



Modul für Erweiterte Logiken - Handbuch

Version 1.0

REV01-20240506

ALLGEMEINE INFORMATIONEN

DIVUS
Pillhof
I-39057 Eppan (BZ)

GmbH
51

Betriebsanleitungen, Handbücher und Software sind urheberrechtlich geschützt. Alle Rechte bleiben vorbehalten. Das Kopieren, Vervielfältigen, Übersetzen, Umsetzen im Ganzen oder in Teilen ist nicht gestattet. Eine Ausnahme gilt für die Anfertigung einer Sicherungskopie der Software für den eigenen Gebrauch.




Änderungen des Handbuchs behalten wir uns ohne Vorankündigung vor. Die Fehlerfreiheit und Richtigkeit der in diesem Dokument und auf den mitgelieferten Speichermedien enthaltenen Daten können wir nicht garantieren. Anregungen zu Verbesserungen sowie Hinweise auf Fehler sind uns jederzeit willkommen. Die Vereinbarungen gelten auch für die speziellen Anhänge zu diesem Handbuch.

Die Bezeichnungen in diesem Dokument können Marken sein, deren Benutzung durch Dritte für eigene Zwecke die Rechte der Inhaber verletzen können.

Benutzerhinweise: Bitte lesen Sie das Handbuch vor dem ersten Einsatz und bewahren Sie es zur späteren Verwendung sorgfältig auf.

Zielgruppe: Das Handbuch ist für Anwender mit Vorkenntnissen in der PC- und Automatisierungstechnik geschrieben.

DARSTELLUNGSKONVENTIONEN

[TASTE]	Tasteneingaben des Benutzers werden in eckigen Klammern dargestellt, z.B. [STRG] oder [ENTF]
COURIER	Bildschirm Ausgaben werden in der Schriftart Courier beschrieben, z.B. C:\>
COURIER FETT	Tastatureingaben durch den Benutzer sind in Schriftart Courier fett beschrieben, z.B. C:\>DIR
„...“	Namen von auszuwählenden Schaltflächen, Menüs oder anderen Bildelementen werden in „Gänsefüßchen“ wiedergegeben.
PIKTOGRAMME	Im Handbuch sind folgende Piktogramme zur Kennzeichnung bestimmter Textabschnitte verwendet:
	Achtung! Möglicherweise gefährliche Situation. Sachschäden können die Folge sein.
	Notizen Tipps und ergänzende Hinweise
	Neu Kennzeichnet Änderungen und neue Features

INHALTSVERZEICHNIS

ALLGEMEINE INFORMATIONEN	2
DARSTELLUNGSKONVENTIONEN	2
INHALTSVERZEICHNIS	3
1 ALLGEMEINER ÜBERBLICK.....	8
1.1 EINLEITUNG.....	8
2 BENUTZEROBERFLÄCHE.....	9
2.1 ALLGEMEINES LAYOUT	9
2.2 NAVIGATIONSMENU	9
2.2.1 LOGIKEINHEITEN	9
2.2.2 BIBLIOTHEK.....	9
2.2.3 ZEICHENUTENSILIEN	9
2.3 SYMBOLLEISTE.....	10
2.4 DETAILBEREICH	10
2.5 ARBEITSBEREICH.....	11
2.6 BENACHRICHTIGUNGSBEREICH	11
3 LOGIKEINHEITEN	13
3.1 EINLEITUNG.....	13
3.2 EINE NEUE LOGIKEINHEIT ERSTELLEN	13
3.3 EINEN NEUEN TASK ERSTELLEN.....	13
3.4 ENTFERNEN ODER DEAKTIVIEREN EINES TASKS	14
3.5 HINZUFÜGEN VON BLÖCKEN ZU EINEN TASK.....	14
3.6 AUSWAHL EINES BLOCKS ODER MEHRERER BLÖCKE.....	15
3.7 ENTFERNUNG EINES ODER MEHRERER BLÖCKE	16
3.8 EIN- UND AUSGABESKNOTEN	16
3.8.1 LOGISCHE BLÖCKE.....	17

3.9	VERBINDUNG VON BLÖCKEN	18
3.10	AUSFÜHRUNGSREIHENFOLGE	19
3.10.1	TASKREIHENFOLGE	19
3.10.2	REIHENFOLGE DER BLÖCKE	20
3.11	ÜBERGABE VON WERTEN ZWISCHEN TASKS.....	21
3.12	DATENTYPEN.....	22
3.13	SIMULATION	22
4	LOGIKEN	24
4.1	EINLEITUNG.....	24
4.2	LOGIKBLÖCKE	24
4.2.1	LAYOUT	24
4.2.2	EINGABEKNOTEN	24
4.2.3	AUGABEKNOTEN.....	25
4.2.4	KNOTEN HINZUFÜGEN UND ENTFERNEN	25
4.3	KOMBINATORISCHE LOGIK.....	25
4.3.1	UND.....	25
4.3.2	ODER	26
4.3.3	XOR	27
4.3.4	NOT	27
4.4	SZENARIEN UND SEQUENZEN.....	28
4.4.1	SEQUENZER	28
4.4.2	BINÄRES SZENARIO	29
4.4.3	NUMERISCHES SZENARIO.....	30
4.5	GATTER.....	31
4.5.1	BINÄRER WAHLSCHALTER	32
4.5.2	NUMERISCHER WAHLSCHALTER	32
4.5.3	BINÄR-ENCODER.....	33

4.5.4	DECODER.....	34
4.5.5	T-FLIPFLOP	35
4.5.6	RS-FLIPFLOP	36
4.5.7	D-FLIPFLOP.....	37
4.5.8	BINÄRE D-VERRIEGELUNG	37
4.5.9	NUMERISCHE D-VERRIEGELUNG.....	38
4.6	VERGLEICHE.....	39
4.7	OPERATIONEN	40
4.7.1	MATHEMATISCHE OPERATOREN.....	40
4.7.1.1	BEREICH.....	41
4.8	ZÄHLER.....	42
4.8.1	ZÄHLER, COUNTDOWN, VOR-/RÜCKWERTSZÄHLER.....	42
4.9	TIMER & ZEITPLÄNE	43
4.9.1	TIMER	43
4.9.2	AUSLÖSER (TRIGGER).....	44
4.10	VARIABLEN.....	45
4.10.1	VORWORT.....	45
4.10.2	BINÄRE VARIABLEN.....	45
4.10.3	NUMERISCHE VARIABLEN.....	46
4.11	KONSTANTEN	46
5	AUSLÖSER	47
5.1	EINLEITUNG.....	47
5.2	EINGANGS- UND AUSGANGSTRIGGER.....	47
5.3	AUSGABESTRATEGIE	48
5.4	TRIGGERVERBINDUNGEN	49
5.5	AUSLÖSER UND KOMPLEXE OBJEKTE	50
6	VERBINDUNGEN.....	52

6.1	EINFÜHRUNG	52
6.2	VERBINDUNGSARTEN	52
7	SIMULATION	53
7.1	EINLEITUNG.....	53
7.2	SIMULATIONSARTEN	53
7.3	GRAFISCHE SIMULATIONSUMGEBUNG.....	53
7.4	MANUELLE WERTEINGABE.....	54
7.5	SIMULATION ANHALTEN.....	55
8	ZEICHENUTENSILIEN	56
8.1	EINLEITUNG.....	56
8.2	BESCHRIFTUNGEN.....	56
8.3	RECHTECKIGE BEREICHE	57
9	ANHANG.....	58
9.1	GLOSSAR	58
9.2	NOTIZEN.....	59

1 Allgemeiner Überblick

1.1 EINLEITUNG

Das Modul für Erweiterte Logiken ermöglicht es, innerhalb vom DIVUS KNX SERVER/DIVUS D+ komplexe logische Netzwerke durch einen integrierten Grafikeditor und eine umfangreiche Bibliothek von logischen Blöcken zu realisieren, mit denen Objekte aller Technologien, die vom Server verwaltet werden, untereinander verbunden werden können.

Das Modul ermöglicht es, ein oder mehrere **Logikeinheiten** zu erstellen, die wiederum aus einem oder mehreren **Tasks** bestehen können; Jeder Task ist ein logisches Netzwerk, das über einen Grafikeditor konfigurierbar ist, die Daten mit den anderen Tasks durch geeignete Objekte vom Typ Variablen austauschen können, wie weiter unten näher spezifiziert.

Die Tasks können aktiviert oder deaktiviert werden, und basierend auf dieser Aktivierung wird beim Starten eines Tasks (simuliert oder "real"), ein Skript in der Sprache LUA generiert, das im Hintergrund ausgeführt wird und durch den Austausch von Ein- und Ausgabedaten in einer Schleife fortfährt, was vom Konzept her ähnlich zu dem ist, was in den PLCs geschieht.

Logikeinheiten können parallel in Betrieb genommen werden - sie sind voneinander unabhängige Prozesse; umgekehrt sind die *Tasks* "Teile" einer *Logikeinheit* und fließen während des Kompilierens in das LUA-Skript ein. Daher werden alle aktivierten Tasks immer vom ersten bis zum letzten ausgeführt, gemäß der auf der Programmkonfigurationsseite festgelegten Reihenfolge.

Im Gegensatz zu den anderen Funktionen von KNX SERVER/D+, die "beim Ereignis" verarbeitet werden (d.h. wenn an einem der Eingänge ein Statuswechsel erfolgt), laufen die LUA-Logikeinheiten ständig und halten die Ausgangs-Objekte in ihrem gewünschten Zustand, falls sie auf Grund eines äußeren Ereignisses ihren Zustand ändern sollten.

Der LUA-Code wird zyklisch im Hintergrund ausgeführt und verarbeitet bei jedem Ausführungszyklus die Logik auf der Grundlage des zu Beginn des Zyklus erkannten Zustands der Eingangsobjekte; die Ausgangsobjekte werden entsprechend den Ergebniswert erhalten, wenn ihr Zustand nicht bereits mit dem von der Logik vorhergesagten kohärent ist.

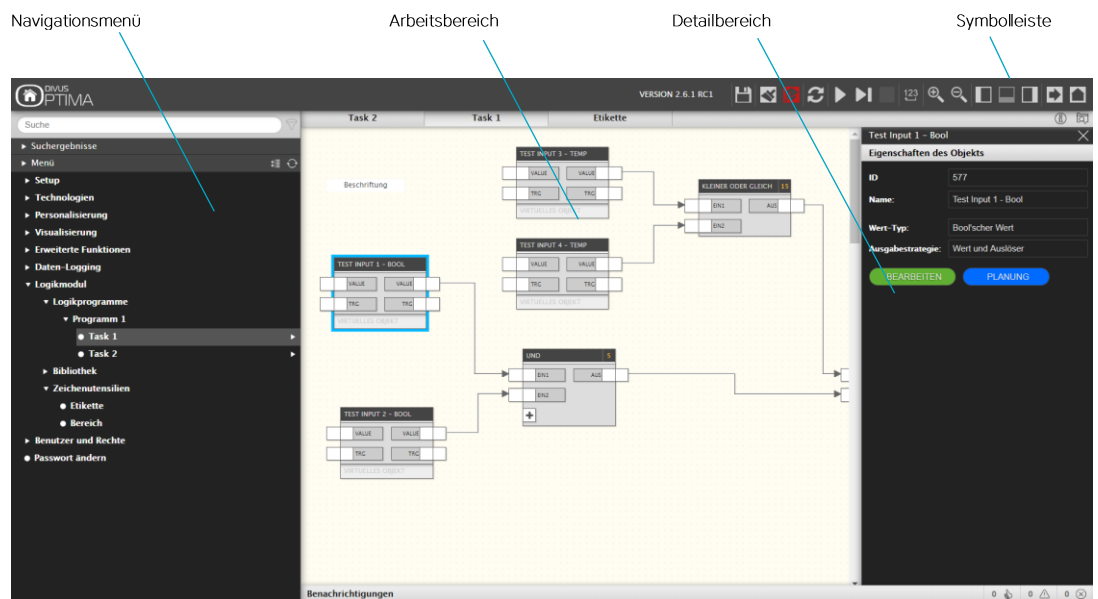
Unter normalen Bedingungen werden daher die Ausgangsobjekte mit der Logik in Übereinstimmung gehalten, auch wenn sich ihr Zustand aus externen Gründen ändert, indem sofort ein entsprechender Befehl gesendet wird (im nächsten LUA-Ausführungszyklus); dieses Verhalten unterscheidet sich stark von anderen KNX SERVER/D+ Ereignislogiken und sollte bei der Erstellung eines Logiknetzwerks innerhalb des erweiterten Logikmoduls beachtet werden.

Es ist jedenfalls möglich, mit sogenannten TRIGGERS (Auslösern) zu regeln, wann die LUA-Logik die Ausgangsobjekte ändern soll, z.B. nur dann, wenn sich die Eingabeobjekte ändern (so wird vermieden, dass die Ausgänge "blockiert" werden, wenn sie sich aus irgendwelchen Gründen ändern, z.B. weil sie vom Benutzer geschaltet werden). Ein spezielles Kapitel (5) dieses Handbuchs erklärt, wie diese zu verwenden sind; eine gründliche Lektüre dieses Abschnitts wird empfohlen.

2 Benutzeroberfläche

2.1 ALLGEMEINES LAYOUT

Die folgende Abbildung zeigt die Struktur der grafischen Oberfläche des Editors, sobald das Fenster geöffnet ist:



2.2 NAVIGATIONSMENU

Das Menü enthält alles, was es zum Erstellen und Verwalten logischer Einheiten braucht. Klicken Sie dazu auf Logikmodul.

2.2.1 LOGIKEINHEITEN

Dieser Abschnitt enthält die Liste der konfigurierten logischen Einheiten. Hier können Sie Logikeinheiten erstellen, bearbeiten und löschen.

2.2.2 BIBLIOTHEK












Dieser Abschnitt enthält die Bibliothek der logischen Blöcke, die in die Tasks eingefügt werden können. Wie später genauer beschrieben, können die logischen Bibliothekseinträge durch „Drag & Drop“ in die Tasks eingefügt werden.

2.2.3 ZEICHENUTENSILIEN

Dieser Abschnitt enthält die Werkzeuge, mit denen wir unsere Tasks personalisieren können. Wie später genauer beschrieben, können die Zeichenutensilien per Drag & Drop den Task hinzugefügt werden.

