

TURCK

Your Global Automation Partner

ARGEE

Getting Started

Inhaltsverzeichnis

1	Über diese Anleitung	3
1.1	Symbolerläuterung	3
1.2	Zielgruppen	3
1.3	Feedback zu dieser Anleitung	3
2	Hinweise zur Software	4
2.1	Produktidentifizierung	4
2.2	Voraussetzungen	4
2.3	Turck-Service	4
2.4	Geräte mit ARGEE-Funktion.....	4
3	Zu Ihrer Sicherheit	5
3.1	Bestimmungsgemäße Verwendung	5
3.2	Naheliegende Fehlanwendung.....	5
4	Systembeschreibung ARGEE	6
4.1	Funktionsprinzip.....	6
4.2	ARGEE Feature-Übersicht.....	6
4.3	Betriebsarten ARGEE PRO und ARGEE Flow	7
4.4	ARGEE-Modus-Auswahl.....	8
5	ARGEE in Betrieb nehmen	10
6	Übersicht ARGEE-Editor	11
6.1	Projekt einrichten	12
6.1.1	Projekt-Tab	12
6.1.2	Projekte öffnen.....	13
6.1.3	Projekte speichern	14
6.1.4	Programmcode als PDF anzeigen oder drucken	15
6.1.5	Projekt benennen.....	15
6.1.6	IO-Block konfigurieren.....	16
6.1.7	Informationen über ARGEE-Version	16
6.2	In der Projektumgebung navigieren	17
6.2.1	Kontextmenüs in ARGEE.....	17
6.2.2	Bedienungshilfen	19
6.3	Programm einrichten	22
6.3.1	Program Variables	22
6.3.2	Variablen initialisieren	23
6.3.3	Array erstellen.....	23
6.3.4	Alias Variables erstellen.....	25
6.3.5	Function Blocks, States und Additional Global Variables erstellen.....	26
6.3.6	Bitfields und I/O-Mapping.....	28
6.3.7	Bibliotheken (Libraries) importieren	32
6.3.8	Program-Blocks	33
6.4	Programm ausführen	36
6.4.1	Programm kompilieren	36
6.4.2	Debug Optionen.....	36
6.4.3	HMI.....	39
7	Turck-Niederlassungen – Kontaktdaten	40

1 Über diese Anleitung

Dieses Dokument beschreibt die Integrated Development Environment (IDE) und die grundlegenden Funktionen von ARGEE 4.

1.1 Symbolerläuterung

In dieser Anleitung werden folgende Symbole verwendet:



GEFAHR

GEFAHR kennzeichnet eine gefährliche Situation mit hohem Risiko, die zum Tod oder zu schweren Verletzungen führt, wenn sie nicht vermieden wird.



WARNUNG

WARNUNG kennzeichnet eine gefährliche Situation mit mittlerem Risiko, die zum Tod oder zu schweren Verletzungen führen kann, wenn sie nicht vermieden wird.



VORSICHT

VORSICHT kennzeichnet eine gefährliche Situation mit mittlerem Risiko, die zu mittelschweren oder leichten Verletzungen führen kann, wenn sie nicht vermieden wird.



ACHTUNG

ACHTUNG kennzeichnet eine Situation, die zu Sachschäden führen kann, wenn sie nicht vermieden wird.



HINWEIS

Unter HINWEIS finden Sie Tipps, Empfehlungen und nützliche Informationen zu speziellen Handlungsschritten und Sachverhalten. Die Hinweise erleichtern Ihnen die Arbeit und helfen Ihnen, Mehrarbeit zu vermeiden.



HANDLUNGSAUFFORDERUNG

Dieses Zeichen kennzeichnet Handlungsschritte, die der Anwender ausführen muss.



HANDLUNGSRISIKO

Dieses Zeichen kennzeichnet relevante Handlungsrisikofaktoren.

1.2 Zielgruppen

Die vorliegende Anleitung richtet sich an Personal mit SPS- oder Hochsprach-Programmierenkenntnissen. Diese Anleitung muss von jeder Person sorgfältig gelesen werden, die die Software in Betrieb nimmt oder betreibt.

1.3 Feedback zu dieser Anleitung

Wir sind bestrebt, diese Anleitung ständig so informativ und übersichtlich wie möglich zu gestalten. Haben Sie Anregungen für eine bessere Gestaltung oder fehlen Ihnen Angaben in der Anleitung, schicken Sie Ihre Vorschläge an techdoc@turck.com.

2 Hinweise zur Software

2.1 Produktidentifizierung

Dieses Handbuch gilt für alle ARGEE-fähigen Turck-Geräte.

Diese Anleitung ist auf der Basis von ARGEE 4 erstellt worden.

2.2 Voraussetzungen

Benötigte Hardware und Software:

- PC
- Browser (mit HTML5- und Javascript-Unterstützung)
- ARGEE Integrated Development Environment

Optionale Hardware und Software:

- Ethernet-Leitung
- ARGEE-fähiges Gerät
- Turck Automation Suite (TAS)

Die ARGEE-Programmierungsumgebung und die Turck Automation Suite (TAS) stehen unter www.turck.com zum Download zur Verfügung.



HINWEIS

Turck stellt für ARGEE auf www.turck.com Programmierbeispiele und Bibliotheken (Libraries) zur Verfügung. Weitere Informationen zu Bibliotheken (Libraries) finden Sie auf S. [▶ 32].

2.3 Turck-Service

Turck unterstützt Sie bei Ihren Projekten von der ersten Analyse bis zur Inbetriebnahme Ihrer Applikation. In der Turck-Produktdatenbank unter www.turck.com finden Sie Software-Tools für Programmierung, Konfiguration oder Inbetriebnahme, Datenblätter und CAD-Dateien in vielen Exportformaten.

Die Kontaktdaten der Turck-Niederlassungen weltweit finden Sie auf S. [▶ 40].

2.4 Geräte mit ARGEE-Funktion

Der ARGEE-4-Funktionsumfang wird ab Runtime-Version 4.1.0.0 unterstützt. Die BLCEN-Geräte der BL compact-Gerätesfamilie unterstützen den ARGEE-3-Funktionsumfang.

Die ARGEE-4-Runtime wird von BL compact-Geräten nicht unterstützt.

3 Zu Ihrer Sicherheit

3.1 Bestimmungsgemäße Verwendung

ARGEE ist ein autarkes Software-Paket bestehend aus ARGEE-Engineering (IDE) und ARGEE-
Runtime, mit der Turcks Block-I/O-Module (ARGEE-fähige Geräte) als Logic Controller eingesetzt
werden können.

Die Software darf nur wie in dieser Anleitung beschrieben verwendet werden. Jede andere Ver-
wendung gilt als nicht bestimmungsgemäß. Für daraus resultierende Schäden übernimmt
Turck keine Haftung.

3.2 Naheliegende Fehlanwendung

- ARGEE ist nicht zur Programmierung und Steuerung von Sicherheitsanwendungen be-
stimmt und darf nicht zum Personen- und Sachschutz eingesetzt werden.

4 Systembeschreibung ARGEE

Diese Anleitung beschränkt sich auf die Beschreibung der Funktionen des ARGEE PRO Mode. Eine detaillierte Beschreibung der Funktionen des ARGEE Flow Mode finden Sie im ARGEE V3 Reference Manual.

4.1 Funktionsprinzip

Mithilfe der ARGEE-IDE können für Turcks ARGEE-fähige Geräte ARGEE-Projekte erstellt werden.

ARGEE-fähige Geräte sind durch die einprogrammierten Logikfunktionen in der Lage:

- Arithmetische Operationen auszuführen
- Interne Variablen, Timer und Counter bis zu 6 kByte Gesamtgröße zu verwenden
- Umfangreichere Daten mit einer SPS auszutauschen
- Kontrollstrukturen auszuführen (bedingte Anweisung, Schleife, Blockstruktur)



HINWEIS

Bei der Anbindung eines ARGEE-Projekts an eine SPS mit Profinet muss die ARGEE-GSDML mit einfachem Byte-Aufbau verwendet werden.

