichnung	Datentyp	Beschreibung
Eingang		
I	BOOL	Trigger-Eingang
Ausgang		
Q	BOOL	Trigger-Ausgang

```
FUNCTION_BLOCK R_trigger //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        RT_in : BOOL;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
```

```
        RT_out : BOOL;
    END_VAR

    VAR //Deklaration lokaler Variablen
        RTrig_1: SYS.RTRIG;
    END_VAR

    //Codebereich

    RTrig_1(I := RT_in, Q => RT_out);

END_FUNCTION_BLOCK
```

12.6.1.4 Fallende Flanke (SYS.FTRIG)

Die fallende Flanke (SYS.FTRIG) wird verwendet, wenn auf eine Zustandsänderung eines digitalen Eingangssignals reagiert werden muss. An der Vorderflanke des Eingangs I wird am Ausgang Q ein einzelner Impuls erzeugt.

Bezeichnung	Datentyp	Beschreibung
Eingang		
I	BOOL	Trigger-Eingang
Ausgang		
Q	BOOL	Trigger-Ausgang

```
FUNCTION_BLOCK F_trigger //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        FT_in : BOOL;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        FT_out : BOOL;
    END_VAR

    VAR //Deklaration lokaler Variablen
        FTrig_1: SYS.FTRIG;
    END_VAR

    //Codebereich

    FTrig_1(I := FT_in, Q => FT_out);

END_FUNCTION_BLOCK
```

12.6.1.5 D-Trigger (SYS.DTRIG)

Der D-Trigger (SYS.DTRIG) wird verwendet, um einen Impuls zu erzeugen, der den Ausgang für die Zeitdauer des Impulses am D-Eingang einschaltet. Die Ausgabedauer wird mit der Taktfrequenz am C-Eingang synchronisiert.
Am Q-Triggerausgang erscheint vor den Taktimpulsen am C-Eingang ein Signal mit HIGH-Pegel, wenn am D-Eingang ein Signal mit HIGH-Pegel anliegt. Die Rückkehr des Q-Ausgangs zum Signal mit LOW-Pegel erfolgt vor den Taktimpulsen am C-Eingang, wenn am D-Eingang ein Signal mit HIGH-Pegel anliegt.
Der Eingang S zwingt den Ausgang Q in einen HIGH-Pegelzustand.
Der Eingang R ist der Prioritätseingang und setzt den Ausgang Q auf LOW-Pegel.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
S	BOOL	SET-Eingang
D	BOOL	Trigger-Eingang
C	BOOl	Taktfrequenz
R	BOOL	RESET-Eingang
Ausgänge		
Q	BOOL	Trigger-Ausgang

```
FUNCTION_BLOCK D_trigger //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        S_in : BOOL;
        D_in : BOOL;
        C_in : BOOL;
        R_in : BOOL;
    END_VAR

    VAR_OUTPUT //dDeklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR //Deklaration lokaler Variablen
        DTrig_1: SYS.DTRIG;
    END_VAR

    //Codebereich

    DTrig_1(S := S_in, D := D_in, C := C_in, R := R_in, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.2 Timer

- *Impuls (SYS.TP)*
- *Einschaltverzögerung (SYS.TON)*
- *Ausschaltverzögerung (SYS:TOF)*

– *Schaltuhr (SYS.CLOCK)*
– *Wochenschaltuhr (SYS:CLOCKWEEK)*

12.6.2.1 Impuls (SYS.TP)

Der Impuls (SYS.TP) dient zur Erzeugung eines Impulses zum Einschalten des Ausgangs für eine bestimmte Zeitspanne. Am Ausgang Q des Bausteins erscheint an der Flanke des Eingangssignals I ein Signal mit HIGH-Pegel. Nach dem Start reagiert der Ausgang Q während der Zeitspanne T nicht auf eine Änderung des Wertes des Eingangssignals. Nach Ablauf der Zeitspanne T wird das Ausgangssignal auf den LOW-Pegel zurückgesetzt.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
I	BOOL	Einschalten des Timers
T	TIME	Pulsdauer
Ausgänge		
Q	BOOL	Timer-Ausgang

```
FUNCTION_BLOCK TP_timer //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        I_in : BOOL := FALSE;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
        TP_1: SYS.TP;
    END_VAR