2.3 SYMBOLLEISTE

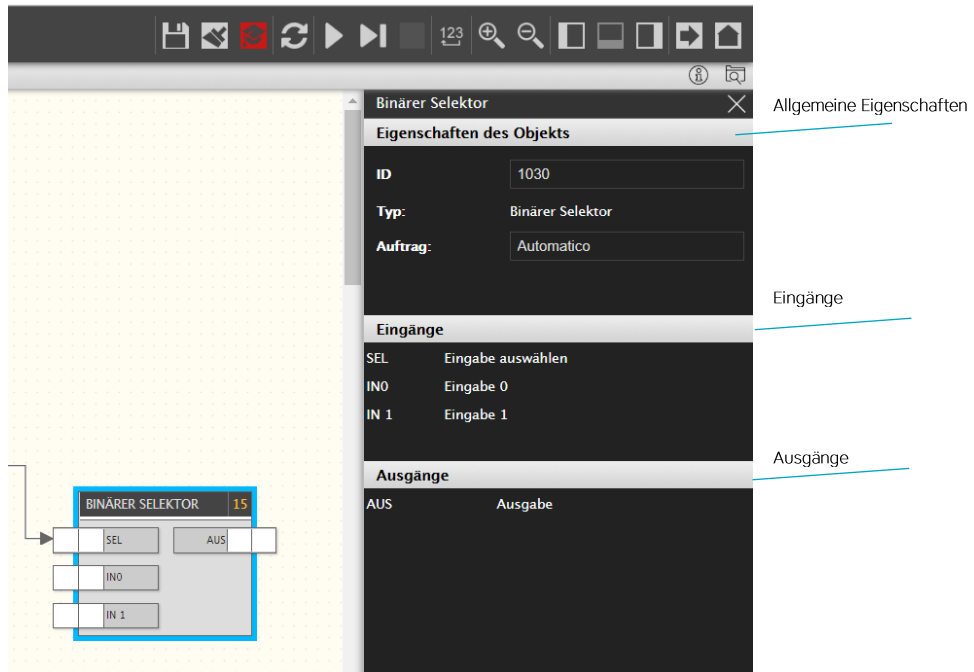
Die Symbolleiste stellt im mittleren Abschnitt die folgenden Werkzeuge zur Verfügung, die während jeder Phase der Realisierung der logischen Einheiten verfügbar sind:

	AUSFÜHRUNG / PAUSE Starten oder stoppen Sie einen Task	
	KONTINUIERLICHE SIMULATION Starten Sie die Simulation im Echtzeitmodus	
	Starten Sie die Simulation im Schritt-für-Schritt-Modus	SCHRITT-FÜR-SCHRITT-SIMULATION
	SIMULATION Anhalten Stoppen Sie die laufende Simulation	
	Ordnen Sie die Blöcke neu an durch automatisches Sortieren nach deren Position	SORTIEREN
	ZOOM Vergrößern Sie den Zoomfaktor des Arbeitsbereichs	+
	ZOOM Verringert den Zoomfaktor des Arbeitsbereichs	-
	MELDUNGEN EIN- / AUSBLENDEN Blenden Sie den Benachrichtigungsbereich unten ein oder aus	
	NAVIGATIONS MENÜ EIN- / AUSBLENDEN Zeigen oder verbergen Sie das Navigationsmenü auf der linken Seite	
	DETAILBEREICH EIN- / AUSBLENDEN Blenden Sie das rechte Fenster mit den Details ein oder aus	
	ERWEITERTE OPTIONEN Ermöglicht den Zugriff auf erweiterte Optionen, die standardmäßig versteckt sind	

2.4 DETAILBEREICH

Dieser normalerweise geschlossene Bereich kann über die entsprechende Schaltfläche in der Symbolleiste oder über ein Rechts-Klick auf das gewünschte Objekt geöffnet werden. Er enthält Details zu den im Arbeitsbereich markierten Objekten und ermöglicht das Ändern der Eigenschaften und Optionen.

Je nach Art des ausgewählten Objekts können die Informationen in mehrere Abschnitte unterteilt werden, wie in der folgenden Abbildung dargestellt:

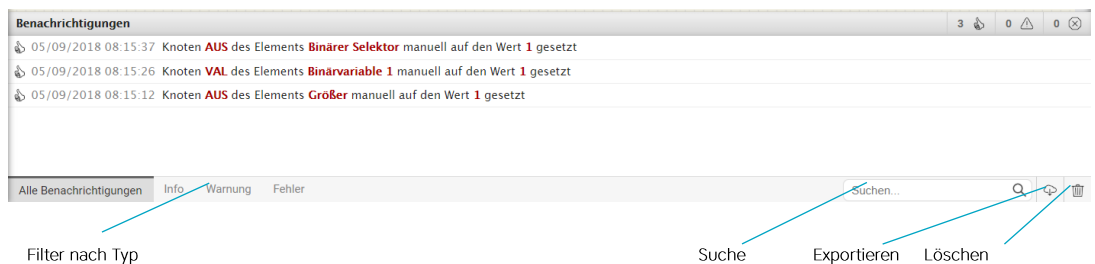


2.5 ARBEITSBEREICH

Der zentrale Teil des Fensters ist dem Arbeitsbereich gewidmet, in dem die Logiken wie unten beschrieben aufgebaut werden. Um den Nutzraum zu erweitern, empfiehlt es sich, insbesondere bei der Bearbeitung der Tasks die Seitenfenster und den unteren Benachrichtigungsbereich zu schließen.

2.6 BENACHRICHTIGUNGSBEREICH

Der untere Teil des Fensters enthält die Nachrichten, die vom Editor während der Realisierung der Logikeinheiten und vor allem während der Simulation erzeugt wurden (wie unten genauer beschrieben).



Die vom Editor generierten Nachrichten können je nach Schweregrad und Typ unterschiedliche Typen aufweisen:

- **Fehler:** Berichte zu Vorgängen oder Bedingungen, die einen Fehler verursachen und normalerweise eine Änderung oder Verifizierung durch den Benutzer erfordern
- **Warnung:** Warnungen vor anomalen Bedingungen, die jedoch nicht unbedingt einen Fehler oder eine zu ändernde Situation darstellen
- **Info:** "normale" Informationsnachrichten, die vom Redakteur durchgeführte Operationen melden, die dem Benutzer berichtenswert sind
- **Debug:** detaillierte Meldungen der von der Simulation durchgeführten Operationen (nur im "Schritt für Schritt" -Modus verfügbar, wie unten beschrieben)

Die verschiedenen Typen unterscheiden sich durch eine Farbe, die auf der Seite jeder Nachricht hervorgehoben ist, zusammen mit dem Datum und der Uhrzeit, zu der die Nachricht generiert wurde. Die Titelleiste des Benachrichtigungsbereichs enthält auf der rechten Seite eine Zusammenfassung der Anzahl der Nachrichten der verschiedenen Typen, die auch dann sichtbar sind, wenn der Benachrichtigungsbereich geschlossen ist.

Die folgenden Befehle stehen am unteren Rand des Benachrichtigungsbereichs zur Verfügung:

- **Nach Nachrichtentyp filtern:** Durch Auswählen eines der verfügbaren Einträge ist es möglich, die Nachrichten auf dem Bildschirm nach dem entsprechenden Typ zu filtern
- **Nach Nachrichten suchen:** Damit können Sie Nachrichten basierend auf einem oder mehreren Schlüsselwörtern filtern
- **Export:** ermöglicht das Exportieren der Nachrichtenhistorie (einschließlich derjenigen, die sich auf vorherige Arbeitssitzungen beziehen) im CSV-Format, das über externe Software (z. B. Tabellenkalkulationen) abgerufen werden kann.
- **Leer:** erlaubt das Löschen von Nachrichten auf dem Bildschirm (Nachrichten bleiben trotzdem im Editor archiviert und können über die entsprechende Schaltfläche für eine "Offline"-Besichtigung exportiert werden)

3 Logikeinheiten

3.1 EINLEITUNG

Die Logikeinheiten sind eingerichtet, um ein oder mehrere logische Netzwerke (Tasks) auszuführen, die typischerweise eine oder mehrere Informationen vom Bus empfangen, diese durch Logikblöcke verarbeiten und die Ergebnisse in Form von Befehlen auf den Bus senden. Jede Logikeinheit kann bis zu 16 Tasks enthalten.

Mit dem Editor können Sie Logikeinheiten konfigurieren, indem Sie Blöcke und logische Funktionen per Drag & Drop und einfache grafische Werkzeuge ohne besondere Programmierkenntnisse verbinden. Wie weiter unten erläutert wird, ermöglicht der Editor auch, das Verhalten logischer Einheiten zu simulieren.

3.2 EINE NEUE LOGIKEINHEIT ERSTELLEN

Um eine neue Logikeinheit zu erstellen, klicken Sie im Bereich "Logikmodul" des Hauptmenüs auf "Logikeinheiten" und drücken Sie die entsprechende "+" -Taste: eine neue leere Logikeinheit namens "Logikeinheit 1" wird angelegt.

3.3 EINEN NEUEN TASK ERSTELLEN

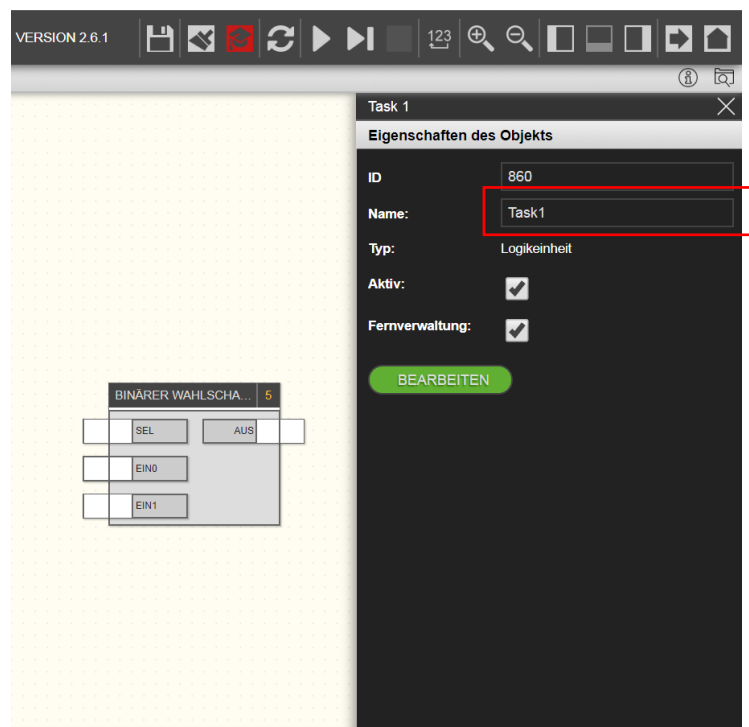
Um einen neuen Task zu erstellen, wählen Sie zuerst im Bereich "Logikeinheiten" des Hauptmenüs die Logikeinheit aus, auf der Sie arbeiten möchten, und drücken Sie die entsprechende "+" -Taste: ein neuer leerer Task namens "Task 1" wird erstellt.

Um den neuen Task zu öffnen, klicken Sie einfach darauf: Im Arbeitsbereich wird ein leeres Fenster angezeigt, auf dem, wie unten beschrieben, mit dem Aufbau der Logik begonnen werden kann.

Um den Namen des Tasks zu ändern, öffnen Sie den Detailbereich und geben Sie den neuen Namen in das entsprechende Textfeld ein, wie in der folgenden Abbildung dargestellt. Der Name darf keine Sonderzeichen enthalten und darf maximal 16 Zeichen lang sein.



Zu bemerken: welcher logischen Einheit sie einen neuen Task hinzufügen, spielt nur in dem Sinn eine Rolle, dass unterschiedliche Tasks eine Teilverarbeitung einer komplexeren Logik darstellen könnten, wo gewisse Tasks das Zwischenergebnis der vorher ausgeführten weiterverarbeiten. Dann spielen sei es die Logikeinheit als auch die Reihenfolge der Tasks innerhalb der Einheit eine Rolle. Außerdem ist zu bedenken, dass alle Tasks einer logischen Einheit gemeinsam ausgeführt oder angehalten werden. Bei logisch unabhängigen Tasks – und also im Großteil der Fälle – spielt die Zugehörigkeit zu Logischen Einheiten hingegen keine Rolle.



3.4 ENTFERNEN ODER DEAKTIVIEREN EINES TASKS

Um einen bestehenden Task zu entfernen, drücken Sie einfach die entsprechende "X" -Taste im Logikeinheiten-Fenster; Sobald das Löschen bestätigt ist, wird der Task und damit alle darin enthaltenen logischen Funktionen eliminiert. Dieser Vorgang kann nicht abgebrochen werden.

Wenn Sie nicht möchten, dass ein Task in die Logikeinheit eingefügt wird, weil er z. B. noch unvollständig ist, können Sie ihn deaktivieren, indem Sie im Detailbereich den entsprechenden Eintrag "AKTIVIEREN" abwählen; Deaktivierte Tasks werden mit einem halbtransparenten Effekt gekennzeichnet.

3.5 HINZUFÜGEN VON BLÖCKEN ZU EINEM TASK

Tasks bilden aus der Verbindung mehrerer Blöcke ein logisches Netzwerk. Die Blöcke können klassische OPTIMA-Objekte sein, insbesondere KNX-Objekte, oder sie können vom logischen Typ sein; erstere sind nützlich, um Informationen auf dem Gebäudeautomationsbus zu lesen und/oder zu schreiben, letztere erlauben, diese Informationen zu verarbeiten und zu kombinieren.

Um einen klassischen Block zu einem Task hinzuzufügen, müssen Sie ihn zuerst im Abschnitt "ETS Projekt" identifizieren, der sich im Abschnitt "KNX" unterhalb des Punktes "Technologien" des Hauptmenüs befindet. Hier sind alle verfügbaren Objekte aufgelistet.

Sobald der Block identifiziert wurde, ziehen Sie ihn einfach per Drag and Drop in den Arbeitsbereich:

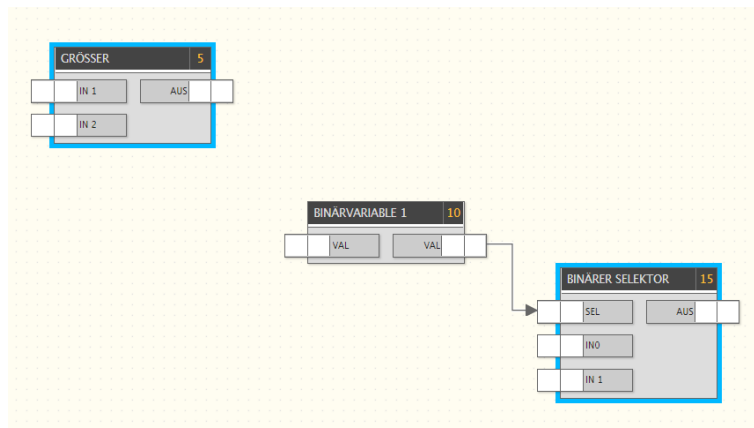
Um einen Logikblock einzufügen, müssen Sie ihn in der Bibliothek unter *Logikmodul* ebenfalls identifizieren (für eine vollständige Liste der verfügbaren logischen Blöcke siehe Kapitel 5) und dann in den Arbeitsbereich ziehen.