4.2 ARGEE Feature-Übersicht

- ARGEE funktioniert als eigenständige Anwendung. Eine SPS wird zur Ausführung von Logik nicht benötigt.
- ARGEE kann im Verbund mit einer SPS verwendet werden.
- Wenn die Anwendung die Kommunikation mit der SPS verliert, kann ARGEE unterstützen die Anwendung in einen kontrollierten Betriebszustand zu versetzen.
- ARGEE kann zur Vorverarbeitung oder für lokale Schnellreaktionen in Verbindung mit einer SPS eingesetzt werden.
- Lokale Steuerung (ARGEE kann eine Anwendung überwachen und Updates an die SPS zurücksenden).
- ARGEE kann zur Programmierung von HMI-Screens verwendet werden.
- ARGEE Simulation Mode (ermöglicht die Parametrierung ohne ARGEE-fähiges Gerät).
- Exportfunktionen des ARGEE-Programmcodes in verschiedenen Formaten
- Importfunktionen für ARGEE-Projekte
- Debug-Funktionen
- Unterstützung von for-, while- und if-statements
- Highspeed Digital IO (max. 4 kHz)
- ARGEE-Bibliotheken
- Unterstützung von Bitfield-Datenstrukturen: Dezimalpunkt-Schreibweise erlaubt das ansteuern einzelner Bits
- Function Blocks

ARGEE-Features in TAS

Folgende Features können nur in TAS ausgeführt werden:

- Schreib- und Löschschutz von ARGEE-Projekten (auch gegen Factory Reset geschützt) aktivieren/deaktivieren
- ARGEE-Projekt auf ausgewählte Geräte laden
- ARGEE-Projekt auf ausgewählten Geräten löschen

4.3 Betriebsarten ARGEE PRO und ARGEE Flow

ARGEE verfügt über zwei Betriebsarten: ARGEE PRO und ARGEE Flow.

ARGEE PRO ist die Standardeinstellung und die empfohlene IDE-Betriebsart.

ARGEE PRO

Mit ARGEE PRO sind ARGEE-fähige Geräte in der Lage:

- Arithmetische Operationen auszuführen
- Interne Variablen, Timer und Counter bis zu 6 kByte Gesamtgröße zu verwenden
- Umfangreichere Daten mit einer SPS auszutauschen
- IF-Bedingungen und Zustandsfolgen auszuführen

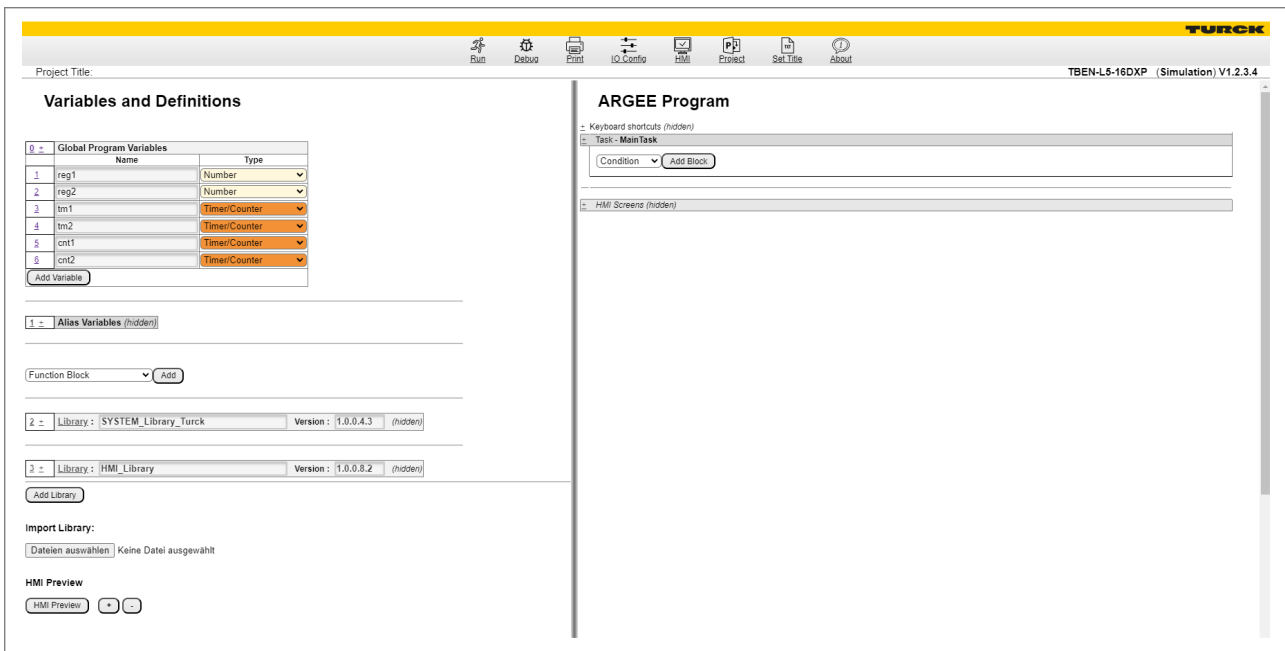


Abb. 1: ARGEE PRO Oberfläche

ARGEE Flow

ARGEE Flow ist eine eingeschränkte Konfigurationsumgebung, in der über Drop-Down-Felder mithilfe von Wenn-Dann-Verknüpfungen Steuerungsfunktionen festgelegt und visualisiert werden können.

Die ARGEE Flow-Umgebung besteht aus drei Bereichen:

1. Bedingungen (Conditions)
2. Operationen (Operations)
3. Aktionen (Actions)

4.4 ARGEE-Modus-Auswahl

Der gewünschte Modus muss vor Beginn des Programmiervorgangs ausgewählt werden.

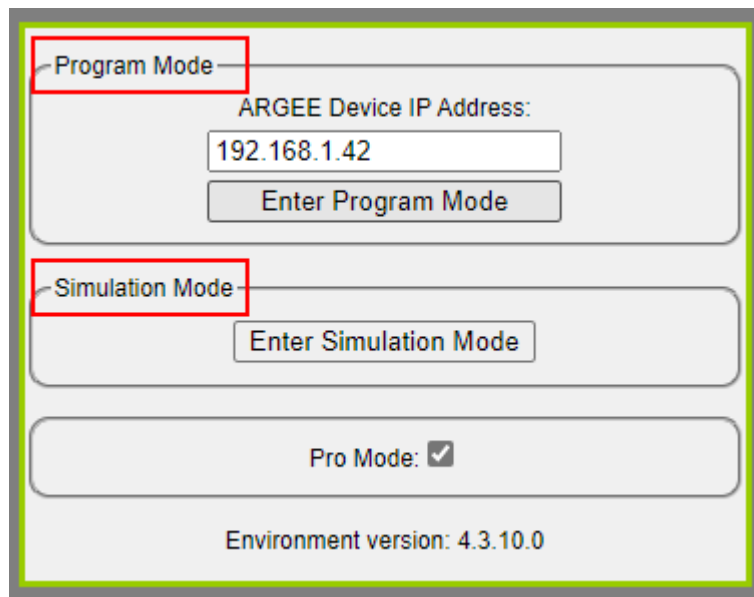


Abb. 2: Modus-Auswahl

Program Mode

Für die Verwendung des Program Mode sind mindestens ein PC und ein angeschlossenes ARGEE-fähiges Gerät erforderlich.

Im Program Mode können Programme für angeschlossene ARGEE-fähige Geräte geschrieben werden.

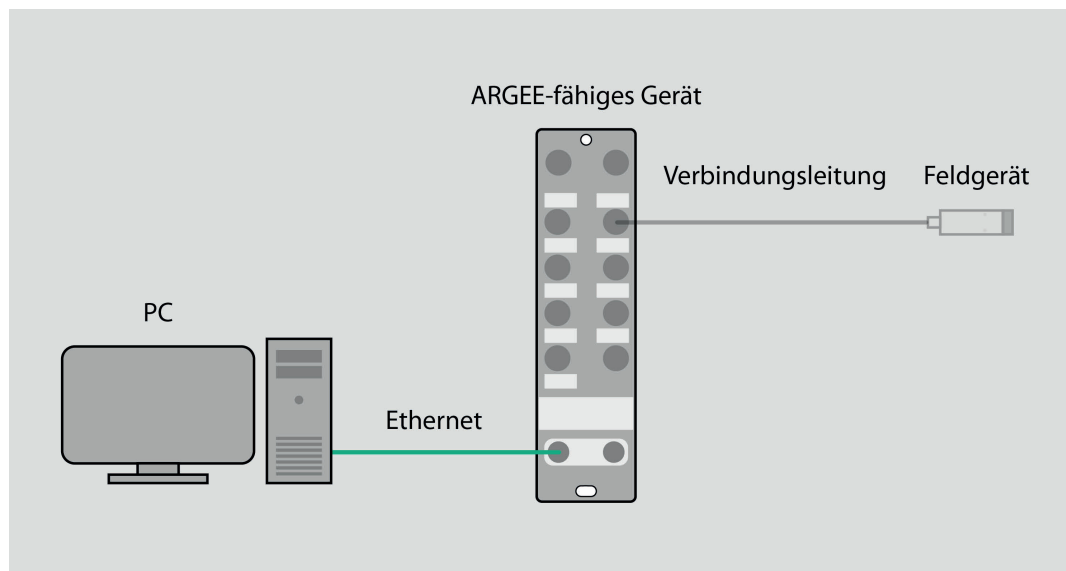


Abb. 3: Program Mode Anwendungsbeispiel

Simulation Mode

Programme können im ARGEE Simulation Mode geschrieben werden, ohne dass ein ARGEE-fähiges Gerät angeschlossen sein muss.