    //Codebereich
    TP_1(I := I_in, T := T#1000ms);
    //wobei ms Millisekunden, s Sekunden, m Minuten, h Stunden und d Tage sind.
    Q_out := TP_1.Q;
END_FUNCTION_BLOCK
FUNCTION_BLOCK TP_timer //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        I_in : BOOL := FALSE;
        T_in : UDINT := 5000;//Millisekundends
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
```

```
        TP_1: SYS.TP;
        T_time: TIME;
    END_VAR

    //Codebereich

    T_time := UDINT_TO_TIME(T_in);
    TP_1(I := I_in, T := T_time, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.2.2 Einschaltverzoegerung (SYS.TON)

Die Einschaltverzögerung (SYS.TON) wird für die Verzögerung der Signalübertragung verwendet. Der Zeitgeberausgang Q erzeugt ein Signal mit hohem Pegel mit einer Verzögerung relativ zur Eingangssignalfront I von mindestens der Dauer T und schaltet sich beim Abfallen des Eingangssignals ab.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
I	BOOL	Timerstart (bei steigender Flanke)
T	TIME	Verzögerung beim Einschalten
Ausgänge		
Q	BOOL	Timer-Ausgang

```
FUNCTION_BLOCK TON_timer //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        I_in : BOOL := FALSE;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
        TON_1: SYS.TON;
    END_VAR

    //Codebereich

    TON_1(I := I_in, T := T#1000ms);
    //wobei ms Millisekunden, s Sekunden, m Minuten, h Stunden und d Tage sind.
    Q_out := TON_1.Q;

END_FUNCTION_BLOCK
```

```
FUNCTION_BLOCK TON_timer //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        I_in : BOOL := FALSE;
        Ton_in : UDINT := 5000;//Millisekunden
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
        TON_1: SYS.TON;
        Ton_time: TIME;
    END_VAR

    //Codebereich

    Ton_time := UDINT_TO_TIME(Ton_in);
    TON_1(I := I_in, T := Ton_time, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.2.3 Ausschaltverzoegerung (SYS.TOF)

Die Ausschaltverzögerung (SYS.TOF) wird verwendet, um das Ausschalten des Ausgangs zu verzögern. Der Ausgang des Zeitgebers Q zeigt ein Signal mit hohem Pegel an der steigenden Flanke des Signals am Eingang I, der Countdown der Ausschaltverzögerungszeit T beginnt an jeder fallenden Flanke des Eingangssignals. Nachdem das Eingangssignal ausgeschaltet ist, zeigt der Ausgang ein Signal mit niedrigem Pegel mit einer Verzögerung von T.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
I	BOOL	Timerstart (bei fallender Flanke)
T	TIME	Verzögerung beim Einschalten
Ausgänge		
Q	BOOL	Timer-Ausgang

```
FUNCTION_BLOCK TOF_timer //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        I_in : BOOL := FALSE;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR
```

```
    VAR
        TOF_1: SYS.TOF;
    END_VAR

    //Codebereich

    TOF_1(I := I_in, T := T#1000ms);
    //wobei ms Millisekunden, s Sekunden, m Minuten, h Stunden und d Tage sind.
    Q_out := TOF_1.Q;

END_FUNCTION_BLOCK
FUNCTION_BLOCK TOF_timer //Name des Funktionsbaustein

    VAR_INPUT //Deklaration von Eingabevariablen
        I_in : BOOL := FALSE;
        Tof_in : UDINT := 5000;//Millisekunden
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
        TOF_1: SYS.TOF;
        Tof_time: TIME;
    END_VAR

    //Codebereich

    Tof_time := UDINT_TO_TIME(Tof_in);
    TOF_1(I := I_in, T := Tof_time, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.2.4 Schaltuhr (SYS.CLOCK)

Die Schaltuhr (SYS.CLOCK) wird verwendet, um einen Impuls zum Einschalten des Q-Ausgangs gemäß der Echtzeituhr zu erzeugen. Die Ausgangs-Einschaltzeit Ton und die Abschaltzeit Toff werden als Timer-Parameter eingestellt.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
Ton	DT	Einschaltzeit
Toff	DT	Abschaltzeit
Mask	UDINT	Auswahl der einzusetzenden Mengen