3.6 AUSWAHL EINES BLOCKS ODER MEHRERER BLÖCKE

Es ist möglich, einen oder mehrere Blöcke innerhalb eines Tasks auf verschiedene Arten auszuwählen:

- Klick auf den "Titel" des Blocks (Einzelauswahl)
- Durch Klicken auf den "Titel" mehrerer Blöcke bei gleichzeitigem Drücken der STRG-Taste
- Indem Sie auf einen Punkt des Arbeitsbereichs klicken und gedrückt halten, bewegen Sie den Cursor, indem Sie einen rechteckigen Auswahlbereich zeichnen

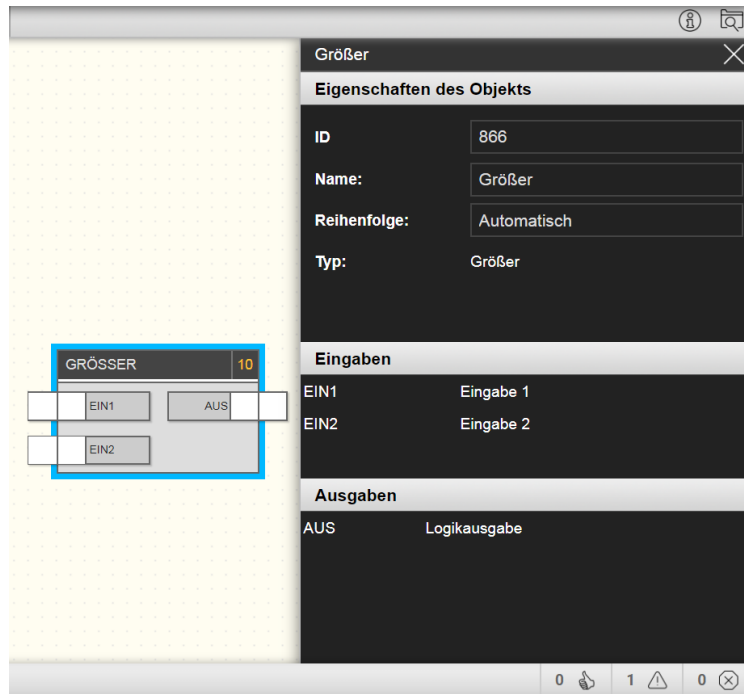
Die ausgewählten Blöcke sind mit einem blauen Rand hervorgehoben:



Die ausgewählten Blöcke können einfach durch *Drag and Drop* innerhalb des Arbeitsbereichs verschoben werden. Indem Sie einen einzelnen Block auswählen und das Detailfenster öffnen, ist es umgekehrt möglich, seine Eigenschaften, die Liste der Eingabe- und Ausgabeknoten anzuzeigen und alle Optionen zu verwalten, wie später für jeden einzelnen Typ beschrieben.



HINWEIS: Wenn Sie mehrere Blöcke gleichzeitig auswählen, ist es nicht möglich, die Details zu sehen, da sie für jeden einzelnen unterschiedlich sind.



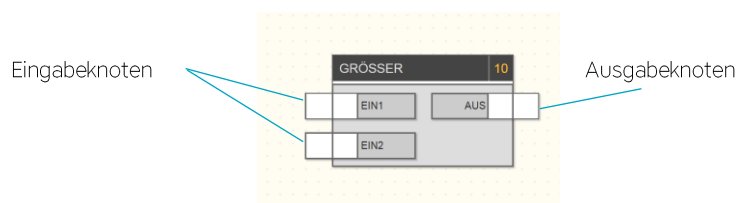
3.7 ENTFERNUNG EINES ODER MEHRERER BLÖCKE

Wählen Sie einen oder mehrere Blöcke und drücken Sie die "ENTFERNEN" -Taste auf der Tastatur

Die ausgewählten Blöcke werden nach einer Bestätigungsnachricht aus dem Fenster entfernt, ebenso wie Verbindungen mit anderen Blöcken, die im Task selbst vorhanden sind. Dieser Vorgang kann nicht abgebrochen oder rückgängig gemacht werden.

3.8 EIN- UND AUSGABESKNOTEN

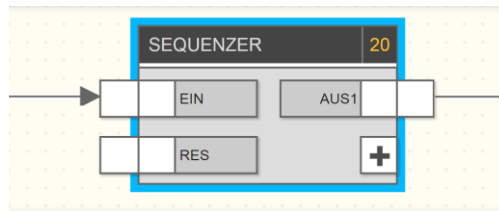
Jeder Block enthält mindestens einen Eingabe- und / oder Ausgabeknoten, wie in der folgenden Abbildung gezeigt:



Die Eingabeknoten befinden sich immer auf der linken Seite eines Blocks, während die Ausgaben auf der rechten Seite sind. Jeder Knoten ist durch eine kurze Bezeichnung gekennzeichnet (z. B. "EIN1", "EIN2" und "AUS" in der vorherigen Abbildung), die in der Eingabe / Ausgabe im Detailfenster aufgelistet ist, zusammen mit einer kurzen Beschreibung jedes Knotens.

3.8.1 LOGISCHE BLÖCKE

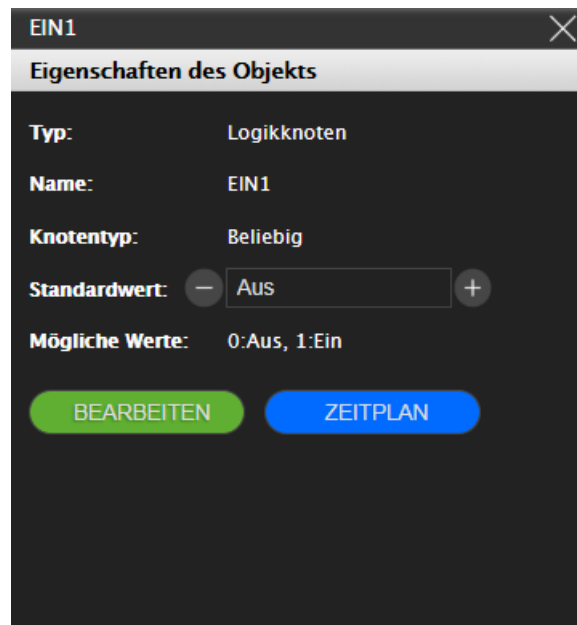
Im Fall von Logikblöcken stellen die Eingabeknoten die Eingänge für die Logikfunktion dar, die dem Block zugeordnet sind, während die Ausgabeknoten die Ausgänge sind:



In einigen Fällen, wie in diesem Beispiel, bietet der Block eine variable Anzahl von Knoten (Eingabe oder Ausgabe); Dann können Sie mit der Schaltfläche "+" dem Block bis zur maximalen Anzahl Knoten hinzufügen.

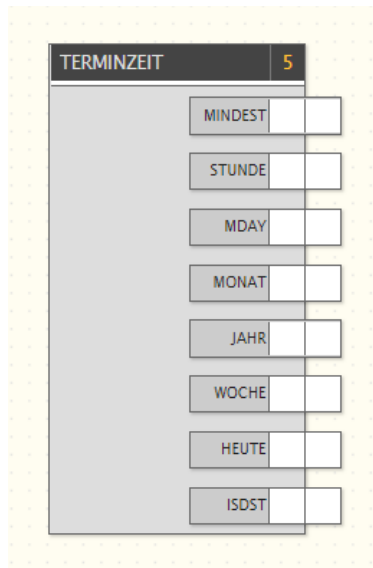
Die Logikfunktion kann nur dann korrekt ausgeführt werden, wenn die Eingabeknoten mit anderen Blöcken verbunden sind und wenn die Ausgabewerte auf die Eingabeknoten der gleichen Anzahl von empfangenden Blöcken übertragen werden.

Nicht alle Eingabeknoten sind unbedingt notwendig für die korrekte Ausführung der Logik; Wenn ein Eingangsknoten nicht verbunden ist, wird der Standardwert verwendet, der durch Auswählen des Knotens und Öffnen des entsprechenden Detailbereichs geändert werden kann, wie in der folgenden Abbildung gezeigt:



Das Detailfenster eines Knotens hebt auch die möglichen Werte hervor, die er annehmen kann; Diese Information kann vor allem für Blöcke nützlich sein, die bestimmte Kombinationen oder Einschränkungen von Werten haben.

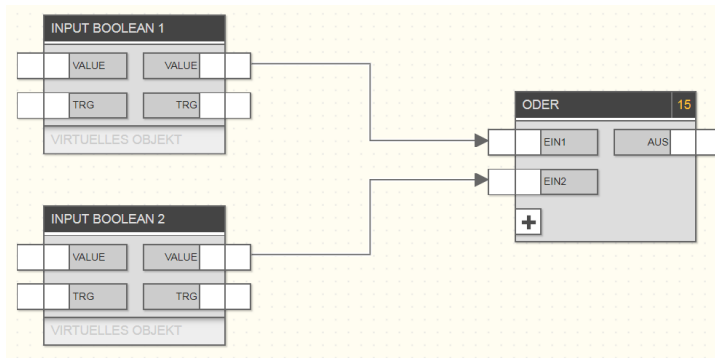
Die Logikblöcke können auch nur Ausgänge bereitstellen, wie im folgenden Beispiel (Datum / Uhrzeitblock):



In diesem Fall können sie nur als Eingabe für andere Blöcke verwendet werden, sie können jedoch nicht gesteuert werden.

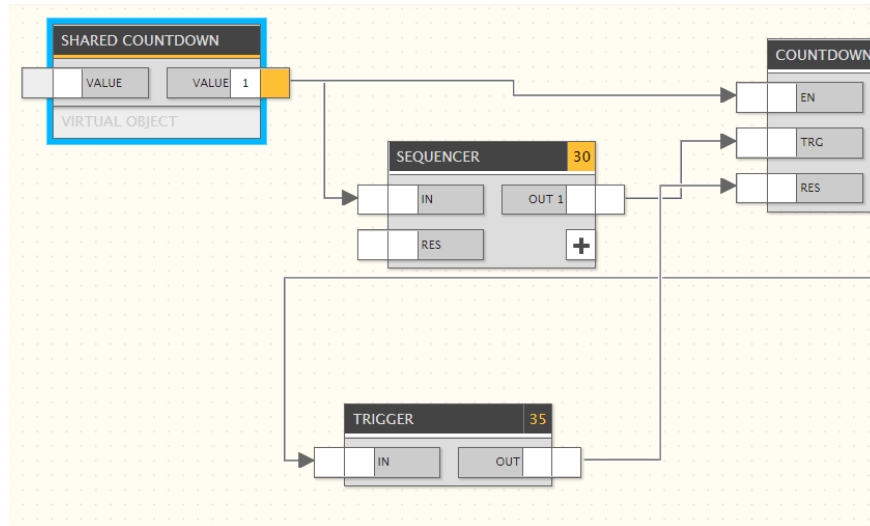
3.9 VERBINDUNG VON BLÖCKEN

Damit das Programm tatsächlich etwas ausführen kann, ist es notwendig, mindestens eine Verbindung zwischen zwei Knoten von zwei Blöcken zu erstellen, sodass der Wert des ersten (Ursprung) an den zweiten (Ziel) übergeben wird. Um zwei Knoten zu verbinden, klicken Sie einfach auf die Mitte des Quellknotens, halten die Maustaste gedrückt und lassen sie in der Mitte des Zielknotens los:



Wenn Sie mit dem Mauszeiger über einen Verbindungspfeil fahren, wird dieser rot. Indem Sie daraufklicken, wird die Verbindung ausgewählt und hellblau markiert. Um die ausgewählte Verbindung zu löschen, drücken Sie direkt die "Entfernen"-Taste auf der Tastatur.

Der Ursprung einer Verbindung muss ein Ausgabeknoten sein (rechte Seite eines Blocks), während das Ziel ein Eingabeknoten sein muss (linke Seite); Ein Ausgabeknoten kann die Quelle mehrerer Verbindungen mit unterschiedlichen Zielen sein, während ein Eingabeknoten das Ziel einer einzelnen Verbindung ist (Ausnahme: TRG-Knoten).



3.10 AUSFÜHRUNGSREIHENFOLGE

Während der Simulations- und Kompilierungsphasen, erzeugt der Editor ausgehend von den graphisch gezeichneten logischen Netzwerken eine "Liste", die zyklisch, so schnell wie möglich, von Anfang bis Ende ausgeführt wird.

3.10.1 TASKREIHENFOLGE

Bei jedem Ausführungszyklus werden die folgenden Operationen ausgeführt (die Zyklusdauer hängt von der Anzahl und Komplexität der Tasks ab):

- Lesen der Eingänge vom Bus
- Ausführung des Tasks 1
- Ausführung des Tasks 2
- ...
- Ausführung des Tasks n
- Befehle (Ausgabewerte) auf dem Bus schreiben

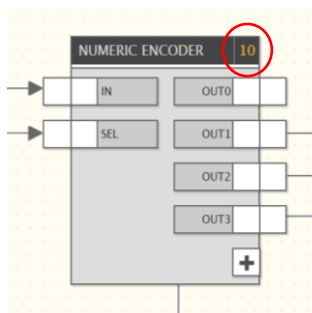
Die Reihenfolge der Tasks ist genau die im Hauptmenü gezeigte; dies bedeutet, dass jegliche Interaktionen zwischen Tasks (wie die Weitergabe von Werten über Variablen oder das Schreiben desselben Knotens eines Blocks durch mehrere Tasks) durch diese Reihenfolge beeinflusst werden und alle Aktionen, die von den Tasks gegen Ende der Warteschlange ausgeführt werden, von den vorherigen erst beim nächsten Ausführungszyklus wahrgenommen werden.



HINWEIS: Wenn ein Task deaktiviert oder angehalten ist, wird er im Ausführungszyklus "übersprungen", jegliche Interaktion mit dem Bus und / oder anderen Tasks wird in diesem Fall ausgesetzt.