Ein angeschlossenes ARGEE-fähiges-Gerät wird nur simuliert.

Das gewünschte zu simulierende Gerät kann nach dem Start des Simulation Mode über eine Liste ausgewählt werden.

Die im Simulation-Mode geschriebenen ARGEE-Programme können anschließend auf ARGEE-fähige-Geräte geladen werden.



HINWEIS

Im Simulation-Mode werden nicht alle ARGEE-fähigen-Geräte aufgelistet.

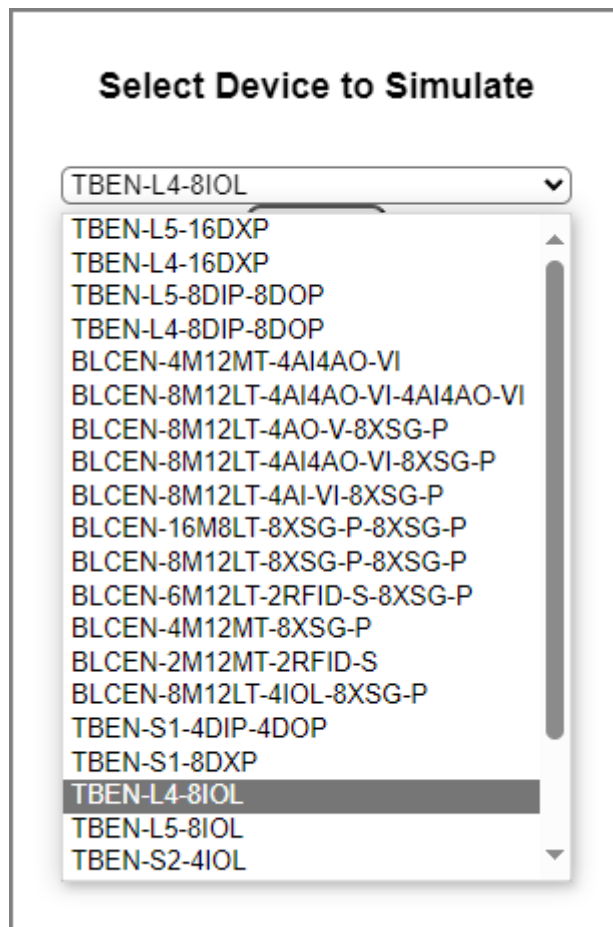


Abb. 4: Simulation Mode Geräteauswahl

5 ARGEE in Betrieb nehmen

Die ARGEE-Programmiersoftware steht unter www.turck.com zum kostenfreien Download zur Verfügung.

- ▶ Ordner entpacken.
- ▶ **Start ARGEE Programming Environment.html** in über den Browser öffnen.
- ▶ IP-Adresse des Geräts eingeben. (Der Simulation-Mode lässt sich auch ohne die Eingabe einer IP-Adresse öffnen.)

IP-Adressen für ARGEE-fähige Geräte können über die Turck Automation Suite (TAS) oder über die Website (Webserver) des jeweiligen Geräts eingestellt werden. Weitere Informationen zur Konfiguration der IP-Adresse finden Sie in der Betriebsanleitung des jeweiligen Geräts.

6 Übersicht ARGEE-Editor

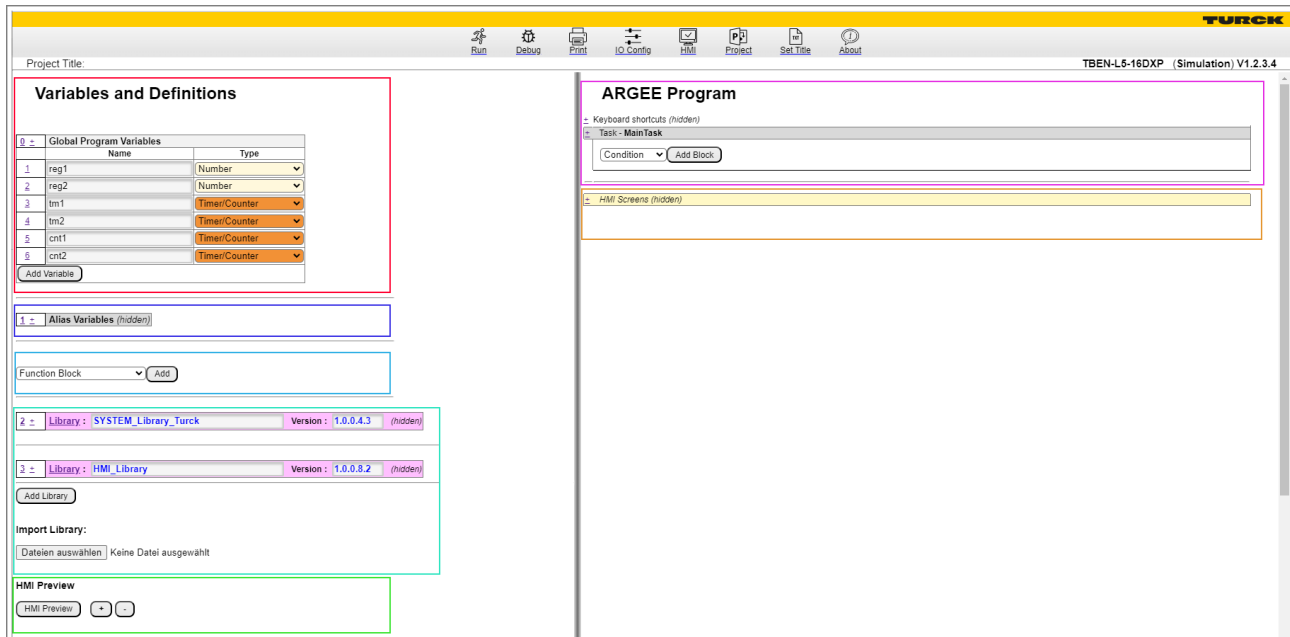


Abb. 5: ARGEE-Editor Hauptansicht

Funktion	Beschreibung
ARGEE Program	Unter dem Reiter ARGEE Program wird der Programm-Code geschrieben.
Program Variables	Program Variables dienen als benannte Speicherorte für sich ändernde Informationen. Der Variablentyp kann über einen Klick auf den ausgewählten Type geändert werden.
Alias Variables	Mithilfe von Alias Variables können für I/O-Punkte und SPS-Variablen benutzerdefinierte Namen vergeben werden. Durch die Anwendung von Alias Variables kann Code verständlicher gestaltet werden. Ein Beispiel für die Erstellung von Alias Variables finden Sie auf S. [▶ 25].
Function Blocks, States, Additional Global Variables	<p>Function Blocks</p> <p>Function Blocks dienen dazu den Codierprozess schneller zu gestalten, den Code schneller zu debuggen oder den Umfang des Programmcodes zu reduzieren. Mithilfe von Funktionsblöcken kann Code wiederverwendet werden.</p> <p>States</p> <p>Ein State ist eine benannte Konstante im Programm.</p> <p>Additional Global Variables</p> <p>Über Additional Global Variables können zusätzliche globale Variablen angelegt werden. Eine genauere Beschreibung von Function Blocks, States und Additional Global Variables finden Sie auf S. [▶ 26].</p>
Libraries	Bibliotheken (Libraries) sind vorgefertigte Sammlungen von Function Blocks, States und Additional Global Variables, die direkt in ARGEE importiert und genutzt werden können. Libraries können auch selbst erstellt werden. In ARGEE 4 sind bereits die SYSTEM_Library_Turck und die HMI_Library enthalten. Weitere Bibliotheken (Libraries) finden Sie auf www.turck.com .
HMI-Preview	Mithilfe von ARGEE können HMI-Screens (Human Machine Interface Screens) programmiert werden. HMI-Screens stellen das User-Interface des Programms dar. Über die HMI-Preview kann ein programmierter HMI-Screen bereits vor Programmstart angezeigt werden.
HMI Screens	Unter dem Reiter HMI Screens wird der Code für die Parametrierung der HMI-Screens geschrieben.