Bezeichnung	Datentyp	Beschreibung
Ausgänge		
Q	BOOL	Timer-Ausgang

HINWEIS
Die Angabe der Variable **Mask** ist optional.
Wenn der Wert der Variable **Maske** nicht angegeben ist, wird für den Block standardmäßig eine Maske = 63 (0b111111) verwendet.
Wobei:
Mask = 63 (0b111111):
- Bit 0: Sekunden
- Bit 1: Minuten
- Bit 2: Stunden
- Bit 3: Tage
- Bit 4: Monate
- Bit 5: Jahre

```
FUNCTION_BLOCK CLOCK_timer //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
       I_in : BOOL;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
        CLOCK_1: SYS.CLOCK;
    END_VAR

    //Codebereich


    CLOCK_1(Ton := DT#2023-09-28-7:20:55, Toff := DT#2023-09-28-12:30:59);

    Q_out := CLOCK_1.Q;

END_FUNCTION_BLOCK
```

12.6.2.5 Wochenschaltuhr (SYS.CLOCKWEEK)

Die Wochenschaltuhr (SYS.CLOCKWEEK) dient zur Erzeugung eines Impulses zum Einschalten des Ausgangs Q gemäß der Echtzeituhr unter Berücksichtigung der Wochentage. Als Timerparameter werden die Einschaltzeit Ton und Ausschaltzeit Toff des Ausgangs Q sowie die Betriebswochentage eingestellt.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
Ton	DT	Einschaltzeit
Toff	DT	Abschaltzeit
DayOfWeekMask	UDINT	Auswählen der zu verwendenden Tage
DateTimeMask	UDINT	Auswahl der einzusetzenden Mengen
Ausgänge		
Q	BOOL	Timer-Ausgang

HINWEIS
Die Angabe der Variablen DayOfWeekMask und DateTimeMask ist optional.
Wenn der Wert der Variable DayOfWeekMask nicht angegeben ist, wird für den Baustein standardmäßig DayOfWeekMask = 127 (0b1111111) verwendet.
Wenn der Wert der DateTimeMask-Variable nicht angegeben ist, wird für den Block standardmäßig DateTimeMask = 63 (0b111111) verwendet.
Wobei:
DayOfWeekMask = 127 (0b1111111):
- Bit 0: Montag
- Bit 1: Dienstag
- Bit 2: Mittwoch
- Bit 3: Donnerstag
- Bit 4: Freitag
- Bit 5: Samstag
- Bit 6: Sonntag

DateTimeMask = 63 (0b111111):
- Bit 0: Sekunden
- Bit 1: Minuten
- Bit 2: Stunden
- Bit 3: Tage
- Bit 4: Monate
- Bit 5: Jahre

```
FUNCTION_BLOCK CLOCKWEEK_timer //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
       I_in : BOOL;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
        CLOCKWEEK_1: SYS.CLOCKWEEK;
    END_VAR

    //Codebereich
```

```
    CLOCKWEEK_1(Ton := DT#2023-09-28-7:20:55, Toff := DT#2023-09-28-12:30:59);
    Q_out := CLOCKWEEK_1.Q;
END_FUNCTION_BLOCK
```

12.6.3 Generatoren

– *Impulsgenerator (SYS.BLINK)*

12.6.3.1 Impulsgenerator (SYS.BLINK)

Der Impulsgenerator (SYS.BLINK) dient zur Bildung von Rechteckimpulsen. Am Ausgang Q des Generators werden Impulse mit festgelegten Parametern der Dauer des Ein- (Th – Signal mit hohem Pegel) und Aus-Zustands (Tl – Signal mit niedrigem Pegel) für die Dauer des Steuersignals am Eingang I (Signal mit hohem Pegel) gebildet.