3.10.2 REIHENFOLGE DER BLÖCKE

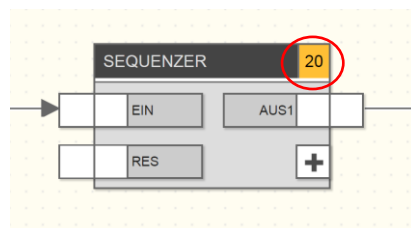
Innerhalb eines jeden Tasks haben die logischen Blöcke auch ihre eigene Reihenfolge der Ausführung: das Logikmodul verarbeitet die Funktion, die den Logikblöcken in dieser Reihenfolge zugeordnet ist. Die Reihenfolge eines Logikblocks wird oben rechts angegeben, wie in der folgenden Abbildung gezeigt:



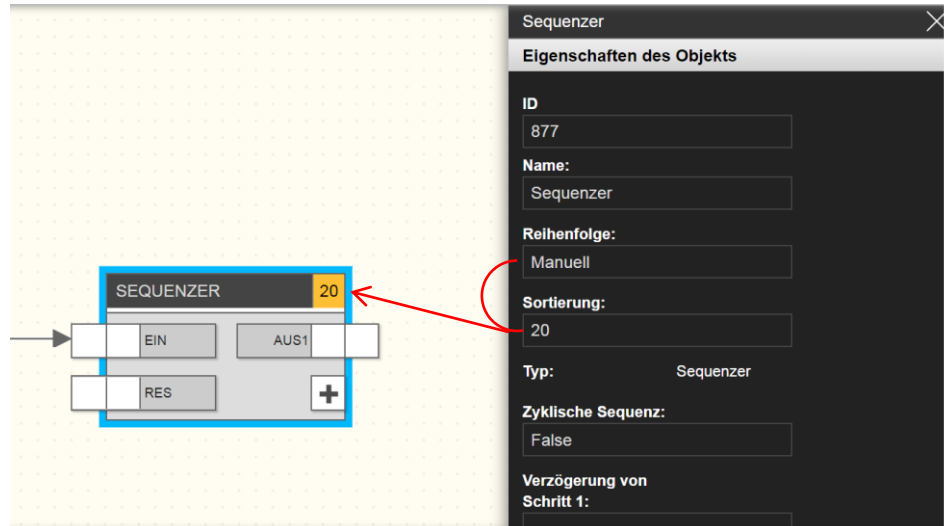
Unter normalen Bedingungen wird den Blöcken eine aufsteigende Reihenfolge zugewiesen, die der Reihenfolge entspricht, in der sie in das Programm eingefügt wurden: von oben nach unten und von links nach rechts; Es ist jedoch möglich, wie folgt eine andere Ausführungsreihenfolge zu erzwingen:

- Wählen Sie den gewünschten Block aus
- Öffnen Sie das Detailfenster
- Wählen Sie unter Reihenfolge "MANUELL"
- Achten Sie darauf, eine nicht schon verwendete Nummer einzugeben

Die Blöcke mit manueller Sortierung sind wie folgt markiert:



Die folgende Abbildung zeigt ein Beispiel für ein logisches Netzwerk mit manueller Reihenfolge und zeigt, wie diese geändert werden kann:



Die Blöcke haben nicht wirklich eine Ausführungsreihenfolge, da sie keine Verarbeitung durch die Logikeinheit darstellen, sondern nur Lese- und Schreibpunkte vom/zum Bus; Wie bereits oben erwähnt, werden die Zustände der Eingabeknoten aller Blöcke (aller aktiven Tasks) zu Beginn jedes Ausführungszyklus gelesen, und die Befehle/Werte an den Ausgabeknoten aller Blöcke (von allen aktive Tasks) werden am Ende des Ausführungszyklus auf den Bus gesendet, unabhängig von der Position der Blöcke in den Tasks und der Reihenfolge der Tasks selbst.

3.11 ÜBERGABE VON WERTEN ZWISCHEN TASKS

Obwohl jeder Task ein eigenständiges logisches Netzwerk definiert, ist es möglich, Werte zwischen verschiedenen Tasks unter Verwendung bestimmter logischer Blöcke, die *Variablen* genannt werden, zu übergeben. Um eine neue Variable anzulegen, gehen Sie folgendermaßen vor:

- Öffnen Sie im "Logikmodul"-Bereich des Hauptmenüs die *Bibliothek*
- Markieren Sie den Unterpunkt "*Binäre Variablen*" (wenn Sie eine ON / OFF-Variable erstellen möchten) oder "*Numerische Variablen*"
- Drücken Sie die blaue "+"-Taste und warten Sie, bis die neue Variable in die Liste eingefügt wurde
- Wählen Sie die neue Variable und ziehen Sie sie in den ersten Task

Es ist möglich, der Variablen über das Detailfenster einen Namen zuzuweisen, um sie innerhalb der Tasks, in denen sie verwendet wird, leichter zu identifizieren.

Wenn der Wert eines Ausgangsknotens eines Logikblocks der Variable zugewiesen werden soll, ist es ausreichend, diesen mit dem Eingangsknoten (linke Seite) der Variable zu verbinden; Um diesen Wert in anderen Tasks zu verwenden, verbinden Sie umgekehrt den Ausgangsknoten (rechte Seite) mit dem Eingangsknoten eines anderen Blocks.

3.12 DATENTYPEN

Die Eingabe- und Ausgabeknoten der Blöcke können zwei Arten von Daten haben:

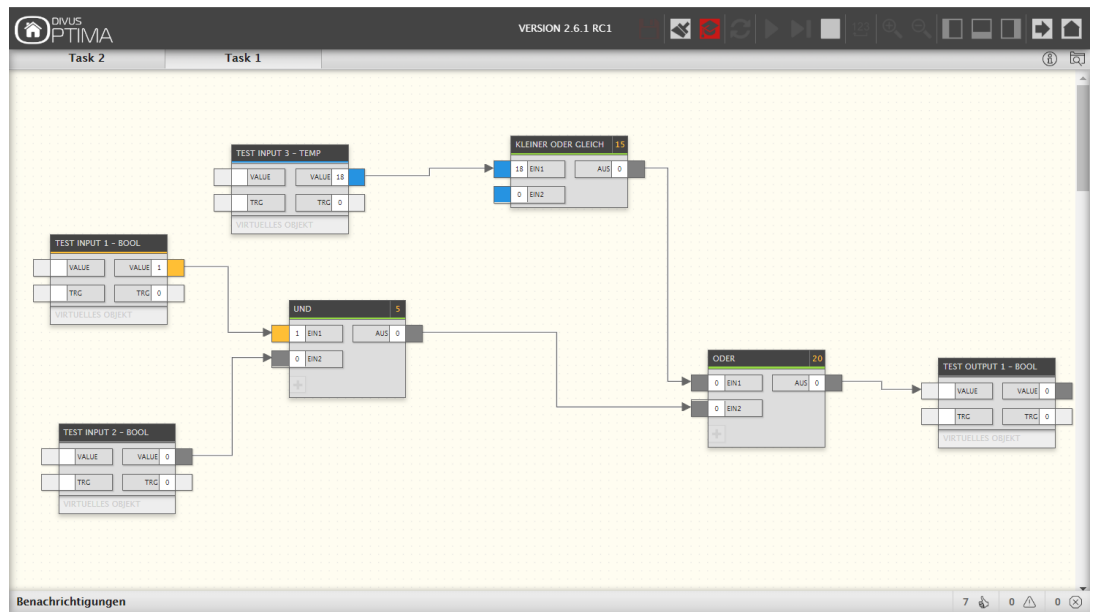
- *Binär*: Nur die Werte 1 (ON) und 0 (OFF) sind erlaubt
- *Numerisch*: Jeder numerische Wert ist erlaubt, mit bestimmten Einschränkungen je nach Block

Diese beiden Datentypen sind nicht kompatibel, so dass der Editor die Verbindung von binären Knoten mit numerischen Knoten und umgekehrt verhindert: sobald Sie ein Drag & Drop beginnen, werden inkompatible Knoten halb-transparent und akzeptieren das Auslassen zum Erstellen der Verbindung nicht.

3.13 SIMULATION

Bevor man die Tasks umsetzt, ist es ratsam, sie im Editor über die „Simulation“ durch manuell eingegebene Werte zu testen und das Verhalten der logischen Netzwerke zu überprüfen - entweder in kontinuierlicher Weise (wiederholte Ausführung der Logik in Echtzeit) oder Schritt für Schritt (d.h. indem jeweils ein Berechnungszyklus ausgeführt wird).

Weitere Informationen zur Simulation finden Sie in Kapitel 6.



4 Logiken

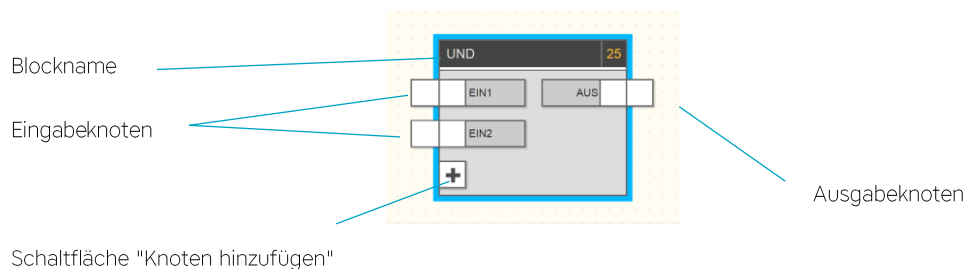
4.1 EINLEITUNG

Logikblöcke ermöglichen, dass Operationen auf einem oder mehreren Eingabewerten ausgeführt werden und geben einen oder mehrere Ausgabewerte zurück, die mit anderen Logikblöcken verbunden werden können.

4.2 LOGIKBLÖCKE

4.2.1 LAYOUT

Wie bereits erwähnt, werden die Logikblöcke grafisch wie im folgenden Beispiel dargestellt:



4.2.2 EINGABEKNOTEN

Die Eingabeknoten ermöglichen es, Werte an die logischen Funktionen zu übergeben. Wenn Sie einen Eingabeknoten auswählen und das Detailfenster öffnen, können Sie folgende Optionen festlegen:

STANDARDWERT

Sie können den Wert des Knotens festlegen, der zu Beginn der Ausführung verwendet werden soll, bis ein anderer Wert empfangen wird oder wenn der Knoten nicht mit einem anderen Block verbunden ist.

Im Detailfenster werden neben den oben genannten Optionen auch die möglichen Werte angezeigt, die der Knoten annehmen kann. im Falle von binären Knoten sind die möglichen Werte nur 0 (AUS) oder 1 (EIN), im Fall von numerischen Knoten hängen die möglichen Werte von der Art des Knotens ab und können spezifische Einschränkungen aufweisen.

4.2.3 AUGABEKNOTEN

Die Ausgabeknoten geben die Ergebnisse der Logikfunktion zurück, die mit dem Block verknüpft ist, und erlauben es, diese an andere Blöcke weiterzuleiten.

Gewisse Blöcke haben konfigurierbare Optionen deren Ausgabeknoten wie z.B. den Standardwert.

4.2.4 KNOTEN HINZUFÜGEN UND ENTFERNEN

Einige Blöcke haben eine variable Anzahl von Knoten; In diesen Fällen enthält der Block, der aus dem Seitenmenü entnommen wird, typischerweise einen Mindestsatz an Knoten, der durch Drücken der Taste "+" bis auf eine maximale Anzahl von Knoten erhöht werden kann.

Um einen zuvor hinzugefügten Knoten zu entfernen, gehen Sie folgendermaßen vor:

- Wählen Sie den Knoten aus
- Öffnen Sie das Detailfenster
- Drücken Sie die Schaltfläche "Löschen"

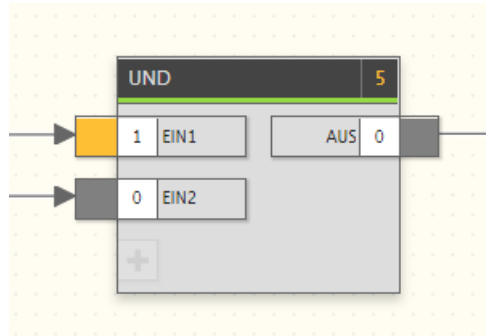
Alle Verbindungen, die dem Knoten eventuell zugeordnet sind, werden gelöscht.

4.3 KOMBINATORISCHE LOGIK

4.3.1 UND

BESCHREIBUNG	Führt die logische UND-Funktion zwischen zwei oder mehr Binäreingängen aus (maximal 10)
---------------------	---

VORSCHAU

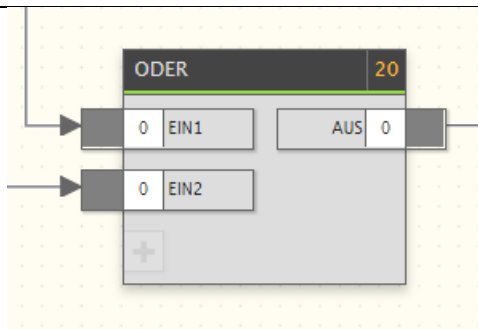


KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN1...EIN10	Eingang Mögliche Werte: 0 → AUS 1 → EIN	X	
	AUS	Ausgang Mögliche Werte: 0 → EIN 1 → AUS		X
	+	Knoten hinzufügen	X	

4.3.2 ODER

BESCHREIBUNG Führt die logische ODER-Funktion zwischen zwei oder mehr Binäreingängen aus (bis zu maximal 10)

VORSCHAU



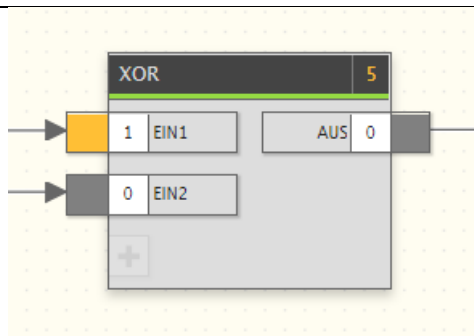
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN1...EIN10	Eingang 1...10 Mögliche Werte: 0 → AUS 1 → EIN	X	
	AUS	Ausgang Mögliche Werte:		X

	0 → AUS 1 → EIN		
+	Knoten hinzufügen	X	

4.3.3 XOR

BESCHREIBUNG Führt die XOR-Logikfunktion zwischen zwei oder mehr Binäreingängen durch (bis zu maximal 10)
G

VORSCHAU

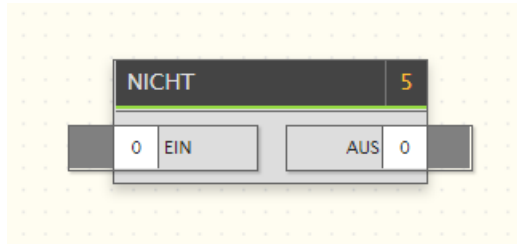


KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN1...EIN10	Eingang 1...10 Mögliche Werte: 0 → AUS 1 → EIN	X	
	US	Ausgang Mögliche Werte: 0 → AUS 1 → EIN		X
	+	Knoten hinzufügen	X	

4.3.4 NOT

BESCHREIBUNG Führt die logische NOT-Funktion der Eingabe aus

VORSCHAU



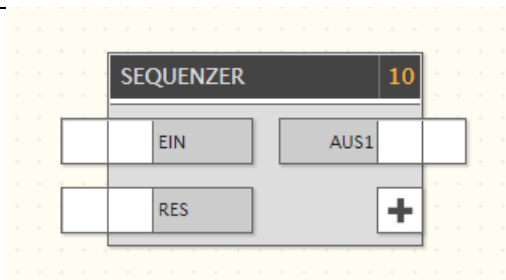
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN1	Eingang 1 Mögliche Werte: 0 → AUS 1 → EIN	X	
	AUS	Ausgang Mögliche Werte: 0 → AUS 1 → EIN		X

4.4 SZENARIEN UND SEQUENZEN

4.4.1 SEQUENZER

BESCHREIBUNG Bei Empfang eines Impulses am Eingang EIN aktiviert und deaktiviert er nacheinander bis zu 10 Ausgänge des Typs Boolean, wobei jeder für eine bestimmte Zeit aktiv bleibt

VORSCHAU



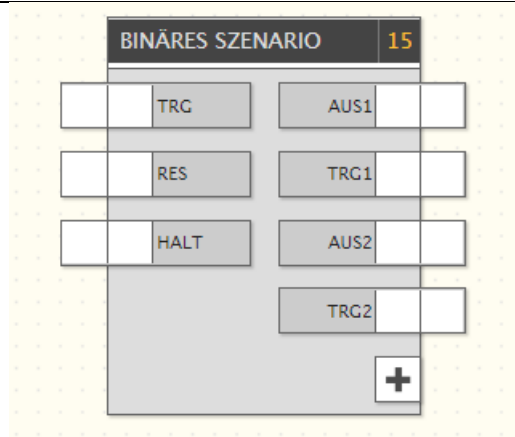
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN	Sequenzanfang Mögliche Werte: 0 → Aus 1 → Ein	X	
	RES	Reset der Seq. Mögliche Werte:	X	

	0 → Aus 1 → Ein		
AUS1...AUS10	Ausgabe 1...10 Mögliche Werte: 0 → Aus 1 → Ein		X
+	Knoten hinzufügen		X
Optionen	Zyklische Sequenz	Definiert ob die Sequenz wiederholt werden soll. Mögliche Werte: WAHR/FALSCH	
	Dauer des Schrittes 1...10	Wartezeit zwischen 2 Schritten Mögliche Werte: von 1 Sek. Bis 12 Std.	