6.1 Projekt einrichten

6.1.1 Projekt-Tab

Mit einem Klick auf **Project** öffnet sich der Projekt-Tab.

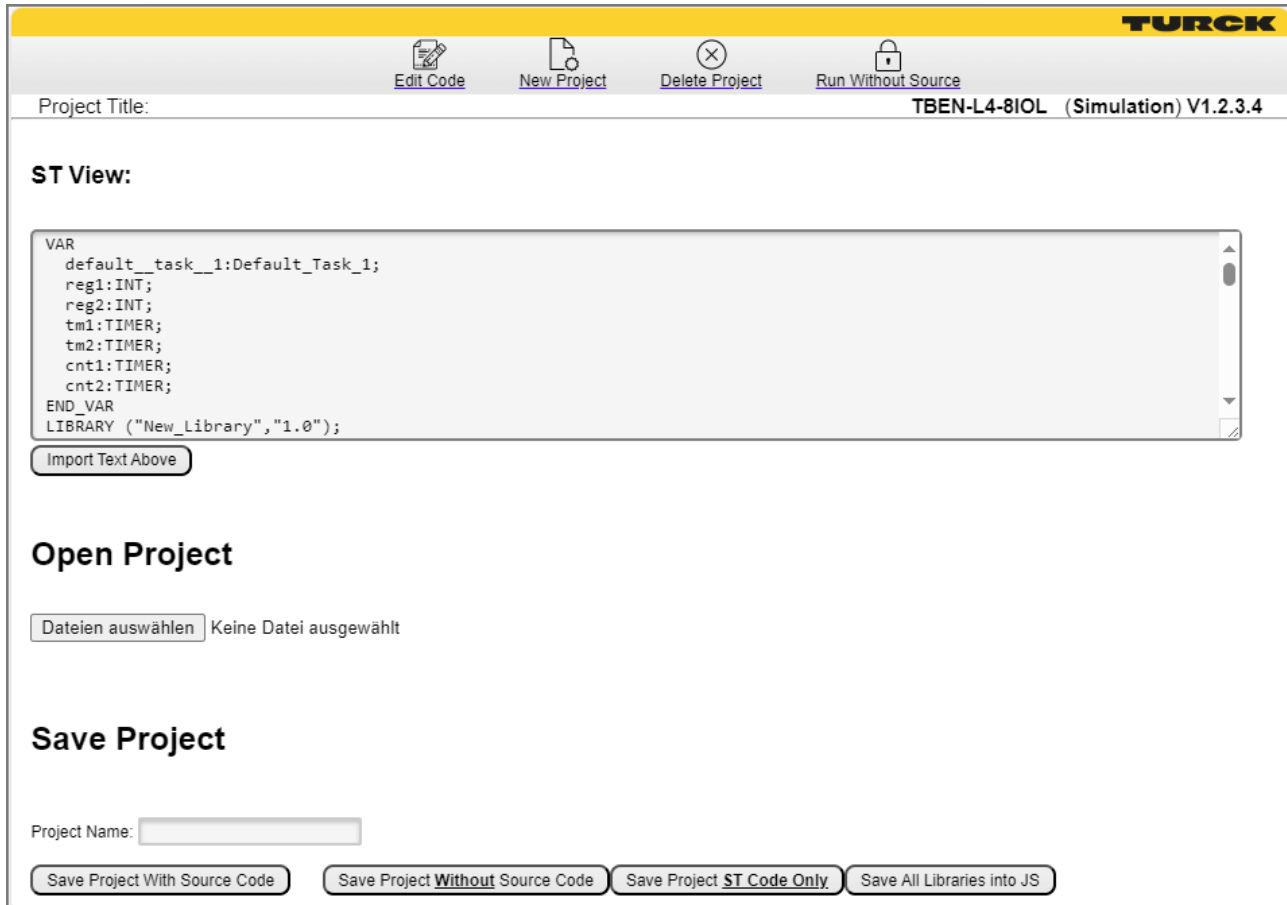


Abb. 6: Projekt-Tab

Neues Projekt anlegen

Beim erstmaligen öffnen von ARGEE wird automatisch ein leeres Projekt erstellt.

Neues Projekt erstellen

- ▶ In der Kopfzeile auf **Project** klicken.
- ▶ In der Kopfzeile auf **New Project** klicken.
- ▶ **OK** klicken.

Projekt löschen

- ▶ In der Kopfzeile auf **Project** klicken.
- ▶ In der Kopfzeile auf **Delete Project** klicken.
- ▶ **OK** klicken.

6.1.2 Projekte öffnen

Datei importieren

- ▶ In der Kopfzeile auf **Project** klicken.
- ▶ Unter der Überschrift **Open Project** auf **Dateien auswählen** klicken.
- ▶ Gewünschte Datei auswählen.
- ▶ **OK** klicken.

ST Text importieren

Diese Funktion dient dazu, Text zu importieren, der zuvor in einem Text-Editor erstellt oder aus einem anderen ARGEE-Projekt kopiert worden ist.

- ▶ Zu importierenden ST-Text kopieren.
- ▶ In der Kopfzeile auf **Project** klicken.
- ▶ ST-Text in das Fenster **ST View** einfügen.
- ▶ Unter dem Fenster **ST View** auf **Import Text Above** klicken.

Hinweis auf Programmversion und Gerätetyp

Bei Import eines bereits erstellten Programms in ARGEE erscheint folgende Benachrichtigung:

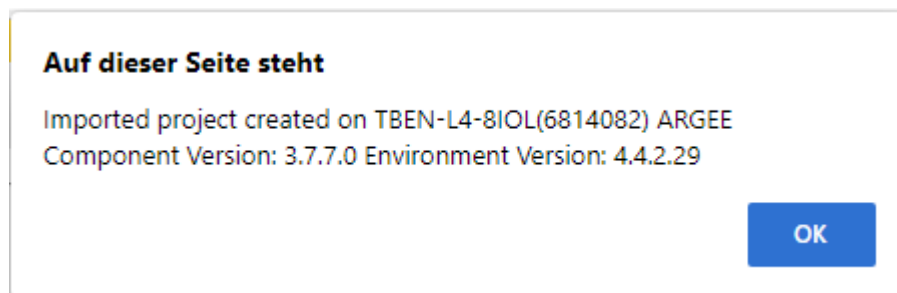


Abb. 7: Hinweis auf Programmversion und Gerätetyp

Die Benachrichtigung zeigt an, auf welchem Gerät das Programm erstellt wurde, die ARGEE Component Version und die Environment Version.

Bei Fehlerauftritt oder Komplikationen bei der Ausführung eines Programms können diese Informationen hilfreich sein.

Mithilfe dieser Informationen kann nachvollzogen werden, ob ARGEE-Engineering (Environment-Version) und ARGEE-Runtime (Component-Version) miteinander kompatibel sind.

6.1.3 Projekte speichern

Projekte und Bibliotheken können in verschiedenen Formaten exportiert werden.

- ▶ In der Kopfzeile auf **Project** klicken, um die Speicherfunktionen aufzurufen.