Bezeichung	Datentyp	Beschreibung
Eingänge		
I	BOOL	Arbeitserlaubnis
Th	TIME	Dauer eines HIGH-Pegelsignals
Tl	TIME	Dauer eines LOW-Pegelsignals
Ausgänge		
Q	BOOL	Generatorausgang

```
FUNCTION_BLOCK BLINK_generator //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
       I_in : BOOL := FALSE;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
        BLINK_1: SYS.BLINK;
    END_VAR

    //Codebereich

    BLINK_1(I := I_in, Th := T#1000ms, Tl := T#1000ms);
    //wobei ms Millisekunden, s Sekunden, m Minuten, h Stunden und d Tage sind.

    Q_out := BLINK_1.Q;

END_FUNCTION_BLOCK
FUNCTION_BLOCK BLINK_generator //Name des Funktionsbereich
```

```
    VAR_INPUT //Deklaration von Eingabevariablen
        I_in : BOOL := FALSE;
        Th_in : UDINT := 5000;//Millisekunden
        Tl_in : UDINT := 5000;//Millisekunden
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR
        BLINK_1: SYS.BLINK;
        Th_time: TIME;
        Tl_time: TIME;
    END_VAR

    //Codebereich

    Th_time := UDINT_TO_TIME(Th_in);
    Tl_time := UDINT_TO_TIME(Tl_in);
    BLINK_1(I := I_in, Th := Th_time, Tl := Tl_time, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.4 Zaehler

- *Schwellwertzähler mit Auto-Reset (SYS.CT)*
- *Universalzähler (SYS.CTN)*
- *Schwellwertzähler (SYS.CTU)*

12.6.4.1 Schwellwertzaehler mit Auto-Reset (SYS.CT)

Der Schwellenzähler mit Selbstrückstellung (SYS.CT) dient zum Zählen einer bestimmten Anzahl von Impulsen N (Eingang N ist die Impulsanzahleinstellung). Am Ausgang Q des Zählers erscheint ein Impuls des Signals mit HIGH-Pegel mit der Dauer des Gerätearbeitszyklus (Zykluszeit), wenn die Anzahl der am Eingang C ankommenden Impulse den eingestellten Wert N erreicht.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
C	BOOL	Zählereingang
N	UDINT	Setpoint
Ausgänge		
Q	BOOL	Signalisierung des erreichten Sollwertes (Dauer ein Zyklus)

```
FUNCTION_BLOCK CT_counter //Name des Funktionsbaustein

    VAR_INPUT //Deklaration von Eingabevariablen
        C_in : BOOL;
```

```
        N_in : UDINT := 10;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR //Deklaration von Eingabevariablen
        CT_1: SYS.CT;
    END_VAR

    //Codebereich

    CT_1(C := C_in, N := N_in, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.4.2 Universalzaehler (SYS.CTN)

Der Universalzähler (SYS.CTN) dient zum direkten und indirekten Zählen. Die Operation „direktes Zählen“ wird durch die steigende Flanke des Impulses am Direktzähleingang U ausgeführt, wodurch der Wert des Ausgangssignals Q erhöht wird. Am Eingang D ankommende Impulse („Abnahmezählen“) verringern den Wert des Ausgangs Q. Wenn am Eingang R ein logisches „1“-Signal ankommt, wird der Ausgang des Zählers Q auf den Wert des Eingangs N gesetzt.

Bezeichnung	Datentyp	Beschreibung
Eingänge		
U	BOOL	Direktes Zählen
D	BOOL	Countdown
R	BOOL	Setzt den Ausgangszustand auf den Wert N zurück
N	UDINT	Setpoint
Ausgänge		
Q	UDINT	Kumulativer Wert der Impulse

```
FUNCTION_BLOCK CTN_counter //Name des Funktionsbausteins

    VAR_INPUT //Deklaration von Eingabevariablen
        U_in : BOOL;
        D_in : BOOL;
        R_in : BOOL;
        N_in : UDINT := 10;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : UDINT;
    END_VAR
```

```
    VAR //Deklaration lokaler Variablen
        CTN_1: SYS.CTN;
    END_VAR

    //Codebereich

    CTN_1(U := U_in, D := D_in, R := R_in, N := N_in, Q => Q_out);

END_FUNCTION_BLOCK
```

12.6.4.3 Schwellwertzaehler (SYS.CTU)

Mit dem Schwellwertzähler (SYS.CTU) wird die Anzahl der am Eingang C ankommenden Impulse gezählt. Ein Impuls des Signalpegels HIGH erscheint am Ausgang des Zählers Q, wenn die Anzahl der am Eingang ankommenden Impulse den am Eingang N eingestellten Wert erreicht (N ist der Sollwert).

Bezeichnung	Datentyp	Beschreibung
Eingänge		
U	BOOL	Direktes Zählen
R	BOOL	Setzt den Zählerstand auf 0 zurück
N	UDINT	Sollwert
Ausgänge		
Q	BOOL	Signalisierung Sollwert erreicht

```
FUNCTION_BLOCK CTU_counter //Functionsbausteinname

    VAR_INPUT //Deklaration von Eingabevariablen
        C_in : BOOL;
        R_in : BOOL;
        N_in : UDINT := 10;
    END_VAR

    VAR_OUTPUT //Deklaration von Ausgabevariablen
        Q_out : BOOL;
    END_VAR

    VAR //Deklaration lokaler Variablen
        CTU_1: SYS.CTU;
    END_VAR

    //Codebereich

    CTU_1(C := C_in, R := R_in, N := N_in, Q => Q_out);

END_FUNCTION_BLOCK
```