4.4.2 BINÄRES SZENARIO

BESCHREIBUNG Wenn ein Impuls am Eingang TRG empfangen wird, führt es eine Folge von Befehlen vom Booleschen Typ aus, die jeweils einstellbar sind, wobei möglicherweise jeder Befehl mit einer festen Zeit wechselt, die allen Ausgängen gemeinsam ist

VORSCHAU



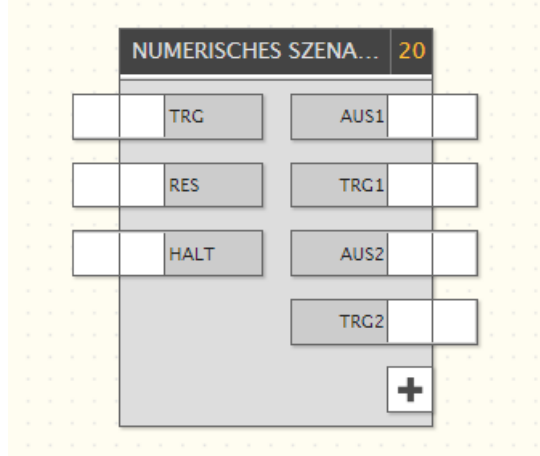
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	TRG	Eingang Auslöser Mögliche Werte: 0 → Aus 1 → Ein	X	
	RES	Reset des Szen. Mögliche Werte:	X	

	0 → Aus 1 → Ein		
AUS1...AUS10	Ausgabe 1...10 Mögliche Werte: 0 → Aus 1 → Ein		X
TRG1...TRG10	Auslöser 1...10 Mögliche Werte: 0 → Aus 1 → Ein		X
+	Knoten (u. Auslöser) hinzufügen		X
Optionen	Ausgabeintervall	Wartezeit zwischen den Ausgangsbefehlen Mögliche Werte 1...60 Sek.	
	Ausgabe 1...10 setzen	Wert an den Ausgängen 1...10 Mögliche Werte: 0 → Falsch (Aus) 1 → Wahr (Ein)	

4.4.3 NUMERISCHES SZENARIO

BESCHREIBUNG Wenn ein Impuls am Eingang TRG empfangen wird, führt es eine Folge von Befehlen des numerischen Typs aus, die jeweils einstellbar sind, wobei möglicherweise jeder Befehl mit einer voreingestellten Zeit wechselt, die allen Ausgängen gemeinsam ist.

VORSCHAU



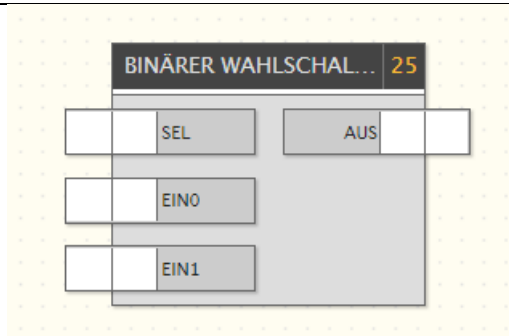
KNOTEN

Etikett	Beschreibung	Eingang	Ausgang
TRG	Eingang Auslöser Mögliche Werte: 0 → Aus 1 → Ein	X	
RES	Reset des Szen. Mögliche Werte: 0 → Aus 1 → Ein	X	
HALT	Stop des Szen. Mögliche Werte: 0 → Aus 1 → Ein	X	
AUS1...AUS10	Ausgang 1...10 Mögliche Werte: jeglicher numerische Wert		X
TRG1...TRG10	Auslöser 1...10 Mögliche Werte: 0 → Aus 1 → Ein		X
+	Knoten u. Auslöser hinzufügen		X
Optionen	Ausgabeintervall	Wartezeit zwischen den Ausgangsbefehlen Mögliche Werte 1...60 Sek.	
	Ausgabe 1...10 setzen	Wert an den Ausgängen 1...10 Mögliche Werte: jeglicher numerische Wert	

4.5.1 BINÄRER WAHLSCHALTER

BESCHREIBUNG Gibt den Wert eines der Eingänge basierend auf dem Wert des SEL-Eingangs als Selektor zurück:
G Wenn SEL = Falsch → AUS = EIN0
 Wenn SEL = True → AUS = EIN1

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	SEL	Eingangswahl Mögliche Werte: 0 → Aus also AUS = EIN0 1 → Ein also AUS = EIN1	X	
	EIN0 EIN1	Eingänge 0 u. 1 Mögliche Werte: 0 → Aus 1 → Ein	X	
	AUS	Ausgang Mögliche Werte: 0 → Aus 1 → Ein		X

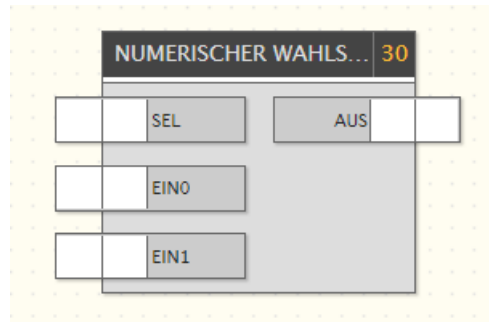
4.5.2 NUMERISCHER WAHLSCHALTER

BESCHREIBUNG Gibt den Wert eines der Eingänge basierend auf dem Wert des SEL-Eingangs als Selektor zurück

Wenn SEL = Falsch → AUS = EIN0

Wenn SEL = True → AUS = EIN1

VORSCHAU



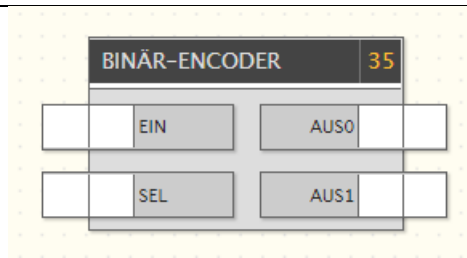
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	SEL	Eingangswahl Mögliche Werte: 0 → Aus also AUS = EIN0 1 → Ein also AUS = EIN1	X	
	EIN0 EIN1	Eingänge 0 u. 1 Mögliche Werte: Beliebiger numerischer Wert	X	
	AUS	Ausgang Mögliche Werte: Beliebiger numerischer Wert		X

4.5.3 BINÄR-ENCODER

BESCHREIBUNG Setzt den Wert von EIN auf einen seiner Ausgänge basierend auf dem Eingangswert SEL, der als Selektor wirkt

Anzahl der Ausgänge: von 2 bis 10

VORSCHAU



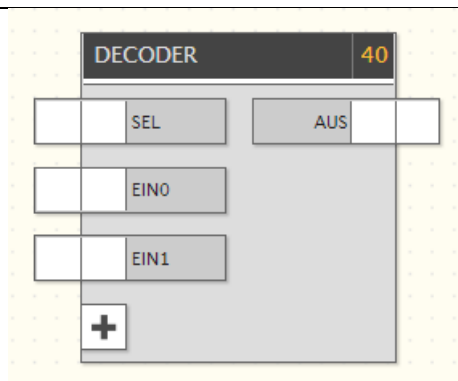
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN	Eingangswahl Mögliche Werte:	X	

	0 → Aus 1 → Ein		
SEL	Ausgangswahl Mögliche Werte: 1...10	X	
AUS0...AUS9	Ausgang 0...9 Mögliche Werte: 0 → Aus 1 → Ein		X
+	Ausgang hinzufügen		X

4.5.4 DECODER

BESCHREIBUN G Der Ausgang gibt den Wert einer der Eingaben zurück basierend auf dem Eingabewert SEL, der als Selektor wirkt.

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	SEL	Eingangswahl Mögliche Werte: 0...9	X	
	EIN0...EIN9	Eingang 0...9 Mögliche Werte: 0 → Aus 1 → Ein	X	
	AUS	Ausgang Mögliche Werte: 0 → Aus 1 → Ein		X

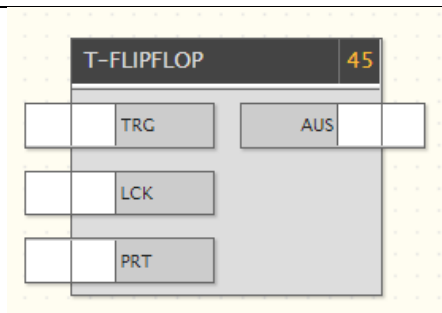
+	Eingang hinzufügen	X	
---	--------------------	---	--

4.5.5 T-FLIPFLOP

BESCHREIBUNG Funktioniert wie ein Schritt-für-Schritt-Relais. Jedes Mal, wenn eine steigende Flanke an seinem Eingang (TRG) erscheint, ändert der Ausgang (AUS) seinen Zustand. Wenn der LCK (Sperr)-Eingang 1 (True) ist, ist der TRG-Effekt gesperrt, so dass sich der Ausgang niemals ändert. Wenn der PRT-Eingang (Priorität) auf 1 steht, nimmt der Ausgang den im Parameter VAL (Prioritätswert) eingestellten Wert an.

Kann beispielsweise verwendet werden, um das Licht eines Korridors zu steuern. Sie können sicherstellen, dass das Licht normalerweise nur gesteuert wird, wenn eine Helligkeitsschwelle erfüllt ist (diese Bedingung würde als LCK konfiguriert) und ist nachts immer eingeschaltet (Flag, das an den PRT-Eingang angeschlossen werden sollte)

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	TRG	Auslöser Mögliche Werte: 0 → Aus 1 → Ein	X	
	LCK	Sperr Mögliche Werte: 0 → Aus 1 → Ein	X	
	PRT	Prioritätsflag Mögliche Werte: 0 → Aus 1 → Ein	X	

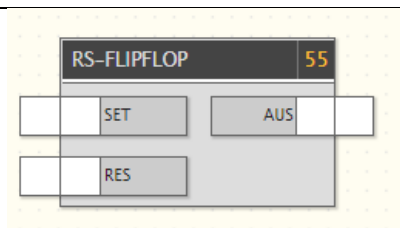
AUS	Ausgang Mögliche Werte: 0 → Aus 1 → Ein		X
Optionen	Prioritätswert	Wert des Ausgangs falls Prioritätsflag gesetzt ist. Mögliche Werte: True/False	

4.5.6 RS-FLIPFLOP

BESCHREIBUNG Elementarer Speicherblock, der mit dem Eingang SET "geladen" und mit dem Eingang RES (Reset) zurückgesetzt wird. Wenn beide Eingänge 1 sind, hat der durch den Parameter "Priorität" festgelegte Vorrang.

Zum Beispiel kann es verwendet werden, um ein Alarmsignal zu verwalten. Ein Alarmkontakt muss mit dem SET verbunden sein. Sobald er auf 1 gesetzt ist, hält der Flipflop den Ausgang auf 1, bis er durch RES zurückgesetzt wird. Auf diese Weise bleibt die Information auch dann erhalten, wenn der Alarm versenkt (auf 0 zurückgeht).

VORSCHAU



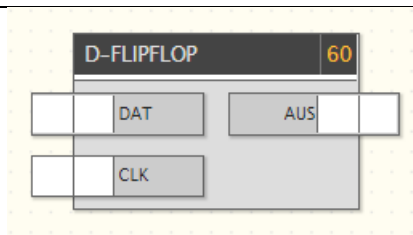
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	SET	SET-Eingang Mögliche Werte: 0 → Aus 1 → Ein	X	
	RES	RESET-Eingang Mögliche Werte: 0 → Aus 1 → Ein	X	
	AUS	Ausgang Mögliche Werte: 0 → Aus 1 → Ein		X
	Optionen	Priorität auswählen	Mögliche Werte: Einstellen Zurücksetzen	

4.5.7 D-FLIPFLOP

BESCHREIBUNG Der Betrieb ist ähnlich dem der D-Verriegelung mit dem Unterschied, dass das D-Flipflop auf die Variation der Front von CLK einwirkt.

Die Daten in DAT werden nur bei der ansteigenden Flanke des CLK-Signals nach AUS geschickt und behalten den Wert bis zur nächsten Flanke von CLK (tatsächlich bildet dieser Block eine Speicherzelle)

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	DAT	Daten Mögliche Werte: 0 → Aus 1 → Ein	X	
	CLK	Uhr Mögliche Werte: 0 → Aus 1 → Ein	X	
	AUS	Ausgang Mögliche Werte: 0 → Aus 1 → Ein		X
	Optionen	Alle 3 Knoten haben Standardwerteinstellungen (Ein/Aus)		

4.5.8 BINÄRE D-VERRIEGELUNG

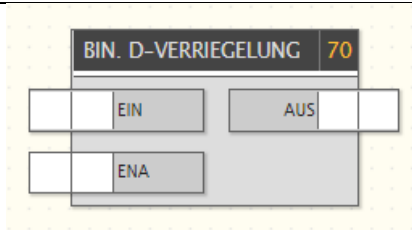
BESCHREIBUNG In diesem Block wird das Eingangssignal EIN zum Ausgang AUS weitergeleitet, wenn das Freigabesignal ENA aktiviert ist (1). Wenn das ENA-Signal deaktiviert ist, bleibt der letzte Zustand am Ausgang (AUS).