Speicheroption	Beschreibung
Save Project With Source Code	Das Projekt wird mit dem Quellcode gespeichert und kann wieder geöffnet und bearbeitet werden. Das Projekt kann nur auf ARGEE-fähige Geräte geladen werden. Das Projekt muss einmal ausgeführt werden.
Save Project Without Source Code	Das Projekt wird ohne den Quellcode gespeichert und kann nicht mehr bearbeitet werden. Das Projekt kann nur auf ARGEE-fähige Geräte geladen werden. Das Projekt muss einmal ausgeführt werden.
Save Project ST Code Only	Das Projekt wird als ST-Code abgespeichert.
Save All Libraries into JS	Speichert alle Bibliotheken als JavaScript-Datei. Die Datei kann dazu verwendet werden, dass ARGEE immer mit den abgespeicherten Bibliotheken startet, ohne dass die Bibliotheken jedes Mal neu importiert werden müssen. Dazu muss die glob_libs.js -Datei im Downloadverzeichnis von ARGEE ersetzt werden: Datei ersetzen: <ul style="list-style-type: none">▶ Javascript-Datei in glob_libs.js umbenennen und speichern.▶ ARGEE-Downloadverzeichnis aufrufen (...\\ARGEE V3\\ARGEE IDE v4.3.10.0\\internal).▶ Die vorhandene glob_libs.js im Verzeichnis mit der neu erstellten JavaScript-Datei ersetzen.▶ ARGEE über die Start ARGEE Programming Environment.html-Datei starten.

Speicherplatz für ARGEE-Projekte und ARGEE-Programme

Der bereitgestellte Speicherplatz für ARGEE-Programme ist abhängig vom jeweiligen Gerätetyp. Der verfügbare Speicher wird vom Compiler angegeben, wenn das Projekt erstellt wird.

Gerätetyp	Speicherplatz für ARGEE-Programme (Program Memory)
ARGEE-fähige-Geräte mit IO-Link und RFID	32 kB
FEN20-16DXP	25 kB
Alle anderen ARGEE-fähigen Geräte	42 kB

Der bereitgestellte Speicherplatz für ARGEE-Projekte (Project Memory) liegt für alle ARGEE-fähigen-Geräte bei 260 kB.

6.1.4 Programmcode als PDF anzeigen oder drucken

Über einen Klick auf **Print** in der Kopfzeile wird eine PDF des Programmcodes erstellt.

6.1.5 Projekt benennen

Über einen Klick auf **Set Title** in der Kopfzeile, kann das Projekt benannt oder umbenannt werden.

Der hier vergebene Titel wird im Webserver und in TAS angezeigt.



HINWEIS

Der vergebene Titel ist nicht der Titel der Projekt-Datei.

6.1.6 IO-Block konfigurieren

Über einen Klick auf **IO Config** in der Kopfzeile können die IO-Link-Parameter (IO-Link, RFID, Analog) des angeschlossenen oder simulierten IO-Blocks manuell definiert werden.

Die in der IO Config eingestellten Parameterdaten werden immer als erste Bezugsadresse verwendet.

Die Parameterkonfiguration ist Teil des ARGEE-Projekts. Im laufenden Betrieb geänderte Parameter (im Webserver oder ARGEE-Programmcode) werden beim Neustart des ARGEE-fähigen-Geräts auf den im ARGEE-Projekt definierten Wert zurückgesetzt.

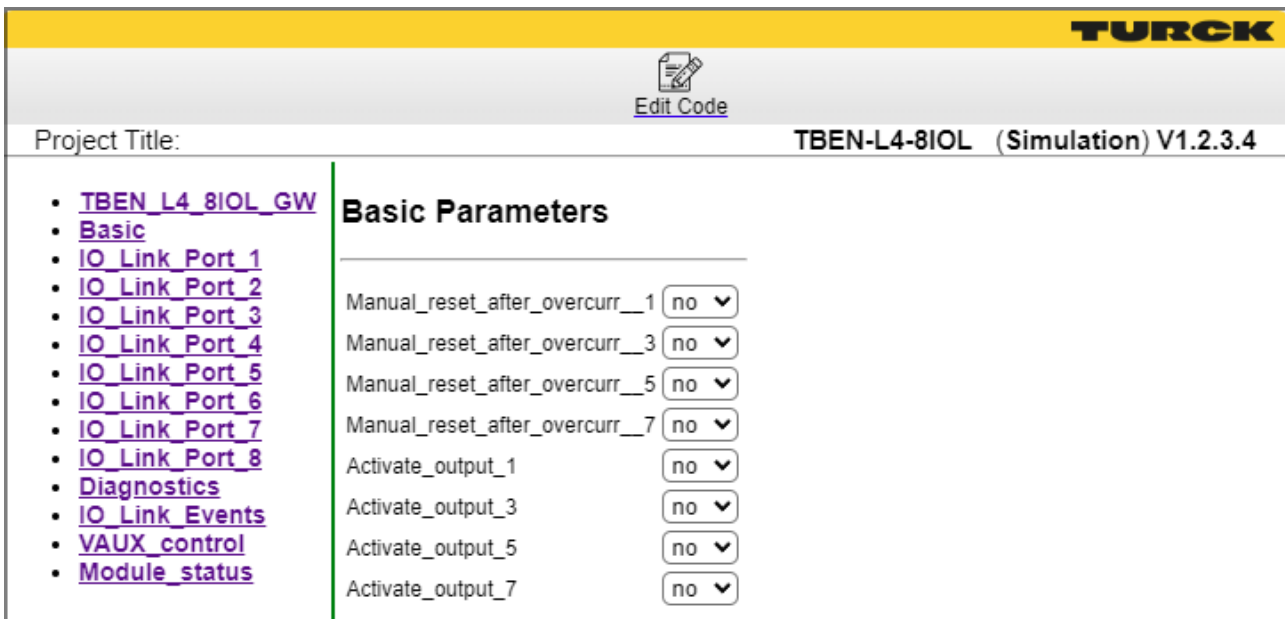


Abb. 8: IO-Config Menü

6.1.7 Informationen über ARGEE-Version

Über einen Klick auf **About** in der Kopfzeile, können die aktuelle IDE-Version und die aktuelle ARGEE-Runtime-Version eingesehen werden.

6.2 In der Projektumgebung navigieren

6.2.1 Kontextmenüs in ARGEE

In ARGEE können Kontextmenüs aufgeblendet werden, die weitere Funktionen in den jeweiligen Menüabschnitten enthalten.

Kontextmenü öffnen

- ▶ Auf die Zahl links neben dem jeweiligen Text-Eingabefeld klicken.

Kontextmenüs

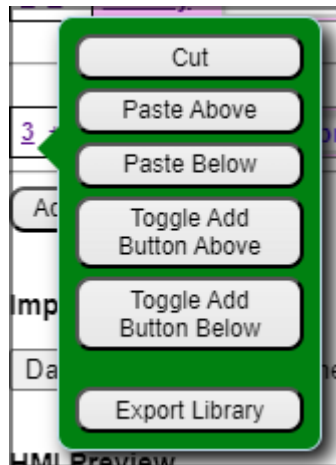


Abb. 9: Kontextmenü Libraries

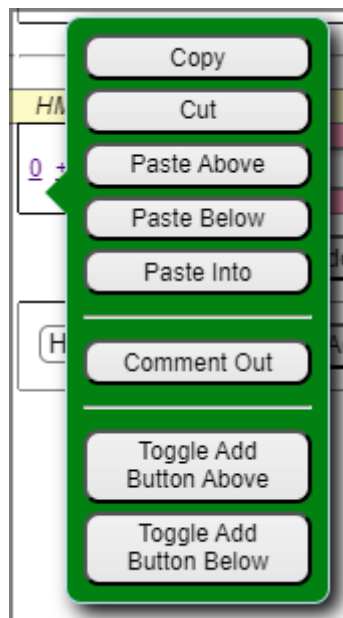


Abb. 10: Kontextmenü ARGEE Program

Funktion	Beschreibung
Copy	Kopiert die Variable in die Zwischenablage.
Cut	Schneidet die Variable aus.
Paste Above	Fügt die kopierte bzw. ausgeschnittene Variable oberhalb der markierten Position ein.
Paste Below	Fügt die kopierte bzw. ausgeschnittene Variable unterhalb der markierten Position ein.
Toggle Add Button Above	Fügt die Add Variable -Schaltfläche oberhalb der markierten Position hinzu, wenn zuvor keine Add Variable -Schaltfläche vorhanden war. Entfernt die Add Variable -Schaltfläche oberhalb der markierten Position, wenn zuvor eine Add Variable -Schaltfläche vorhanden war.
Toggle Add Button Below	Fügt die Add Variable -Schaltfläche unterhalb der markierten Position hinzu, wenn zuvor keine Add Variable -Schaltfläche vorhanden war. Entfernt die Add Variable -Schaltfläche unterhalb der markierten Position, wenn zuvor eine Add Variable -Schaltfläche vorhanden war.
Comment	Fügt ein Kommentarfeld oberhalb der Programmvariable ein.
Export Library	Exportiert die Bibliothek.
Comment Out	Wandelt das ausgewählte Statement in ein Kommentarfeld um. Kommentare werden bei Start des Programms nicht mit kompiliert.

6.2.2 Bedienungshilfen

ARGEE bietet zur erleichterten Bedienung der Software folgende Bedienungshilfen:

- Tastenkombinationen (Shortcuts)
- Kontextmenüs in ARGEE
- Drag-and-Select-Features

Tastenkombinationen (Shortcuts)

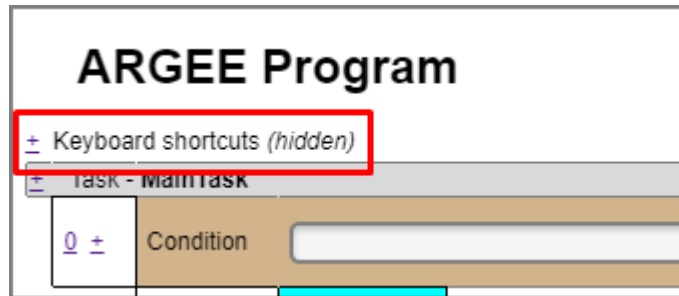


Abb. 11: Shortcuts (hidden)

ARGEE verfügt über eine Liste an Tastenkombinationen (Shortcuts).

Tastenkombination	Beschreibung	Voraussetzungen
Strg + Q	Öffnet eine Liste der Programmvariablen.	Mit dem Cursor in eine Zeile des Programmcodes klicken.
Strg + L	Öffnet eine Liste der lokalen Variablen für Funktionen, Tasks, Funktionsblöcke und Prozeduren.	Mit dem Cursor in das Feld des zu bearbeitenden Funktionsblocks klicken.
Strg + I	Öffnet eine Liste der I/O-Variablen.	Mit dem Cursor in eine Zeile des Programmcodes klicken.
Strg + F	Öffnet eine Liste der eingebauten Funktionen („Built-in Functions“) des Feldes, in welches zuvor mit dem Cursor geklickt wurde.	Mit dem Cursor in eine Zeile des Programmcodes klicken.
Strg + S	Öffnet eine Liste der Zustands-Namen (State Names).	
Strg + Pfeiltaste unten	Öffnet das Add Below -Menü. Im Add Below -Menü kann zwischen verschiedenen Tasks ausgewählt werden, die über der markierten Programmzeile eingefügt werden sollen.	Mit dem Cursor in eine Zeile des Programmcodes klicken.
Strg + Pfeiltaste oben	Öffnet das Add Above -Menü. Im Add Above -Menü kann zwischen verschiedenen Tasks ausgewählt werden, die über der markierten Programmzeile eingefügt werden sollen.	Mit dem Cursor in eine Zeile des Programmcodes klicken.
Strg + X	Schneidet das ausgewählte Statement aus.	
Strg + C	Kopiert das ausgewählte Statement in die Zwischenablage.	
Strg + Z	Macht die letzte Änderung rückgängig (bis zu 32 Aktionen).	
Strg + Y	Stellt die zuletzt rückgängig gemachte Änderung wieder her (bis zu 32 Aktionen).	
Strg + D	Wandelt Statements in Kommentare um. Kommentare werden bei Start des Programms nicht kompiliert.	Mit dem Cursor in das nummerierte Feld links neben „Assignment“ klicken.

Tastenkombination	Beschreibung	Voraussetzungen
Strg + Shift + D	Wandelt Kommentare in Statements um. Statements werden bei Start des Programms kompiliert.	Mit dem Cursor in das nummerierte Feld links neben „Assignment“ klicken.
F1	Öffnet eine Übersicht der Tastenkombinationen.	
F2	Öffnet eine „read-only“-Ansicht des Projekts.	
F3	Öffnet die IO-Description . In der IO-Description werden die IO-Daten der angeschlossenen Geräte tabellarisch dargestellt.	
Doppelklick	Schaltet die Add Block -Zeile oberhalb der Zeile im Programmcode um. Dient dazu, den Code übersichtlicher zu gestalten. Siehe Beispiel Doppelklick	Doppelklick muss in das weiße Feld links neben der Zeile im Programmcode ausgeführt werden.
Strg + Doppelklick	Schaltet die Add Block -Zeile unterhalb der Zeile im Programmcode um. Dient dazu, den Code übersichtlicher zu gestalten. Siehe Beispiel Doppelklick	Doppelklick muss in das weiße Feld links neben der Zeile im Programmcode ausgeführt werden.

Beispiel Doppelklick

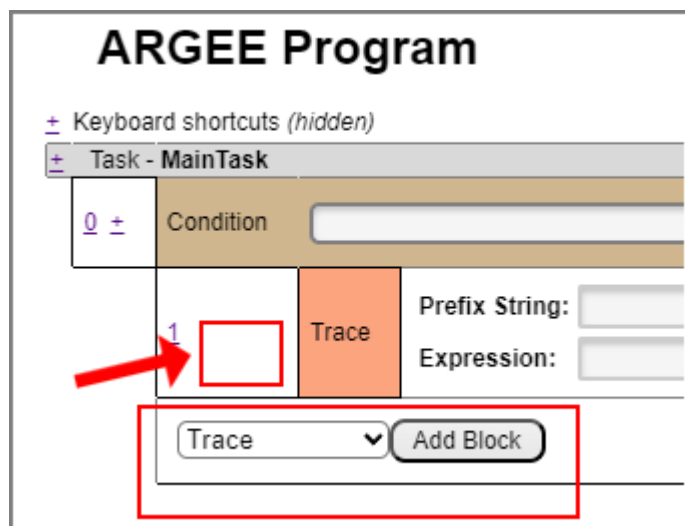


Abb. 12: Beispiel Strg + Doppelklick Vorher

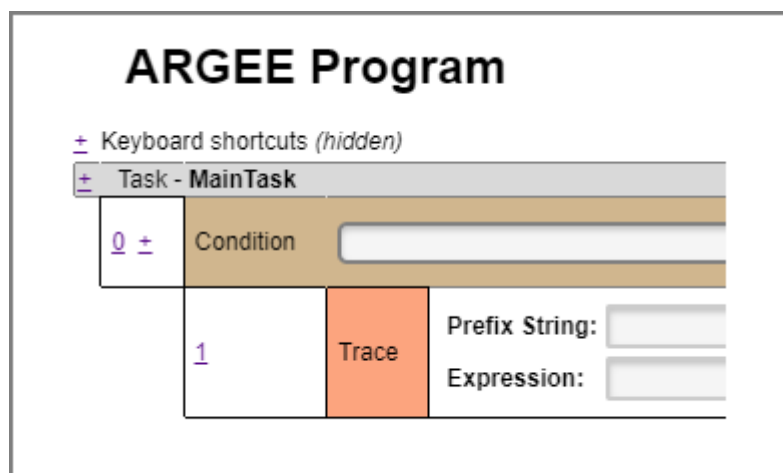


Abb. 13: Beispiel Strg + Doppelklick Nachher

Drag-and-Select-Features

In ARGEE ist es möglich, mehrere Programm-Code-Zeilen per Drag-and-Select zu markieren und anschließend zu kopieren, auszuschneiden und einzufügen.



HINWEIS

Die Aktionen (Kopieren, Ausschneiden und Einfügen) können nicht über die Kontextmenüs ausgeführt werden. Das Kopieren, Ausschneiden und Einfügen der markierten Code-Abschnitte funktioniert nur mit den Tastenkombinationen Strg + C = Kopieren, Strg + X = Ausschneiden, Strg + V = Einfügen.

Zeilen auswählen:

- ▶ Mit dem Mauszeiger einen Links-Klick in den Weißraum des ersten zu markierenden Condition-Blocks ausführen und gedrückt halten.
- ▶ Den Mauszeiger bis zum letzten zu markierenden Condition-Block herunterziehen und Linke Maustaste loslassen.
- ▶ Per Tastenkombination gewünschte Aktion ausführen (Strg + C = Kopieren, Strg + X = Ausschneiden, Strg + V = Einfügen).