Wenn ENA wieder aktiv ist, also (Übergang 0 -> 1), wird der letzte im Eingangsknoten EIN gelesene Wert zum Ausgang AUS gesendet.

Bei ENA = 0 speichert der Verriegelungs-Block also den zuletzt gelesenen Wert, um ihn zu dem Zeitpunkt zu senden, an dem ENA wieder aktiviert wird.

Das Datenformat von EIN und AUS ist binär.

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN	Eingang Mögliche Werte: 0 → Aus 1 → Ein	X	
	ENA	Freigabe Mögliche Werte: 0 → Aus 1 → Ein	X	
	AUS	Ausgang Mögliche Werte: 0 → Aus 1 → Ein		X
	Optionen	Alle 3 Knoten haben Standardwerteinstellungen (Ein/Aus)		

4.5.9 NUMERISCHE D-VERRIEGELUNG

BESCHREIBUNG

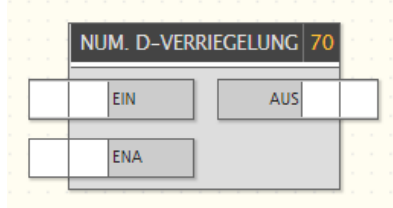
In diesem Block wird das Eingangssignal EIN zum Ausgang AUS weitergeleitet, wenn das Freigabesignal ENA aktiviert (1) ist. Wenn das ENA-Signal deaktiviert ist, bleibt der letzte Zustand am Ausgang (AUS).

Wenn ENA wieder aktiv ist, also (Übergang 0 → 1), wird der letzte im Eingangsknoten EIN gelesene Wert zum Ausgang AUS gesendet.

Bei ENA = 0 speichert der Verriegelungs-Block also den zuletzt gelesenen Wert, um ihn zu dem Zeitpunkt zu senden, an dem ENA wieder aktiviert wird.

Das Datenformat von EIN und AUS ist numerisch

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN	Eingang Mögliche Werte: 0 → Aus 1 → Ein	X	
	ENA	Freigabe Mögliche Werte: 0 → Aus 1 → Ein	X	
	AUS	Ausgang Mögliche Werte: 0 → Aus 1 → Ein		X
	Optionen	Die 2 Eingangsknoten haben Standardwerteeinstellungen (Ein/Aus)		

4.6 VERGLEICHE

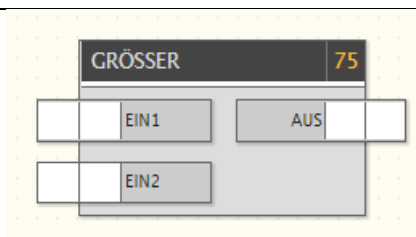
VERGLEICHSDOPERATOREN

BESCHREIBUNG Vergleicht den Wert der beiden Eingänge und gibt als Ausgabe TRUE / FALSE gemäß dem spezifischen Operator zurück.

Verfügbare Operatoren:

- Größer
- Größer oder gleich
- Kleiner
- Kleiner oder gleich
- Gleich
- Nicht gleich

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
--------	---------	--------------	---------	---------

EIN1 EIN2	Eingang 1, 2 Mögliche Werte: Beliebige numerische Werte	X	
AUS	Vergleichsergebnis 0 → Aus 1 → Ein		X

4.7 OPERATIONEN

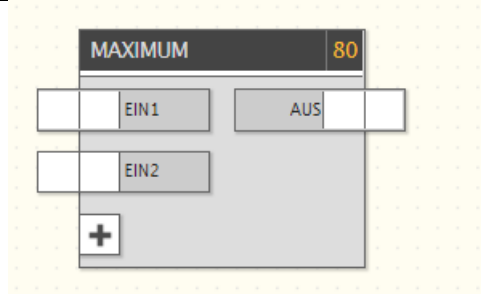
4.7.1 MATHEMATISCHE OPERATOREN

BESCHREIBUNG Führt eine mathematische Operation auf den Eingängen basierend auf dem Typ des Operators durch

Verfügbare Operatoren:

- Maximum
- Minimum
- Mittelwert
- Summe
- Subtraktion
- Multiplikation
- Division
- Bereich
- Absoluter Wert
- Log10
- Integration

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
--------	---------	--------------	---------	---------

EIN1 EIN2 *	Eingang 1, 2 Mögliche Werte: Beliebige numerische Werte	X	
AUS	Ergebnis Mögliche Werte: Jeglicher numerische Wert		X

* Die Anzahl der Eingänge kann je nach Operation begrenzt werden (z.B.: Division max. 2, absoluter Wert max. 1)

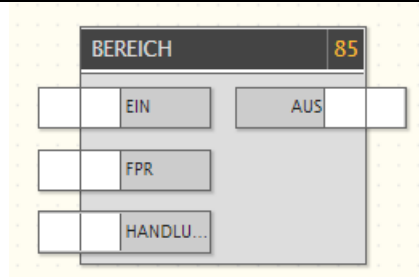
4.7.1.1 BEREICH

BESCHREIBUNG Führt eine lineare Interpolation des Eingangswerts EIN durch, basierend auf einem zugeordneten Mapping, auch "Charakteristik" od. „Merkmal“ genannt, das durch zwei Wertepaare (X, Y) definiert ist. Der EIN-Wert steht in Beziehung zwischen X0 und X1, und dieses Verhältnis wird wiederum zwischen den Werten Y0 und Y1 berechnet, um den Ausgabewert zu bestimmen.

Wenn der Prioritätsmodus eingestellt ist, wird ein voreingestellter Wert zurückgegeben.

Das typische Anwendungsgebiet dieses Blocks ist die Umwandlung von Werten zwischen verschiedenen Größen.

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN1	Eingang Mögliche Werte: Beliebiger numerischer Wert	X	

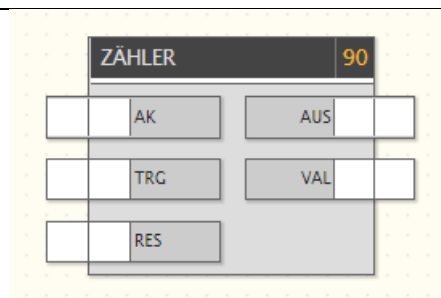
FPR	Priorität aktiviert Mögliche Werte: 0 → nein 1 → Prioritätswert wird ausgegeben	X	
ACT	Direkte/verkehrte Funktion Mögliche Werte: 0 → direkt 1 → verkehrt	X	
AUS	Ausgang Mögliche Werte: Jeglicher numerische Wert		X
OPTIONEN	X0 Y0 X1 Y1	Charakteristiken der linearen Interpolation Mögliche Werte: Jeglicher numerische Wert	
	Prioritätswert	Auszugebender Wert falls Priorität gesetzt ist	

4.8 ZÄHLER

4.8.1 ZÄHLER, COUNTDOWN, VOR-/RÜCKWERTSZÄHLER

BESCHREIBUNG Zählt die Anzahl der Impulse, die am Eingang (TRG) empfangen werden, und erhöht oder verringert sich bei jedem Empfang - je nach Zählertyp.
Zählertypen: Zähler, Countdown, Vor-/Rückwertzähler

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	AK	Aktivierung Mögliche Werte: 0 → nicht aktiviert 1 → aktiviert	X	
	TRG	Auslöser Mögliche Werte: 0 → Aus 1 → Ein(erhöht Zähler)	X	
	RES	Reset Mögliche Werte: 0 → Aus 1 → Ein (Zähler zurückgesetzt)	X	
	AUS	Ausgang Mögliche Werte: 0 → Aus 1 → Ein		X
	VAL	Aktueller Wert Mögliche Werte: Jeglicher numerische Wert		X
	Optionen	Preset	Standardwert der eingestellt wird, wenn der Reset durchgeführt wird oder wenn die Logik gestartet wird. Mögliche Werte: Jeglicher numerischer Wert	

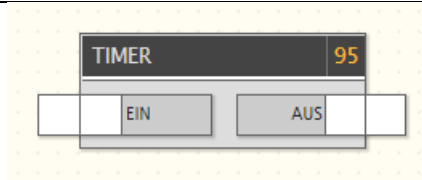
4.9 TIMER & ZEITPLÄNE

4.9.1 TIMER

BESCHREIBUNG Verzögert den empfangenen Eingangswert um eine voreingestellte Zeit.

Wenn eine 1 am Eingang (EIN) empfangen wird (steigende Flanke), startet ein interner Zähler bis zu der als "steigende Verzögerung" angegebenen Zeit, danach wird der Ausgang auf 1 angehoben; umgekehrt wartet der Block beim Empfang einer 0 am Eingang (fallende Flanke) auf die als "fallende Verzögerung" angegebene Zeit, bevor er den Ausgang auf 0 setzt.

VORSCHAU



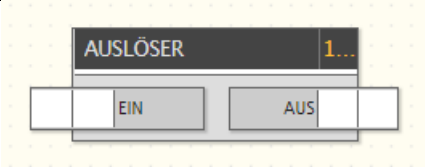
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN	Eingang Mögliche Werte: 0 → Aus 1 → Ein	X	
	AUS	Ausgang, vom Timer verzögert Mögliche Werte: 0 → Aus 1 → Ein		X
	Optionen	Steigende Verzögerung	Verzögerung der Weiterleitung der am Eingang empfangenen steigenden Flanke Mögliche Werte: von 1 Sekunde bis 24 Stunden	
		Fallende Verzögerung	Verzögerung der Weiterleitung der fallenden Flanke, die am Eingang empfangen wurde Mögliche Werte: 1 Sekunde bis 24 Std	

4.9.2 AUSLÖSER (TRIGGER)

BESCHREIBUNG Erzeugt bei einer am Eingang erkannten Flanke einen Auslöser (Impuls der Dauer eines Zyklus) am Ausgang.

Wenn er am Eingang 1 empfängt, setzt er den Ausgang für die Dauer eines einzelnen Verarbeitungszyklus auf 1, dann wird der Ausgang wieder auf 0 gesetzt. Auf diese Weise

ist es möglich, einen "Impuls" für Logikblöcke zu erzeugen, die dies erfordern (z.B.: Szenarien, Sequenzer usw.) an der steigenden Flanke des Eingangs.

VORSCHAU				
				
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	EIN	Eingang	X	
	AUS	Impuls		X
	Optionen	Flanke	Steigende oder fallende Flanke, die am Eingang erkannt werden soll.	

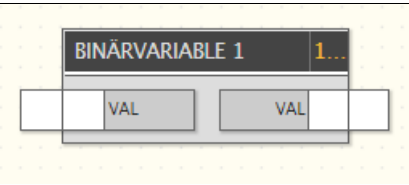
4.10 VARIABLEN

4.10.1 VORWORT

Wie in Abschnitt 3.11 gezeigt, können Variablen Werte zwischen verschiedenen Tasks übergeben. Die Variablen müssen zuvor über die Schaltfläche "+" im entsprechenden Abschnitt des Hauptmenüs erstellt werden, damit sie in die Tasks gezogen werden können, die sie verwenden sollen.

4.10.2 BINÄRE VARIABLEN

BESCHREIBUNG Erlauben einen booleschen Wert zwischen verschiedenen Tasks zu übertragen.

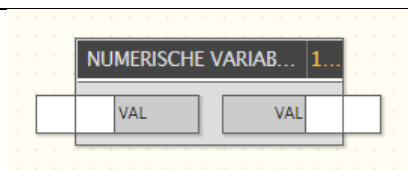
VORSCHAU				
				
KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	VAL	Eingang Mögliche Werte:	X	

	0 → Aus 1 → Ein		
VAL	Aktueller Wert Mögliche Werte: 0 → Aus 1 → Ein		X

4.10.3 NUMERISCHE VARIABLEN

BESCHREIBUNG Erlauben einen numerischen Wert zwischen verschiedenen Tasks zu übertragen.

VORSCHAU



KNOTEN	Etikett	Beschreibung	Eingang	Ausgang
	VAL	Wert den die Variable erhält Mögliche Werte: beliebiger numerischer Wert	X	
	VAL	Aktueller Wert der Variable Mögliche Werte: beliebiger numerischer Wert		X

4.11 KONSTANTEN

Sie brauchen vielleicht in Ihren Logiken Konstanten (z.B. für Vergleiche). Hier können Sie sie erstellen und ihnen den gewünschten Wert zuweisen. Ziehen Sie sie dann in ihre Logik.

5 Auslöser

5.1 EINLEITUNG

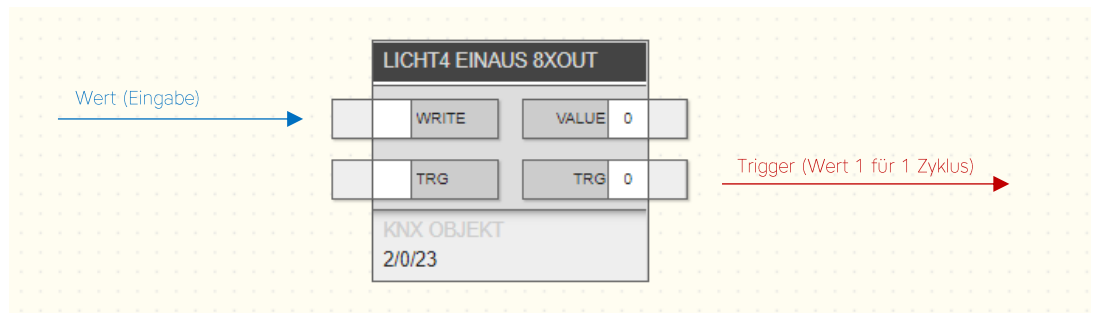
Im Normalfall hält das Logikmodul den Zustand der Objekte an den Ausgängen entsprechend dem Zustand der Logik; sobald diese Objekte ihren Zustand ändern (z.B. weil sie einen Befehl vom Bus, außerhalb der Logik, erhalten), wird ihr Zustand wieder in den der Logik entsprechenden geändert.

Dies ist nicht immer das gewünschte Verhalten; in vielen Situationen möchte man einem Objekt aufgrund der Zustandsänderung anderer Objekte Befehle erteilen, aber wenn dieses Objekt von außen (von der Logik) geändert wird, soll es den neuen Zustand behalten. Mit anderen Worten, Sie würden eine ereignisgesteuerte Logik bevorzugen.