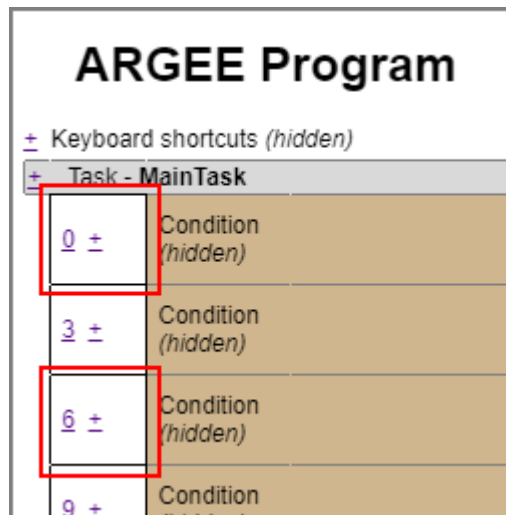


Abb. 14: Beispiel Drag-and-Select vorher



Abb. 15: Beispiel Drag-and-Select nachher

6.3 Programm einrichten

6.3.1 Program Variables

Program Variables dienen als benannte Speicherorte für sich ändernde Informationen.

Der Name einer globalen Programmvariable kann nicht doppelt vergeben werden.

Der Variablentyp kann über einen Klick auf den ausgewählten Type geändert werden.

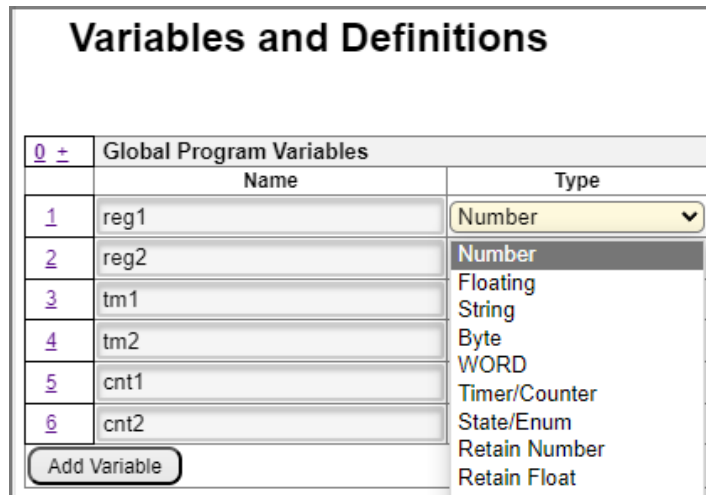


Abb. 16: ARGEE Variablen-Typ



HINWEIS

Die Anzahl der Retain-Variablen (Retain Number und Retain Float) ist begrenzt. Die Lebensdauer des Retain-Speichers ist auf 100.000 Schreibvorgänge begrenzt.

Variablentyp	Beschreibung
Number	Speichert einen ganzzahligen Wert (Integer) zwischen -2.147.483.658 und 2.147.483.657 (4 Byte vorzeichenbehaftete Ganzzahl) ab.
Floating	Speichert eine Gleitkommazahl ab.
String	Speichert ganzzahlige Werte (Integer) und/oder Buchstaben ab.
Byte	Ein Byte ohne Vorzeichen. Speichert Ganzzahlen von 0 bis 255 oder Hex-Werte von 0x00 bis 0xFF.
Word	Zwei Bytes ohne Vorzeichen. Speichert ganze Zahlen von 0 bis 65535, oder Hex-Werte von 0x0000 bis 0xFFFF.
Timer/Counter	Timer/Counter -Register können einen Wert zwischen -2.147.483.658 und 2.147.483.657 speichern.
State/Enum	State/Enum (Aufzählung) dient dazu eine Zustandsvariable zu erstellen. Zustandsvariablen werden in Zustandsautomaten (state machines) verwendet.
Retain Number	Speichert ganze Zahlen zwischen -2.147.483.658 und 2.147.483.657 während eines Stromzyklus ab. Der Wert wird etwa alle 2 Minuten synchronisiert.
Retain Float	Speichert eine Gleitkommazahl über einen Stromzyklus hinweg ab. Der Wert wird etwa alle 2 Minuten synchronisiert.

Weitere vordefinierte Variablen können mithilfe von Bibliotheken (Libraries) importiert werden. Siehe [▶ 32].

6.3.2 Variablen initialisieren

Über einen Klick auf die Zahl neben der zu initialisierenden globalen Variable öffnet sich ein Kontext-Menü. Mit einem Klick auf **Init** öffnet sich ein Dialogfeld, in dem der Wert für das Register einer Programmvariable voreingestellt werden kann.

6.3.3 Array erstellen

Über einen Klick auf die Zahl neben der globalen Variable öffnet sich ein Kontext-Menü. Mit einem Klick auf **Make it Array** wird die Variable in ein Array umgewandelt.

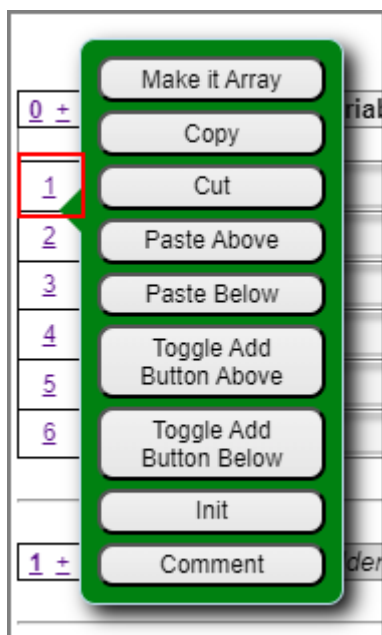


Abb. 17: Variablen initialisieren

Array initialisieren

Über einen Klick auf die Zahl neben des zu initialisierenden Arrays öffnet sich ein Kontext-Menü. Mit einem Klick auf **Init** öffnet sich eine weitere Textzeile, in der das Array initialisiert werden kann.

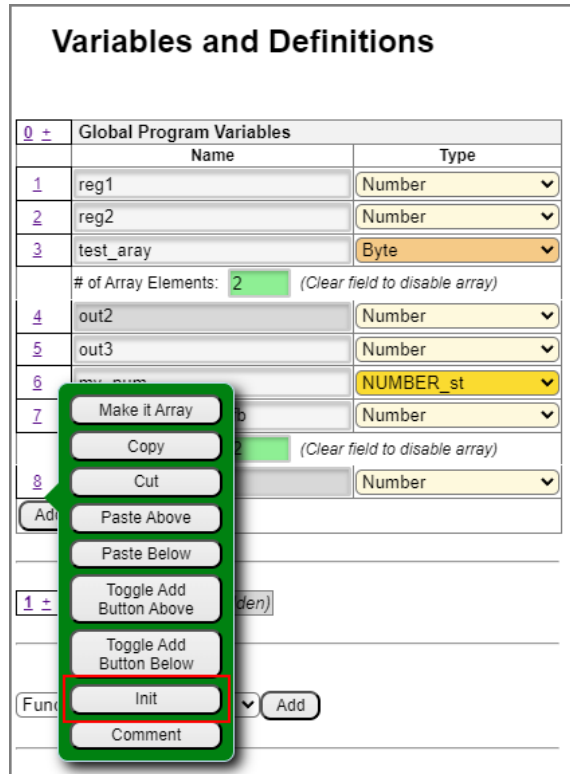


Abb. 18: Array Init

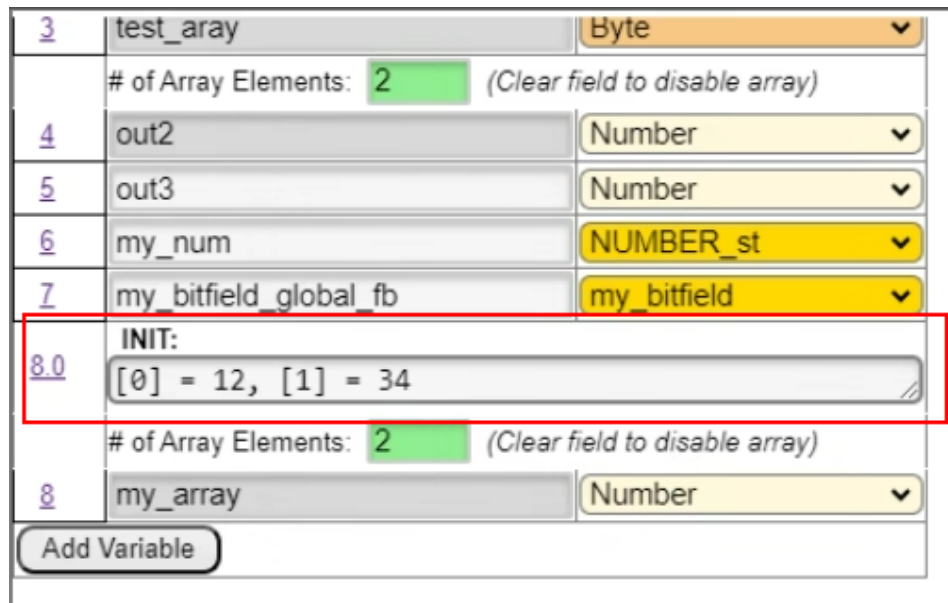


Abb. 19: Array Init Textfeld

6.3.4 Alias Variables erstellen

Mithilfe von Alias Variables können für I/O-Punkte und globale Variablen benutzerdefinierte Namen vergeben werden.