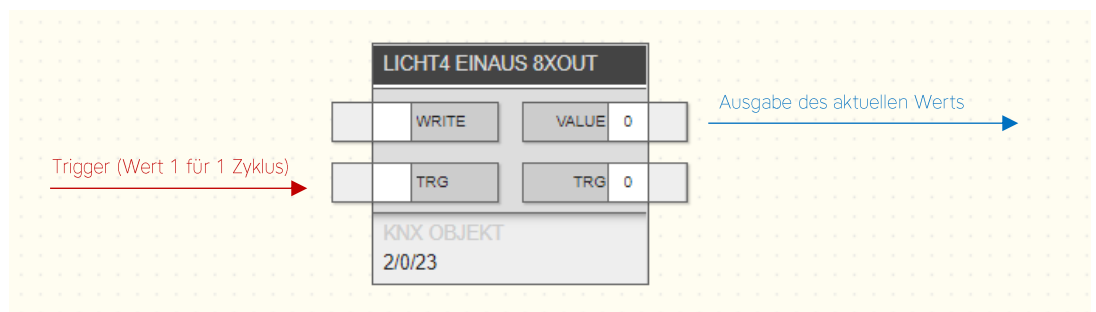
In solchen Fällen ist die Verwendung von Auslösern für die Ein- und Ausgabe erforderlich.

5.2 EINGANGS- UND AUSGANGSTRIGGER

Ausgangstrigger erzeugen jedes Mal einen Impuls (d.h. den Wert 1 für einen einzigen Ausführungszyklus des Skripts, dann wird er auf 0 zurückgesetzt), wenn ein Wert an einem bestimmten Objekt (Eingang) empfangen wird.



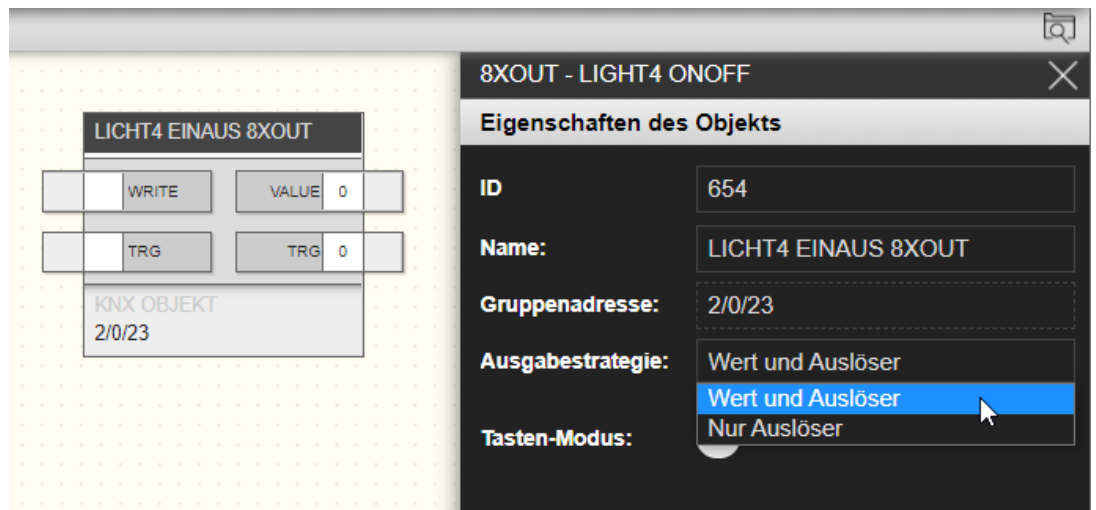
Bei den **Eingangstriggern** hingegen kann man wählen, wann ein Wert gesendet werden soll: Jedes Mal, wenn ein Impuls (Wert 1) an diesen Auslöser gesendet wird, wird der Wert des Objekts an den Bus gesendet - auch wenn er sich seit dem vorherigen Ausführungszyklus nicht geändert hat.



5.3 AUSGABESTRATEGIE

Wie bereits erläutert, werden im Normalfall die Ausgangsknoten eines Objekts innerhalb eines Tasks entsprechend der Wertänderung des Objekts/Blocks angesteuert. Außerdem wird bei einem Impuls am Eingangstrigger (TRG-Knoten links vom Block) der Wert ebenfalls auf den Bus gesendet - auch wenn er sich nicht ändert.

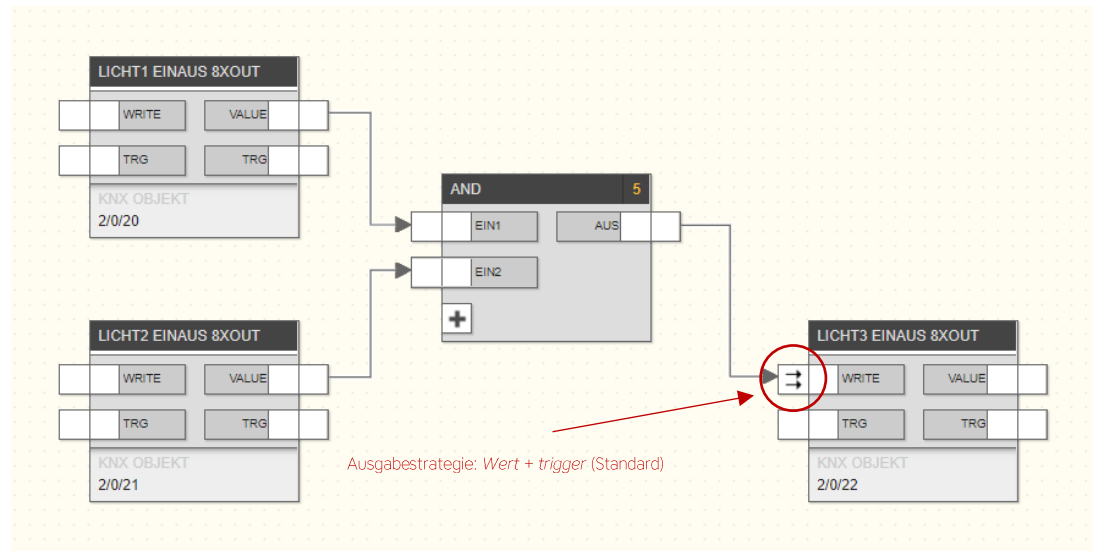
Es ist jedoch möglich, die Ausgangsknoten so einzustellen, dass sie den Wert nur per Auslöser an den Bus senden, indem der Parameter für die Ausgabestrategie im Detailfenster des Blocks geändert wird, nachdem der Block im Logikeditor ausgewählt wurde.



Es wird also nur dann ein Wert auf den Bus gesendet, wenn ein Impuls den Eingangs-Triggerknoten des Blocks erreicht. Andernfalls wird kein Wert gesendet - unabhängig davon, ob sich das Ergebnis der Logik zuvor geändert hat oder ob der Wert des Objekts von außerhalb der Logik geändert wurde (z. B. durch den Benutzer über eine physische Taste).

5.4 TRIGGERVERBINDUNGEN

Nehmen Sie zum Beispiel die folgende einfache Logik:



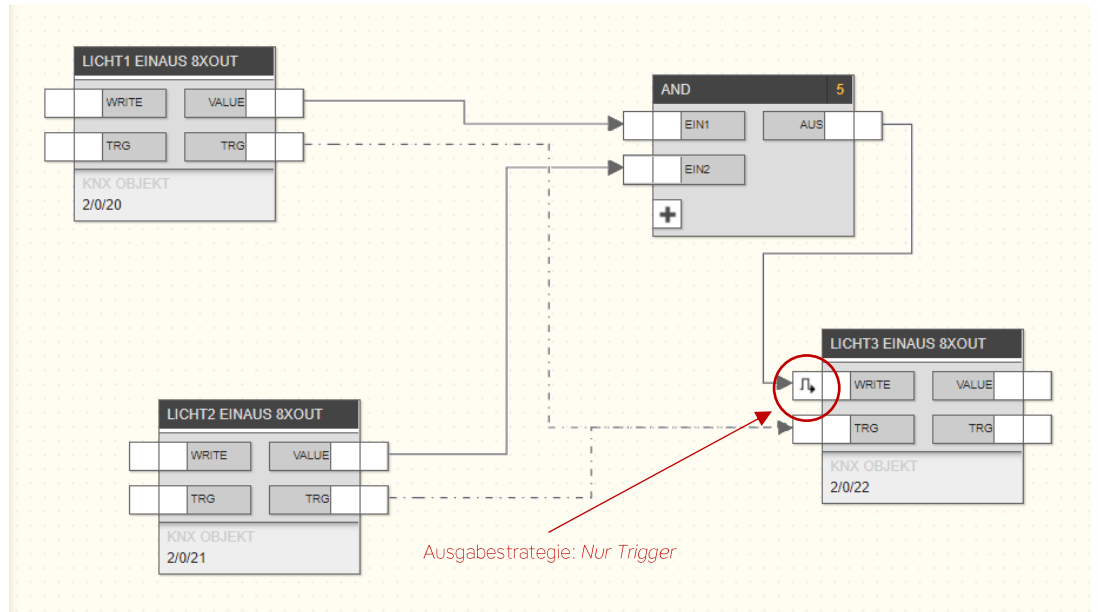
Wie dargestellt, hat der Ausgangsblock Wert(änderung) + Trigger als Ausgabestrategie, was bedeutet, dass:

- Wenn sich einer der Eingänge (Licht 1 oder Licht 2) ändert, erhält Licht 3 den Wert des UND zwischen den beiden Eingängen
- Ändert Licht 3 seinen Wert aus Gründen, die außerhalb der Logik liegen (z.B. ein manueller Befehl eines Benutzers), wird es sofort wieder auf den Wert der UND-Logik gesetzt.

Wenn Sie möchten, dass Licht 3 nur dann geändert wird, wenn sich einer der Eingänge ändert, aber gleichzeitig manuell (d. h. in irgendeiner Weise außerhalb der Logik) geändert werden kann, müssen Sie Folgendes tun:

- die Ausgabestrategie auf "nur Trigger" setzen
- die Ausgabe-Trigger von Licht 1 und Licht 2 mit dem Eingangs-Trigger von Licht 3 verbinden

Um dies zu ermöglichen, können die Eingangstrigger mit mehreren Ausgängen verbunden werden, während die Eingangsknoten normalerweise nur eine einzige Eingangsverbindung pro Knoten innerhalb eines Tasks haben können.



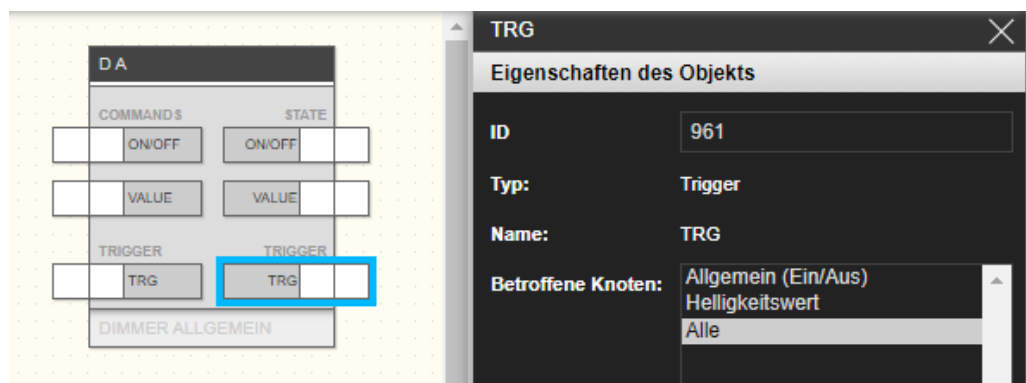
Wenn der Eingangstrigger verbunden wird, schaltet seine Ausgabestrategie automatisch auf "nur Trigger" und das angezeigte Symbol auf dem Knoten folglich auch.

Es ist immer noch möglich, die Ausgabestrategie manuell anzupassen; sie könnte z.B. verwendet werden, um den Wert zusätzlich zu den Wertänderungen auch periodisch zu senden, wie es viele KNX Geräte tun, indem eine periodische Logik im Vorfeld eingebaut wird.

5.5 AUSLÖSER UND KOMPLEXE OBJEKTE

Für Komplexe Objekte ist ein einziges TRG verfügbar (nicht eines für jedes Unterobjekt). In diesem Fall ist es möglich, im Detailfenster (nach Auswahl des Triggerknotens) anzugeben, auf welches Unterobjekt sich der Trigger beziehen soll.

Im Falle von Ausgangstriggern ist es möglich, über den Selektor für *Betroffene Knoten* auszuwählen, welches Unterobjekt bei einer Zustandsänderung einen Impuls auslösen soll. Standardmäßig sind alle Unterobjekte aktiv, so dass jede Wertänderung einen Impuls in der Logik auslöst, aber es ist möglich, die Auswahl auf eine Teilmenge oder ein einzelnes Unterobjekt zu reduzieren.



In ähnlicher Weise ist es möglich, für Eingangsauslöser festzulegen, welche Teilobjekte bei Empfang eines Impulses dabei betroffen sein sollen.

The image shows a screenshot of the DIVUS OPTIMA software interface. On the left, there is a control panel for a device labeled 'D A'. The panel is divided into several sections: 'COMMANDS' and 'STATE' at the top, each containing 'ON/OFF' and 'VALUE' buttons; 'TRIGGER' below, containing two 'TRG' buttons; and 'DIMMER ALLGEMEIN' at the bottom. The leftmost 'TRG' button is highlighted with a blue border. On the right, a properties window titled 'TRG' is open, showing the following details:

Eigenschaften des Objekts	
ID	960
Typ:	Trigger
Name:	TRG
Betroffene Knoten:	Allgemein (Ein/Aus) Helligkeitswert Alle

6 Verbindungen

6.1 EINFÜHRUNG

Wie wir bereits gesehen haben, müssen Sie, um einen logischen Task zu erstellen, Ausgabeknoten mit Eingabeknoten verschiedener Blöcke verbinden. In diesem Kapitel werden die Verbindungstypen, ihre Optionen und ihre Auswirkungen auf den LUA-Code, der einem Logikprogramm zugrunde liegt, besprochen.

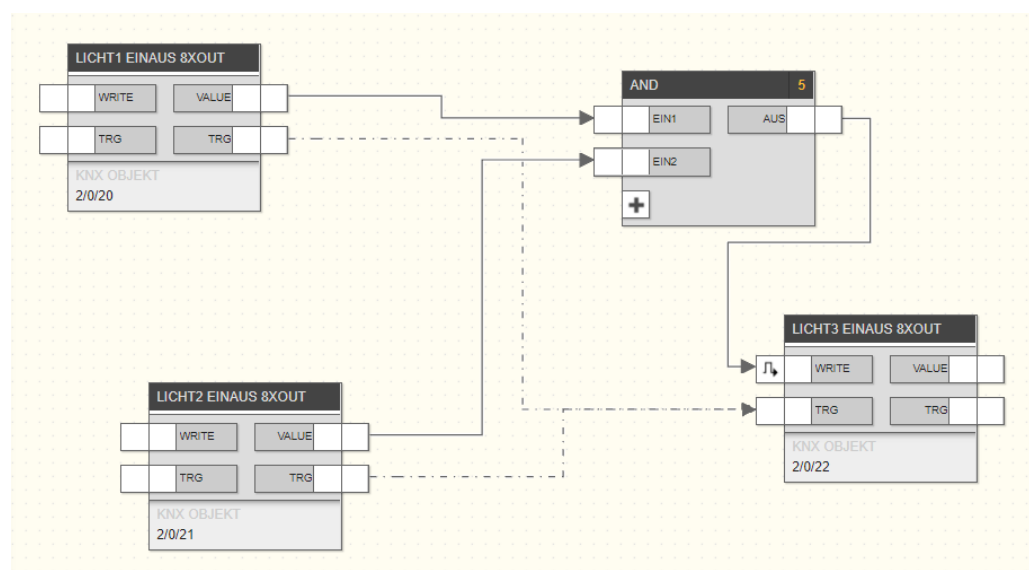
6.2 VERBINDUNGSARTEN

Abhängig vom Knotentyp, den sie verbinden, gibt es 2 Arten von Verbindungen:

- Standard-Verbindungen
- Trigger-Verbindungen

Die ersten verbinden Ausgabe- und Eingabeknoten, seien es Optima-Objekte oder logische Blöcke aus der Bibliothek, und übergeben boolesche oder numerische Werte (je nach Blocktyp). Diese Verbindungstypen stellen den Übergang eines Wertes zwischen einem Ausgangsknoten eines Blocks und dem Eingangsknoten eines anderen Blocks während der Ausführung des Logikprogramms dar.

Trigger-Verbindungen hingegen leiten einen Impuls von der Dauer eines einzelnen Ausführungszyklus weiter, wenn er vom TRG-Knoten eines Optima-Objekts erzeugt wird; normalerweise ist dieser Impuls mit einem oder mehreren anderen Eingangs-TRG-Knoten anderer Objekte verbunden, um nur dann eine Ausgabe zu haben, wenn er ausgelöst wird (wie im Kapitel über Trigger erklärt), aber er kann auch für jede andere Art von Eingang des Logiknetzes verwendet werden, zusammen mit anderen Triggern oder booleschen Werten. Die Verbindungen von Ausgangs-Triggern werden im Editor als punkt-gestrichelte Linien dargestellt, wie Sie hier sehen können:



7 Simulation

7.1 EINLEITUNG

Sobald ein logischer Task realisiert wurde, ist es möglich, dessen Betrieb innerhalb des Editors zu simulieren, indem man den Status der Eingänge manuell einfügt und die Verarbeitung der Ausgänge in Echtzeit überprüft, sogar durch die Logikblöcke, die eine Variation der Ausgänge im Laufe der Zeit beinhalten.

7.2 SIMULATIONSARTEN

Zwei Arten von Simulation sind verfügbar:

- **Kontinuierliche Simulation:** Die Ausführung des Tasks findet im Hintergrund statt und wird von Änderungen des Knotenstatus in Echtzeit beeinflusst.
- **Schritt-für-Schritt-Simulation:** Jeder Task-Ausführungszyklus muss manuell gestartet werden, und der Status der Knoten kann zwischen einem und dem nächsten Zyklus geändert werden.

Der erste Typ ermöglicht eine realistischere Auswertung der realisierten logischen Netzwerke, der zweite erlaubt eine gründliche und pünktliche Überprüfung jedes einzelnen Wertedurchgangs zwischen Blöcken und bietet ein höheres Diagnose-Niveau.

7.3 GRAFISCHE SIMULATIONSUMGEBUNG

Durch Drücken einer der Simulationstasten (kontinuierlich oder Schritt für Schritt) erfährt das Editor-Fenster folgende Änderungen:

- Das Detailfenster wird geschlossen, um maximalen Arbeitsraum für die Simulation zu bieten
- Jede Operation von Drag & Drop, Verknüpfen, Ändern oder Löschen der Inhalte des Tasks ist blockiert
- Die Knoten nehmen je nach Zustand eine Farbe an und erlauben es, den Wert manuell zu erzwingen (wie unten beschrieben)

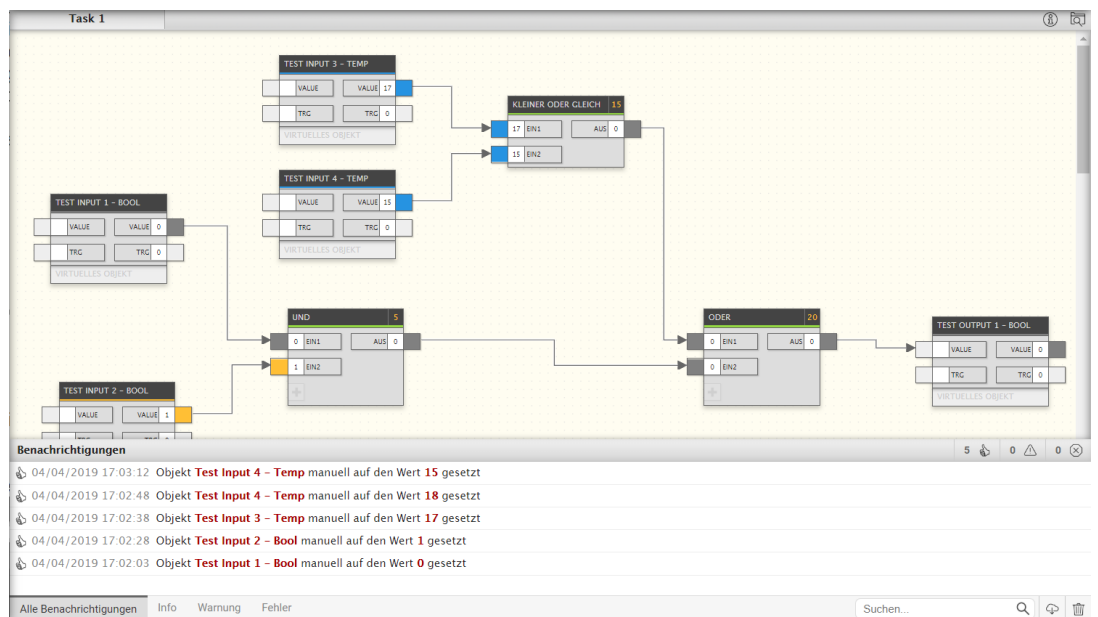
Die Farbe der Knoten folgt der folgenden Konvention:

Binärknoten	grau	Wert 0 (AUS)
	gelb	Wert 1 (EIN)
<hr/>		
Numerische Knoten	blau	Jeder Wert

Während der Simulation zeigt der Editor im Benachrichtigungsbereich eine Reihe von Informationen an über Programmausführung, manuelle Statusänderungen (vom Benutzer ausgeführt) und automatische (von Logikblöcken erkannte). Darüber hinaus werden während der schrittweisen Simulation viele "Debug"-Nachrichten gemeldet, die eine gründliche Analyse der Programmausführung ermöglichen, die insbesondere im Falle von Fehlern oder Fehlfunktionen nützlich ist.

Der Benachrichtigungsbereich, der normalerweise geschlossen, um maximalen Platz für die Simulation zu bieten, kann geöffnet werden, um die Nachrichten zu lesen, deren Nummer - abhängig vom Typ - im rechten Teil der Meldungsleiste sichtbar ist, auch wenn er geschlossen ist. Weitere Details zum Benachrichtigungsbereich finden Sie in Abschnitt 2.6.

Die folgende Abbildung zeigt ein Beispiel für eine Simulation mit geöffnetem Benachrichtigungsbereich:



7.4 MANUELLE WERTINGABE

Um den Status eines Knotens manuell einzustellen, gehen Sie folgendermaßen vor:

- Doppelklicken auf den Knotenwert

- den aktuellen Wert löschen und den neuen Wert eingeben
- *Eingabe* Drücken

Die Farbe des Knotens (wenn binär) ändert sich entsprechend dem neuen Wert und dieser wird an dem Simulator übergeben, der ihn sofort (im Fall der kontinuierlichen Simulation) oder zum nächsten Ausführungszyklus (im schrittweisen Modus) propagiert.

7.5 SIMULATION ANHALTEN

Die Simulation kann jederzeit durch Drücken der Schaltfläche "Halt" in der Symbolleiste gestoppt werden (außerhalb der Simulation nicht zugänglich).

8 Zeichenutensilien

8.1 EINLEITUNG

Um die Lesbarkeit logischer Tasks insbesondere bei komplexen logischen Netzwerken zu erhöhen, stellt der Editor einige Zeichenutensilien zur Verfügung, mit denen der Benutzer Anmerkungen eingeben und Bereiche des Tasks hervorheben kann.

Diese Werkzeuge sind im Bereich "Logikmodul" des Hauptmenüs unter dem Punkt "Zeichenutensilien" verfügbar. Sie können, wie bereits erwähnt, wie die anderen Arten von Objekten in die logischen Tasks gezogen werden.

8.2 BESCHRIFTUNGEN

Mit den Beschriftungen können Sie freien Text in die Tasks einfügen. Es ist möglich, eine unbegrenzte Anzahl von Beschriftungen für jeden logischen Task einzufügen.

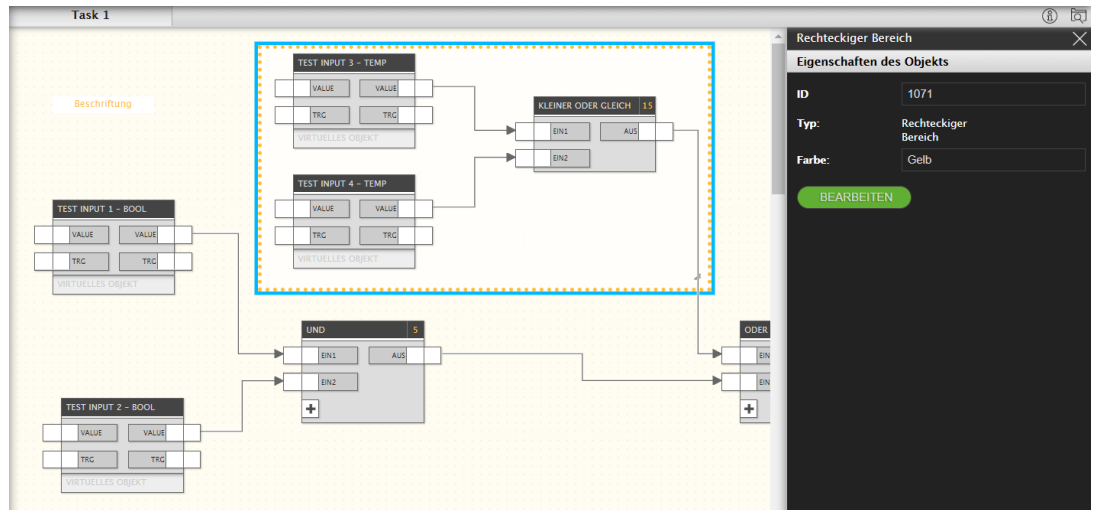
Sobald eine Beschriftung in einen Task gezogen und an dem gewünschten Punkt positioniert wurde, ist es möglich, sie anzupassen, indem das Detailfenster geöffnet wird (nachdem sie ausgewählt wurde); Folgende Optionen sind verfügbar:

- **Text** Text, der im logischen Task angezeigt wird
- **Farbe** Ermöglicht, die Farbe des Textes auszuwählen (Schwarz, Gelb, Rot, Grün, Blau, Grau)
- **Schriftgröße** Erlaubt die Schriftgröße zu wählen (XS, S, N, L, XL)

Beschriftungen können aus den Tasks entfernt werden, indem direkt die ENTFERNEN-Taste auf der Tastatur gedrückt wird, nachdem sie ausgewählt wurden.

8.3 RECHTECKIGE BEREICHE

Es ist möglich, eine oder mehrere Teile der Logikeinheit hervorzuheben, indem man farbige rechteckige Bereiche aus dem Hauptmenü zieht, wie in der folgenden Abbildung beispielhaft dargestellt:



Sobald Sie einen rechteckigen Bereich in einen Task ziehen, können Sie:

- die Größe ändern, indem Sie den entsprechenden Cursor in der unteren rechten Ecke ziehen
- die Rahmenfarbe mithilfe des Farbauswahlfelds im Detailbereich ändern

Die rechteckigen Bereiche werden immer unter den Blöcken und ihren Verbindungen gezeichnet; Sie unterstützen keine Mehrfachauswahl wie Blöcke oder Beschriftungen. Um sie anzupassen oder sie aus dem Task zu entfernen, müssen Sie einzeln auf sie klicken und die Werkzeuge im Detailbereich verwenden (Farbe und Schaltfläche ändern) oder die "Entfernen"-Taste der Tastatur drücken, um sie aus dem Task zu entfernen.

9 Anhang

9.1 GLOSSAR

Logisches Modul	Die Katalognomenklatur ist "Logisches Modul".
Logikeinheit	Logisches Netzwerk, das aus einem oder mehreren logischen Tasks besteht.
Logischer Task	Logisches Netzwerk, bestehend aus einem oder mehreren verknüpften Bausteinen und Logiken. Jede Logikeinheit kann bis zu 16 logische Tasks enthalten.
Logischer Block	Block, der in eine Logikeinheit eingefügt werden kann, um eine bestimmte Funktion auszuführen und mit anderen Blöcken über Eingangs- und/oder Ausgangsknoten zu interagieren
Block	Baustein, der in ein Logikeinheit eingefügt werden kann, um Informationen auf dem Heimautomationsbus zu lesen und/oder zu schreiben und mit anderen Blöcken über Eingangs- und/oder Ausgangsknoten zu interagieren
Knoten	Einzelelement eines Logikblocks, der eine bestimmte Information in der Eingabe oder Ausgabe bereitstellt; Die Knoten können über entsprechende Verbindungen mit anderen <i>Knoten</i> verbunden sein
Verbindung	Verknüpfung zwischen zwei Knoten aus zwei Blöcken. Die Verbindung hat eine „Richtung“, die die Reihenfolge bestimmt, in der Informationen zwischen den Knoten ausgetauscht werden; insbesondere wird der Status des Quellknotens an den Zielknoten übergeben.
Editor	Grafische Konfigurationsumgebung für Logikeinheiten. Er ermöglicht, logische Tasks für die im Projekt vorhandenen logischen Einheiten zu erstellen und die für ihre Ausführung erforderlichen Informationen innerhalb des Projekts zu erhalten.