Durch die Anwendung von Alias-Variablen kann Code verständlicher gestaltet werden.

Aus dem I/O-Punkt `IO_Basic_Input_Input_value_4` wird hier zum Beispiel `car_sensor`.

1 ± Alias Variables		
	Name	IO Point
0	Friendly_IO_Point_Name	IO_Point
1	car_sensor	IO_Basic_Input_Input_value_4
2	greenlight	IO_Basic_Output_Output_value_6
3	PLC_in	IO_PLC_TO_ARGEE_Word4
Add Variable		

Abb. 20: Alias Variables Beispiel



HINWEIS

Über Alias Variables referenzierte I/O-Punkte werden im HMI-View nicht erkannt (I/O-Punkte können nicht an anderer Stelle im Projekt dupliziert werden).

Fehlerhafte Kompilierung/Darstellung des programmierten HMI

- ▶ Globale Variable anlegen.
- ▶ Alias Variable mit I/O-Punkt anlegen.
- ▶ Assignment im ARGEE Program erstellen (Destination: „Name der globalen Programmvariable“, Expression: „Name der Alias Variable“).
- ▶ Im HMI-View den Namen der globalen Programmvariable verwenden.

6.3.5 Function Blocks, States und Additional Global Variables erstellen

Über ein Drop-down-Menü kann ausgewählt werden, ob ein Function Block, ein State oder eine Additional Global Variable angelegt werden soll.

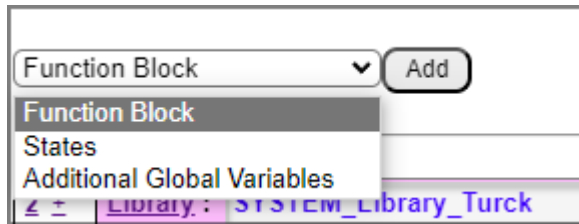


Abb. 21: Function Block, States, Additional Global Variable anlegen

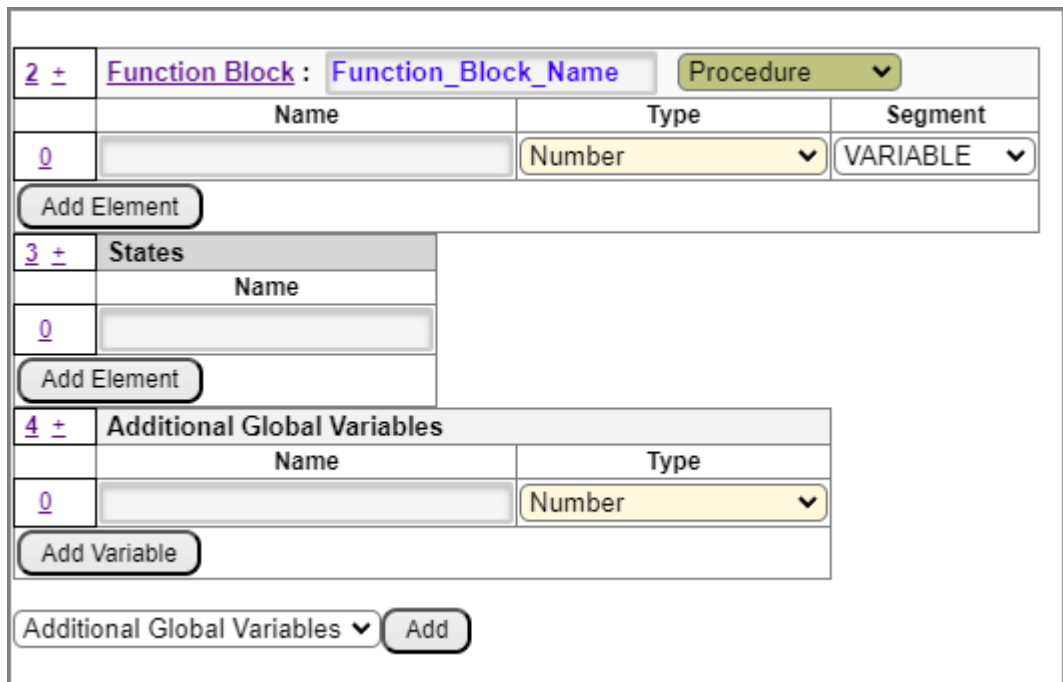


Abb. 22: Function Block, State und Additional Global Variable in ARGEE

Function Block

Function Blocks dienen dazu, den Codierprozess zu verschnellern und den Code übersichtlicher zu gestalten. Eine detaillierte Beschreibung der Function Block Types finden Sie in der Tabelle auf der nächsten Seite.

State

Ein State ist eine benannte Konstante. States können einen benannten Wert erzwingen.

Additional Global Variables

Über Additional Global Variables können zusätzliche globale Variablen erstellt werden. Additional Global Variables können eigenständige Bibliotheken enthalten.

Function Block Type	Beschreibung
Procedure	<ul style="list-style-type: none"> ■ Eine Procedure ist ähnlich aufgebaut wie ein Function Block, muss aber nicht instanziiert werden. ■ Komplexe Datenstrukturen und Arrays können als Argumente an Procedures übergeben werden. ■ Standard-Datentypen, die mit Byte/Num nummeriert sind, können an Prozeduren als reine Eingabeargumente übergeben werden.
Structure	<ul style="list-style-type: none"> ■ Eine Structure ist ein benutzerdefinierter komplexer Datentyp, der sich aus einem oder mehreren Datentypen (Variablen) zusammensetzt. ■ Structures können dazu verwendet werden verwandte Variablen zusammenzufassen. ■ Structures werden wie Standardvariablen deklariert. <p>Beispiel: Die Structure eines Würfels könnte Elemente der Breite, Höhe und Länge enthalten. Auf die Elemente kann mit der Dezimalpunkt-Schreibweise „.“ direkt zugegriffen werden, z. B. cube.length = 10.</p>
Function	<ul style="list-style-type: none"> ■ Functions dienen zur Reduzierung des geschriebenen Codes und zur Verbesserung der Lesbarkeit des Codes, insbesondere wenn sie in Expressions verwendet werden, da das Ergebnis der Function durch die Expression bewertet werden kann. ■ Functions können nur lesend auf komplexe Datentypen zugreifen. ■ Functions müssen eine einzige Rückgabeveriable namens „Result“ enthalten. ■ Functions dürfen nicht in globale Variablen schreiben.
Task	<ul style="list-style-type: none"> ■ Tasks können verwendet werden, um mehrere Programmabschnitte innerhalb des Projekts zu erstellen, die parallel ausgeführt werden. ■ Eine Task muss instanziiert werden, muss aber nicht aufgerufen werden (unabhängiger Programmabschnitt). ■ Tasks können durch die Verwendung von Blocking Statements zur Trennung von Programmblöcken verwendet werden. ■ Blocking Statements werden von einer Task verwendet, um auf das Eintreten einer bestimmten Bedingung zu warten, ohne die Ausführung anderer Tasks zu blockieren.
Bitfield	<ul style="list-style-type: none"> ■ Ein Bitfield ist eine Datenstruktur, mit der einzelne benannte Elemente von ≤ 32 Bit erstellt werden können. ■ Bitfields können sich über mehr als 32 Bits erstrecken, hierbei darf sich aber keine Variable zwischen zwei aufeinanderfolgenden 32-Bit-Grenzen innerhalb des Bitfields befinden. ■ Auf einzelne Elemente (Variablen) des Bitfields kann über bitfieldname.elementname zugegriffen werden. ■ Bitfields können zur Organisation von zusammenhängenden Datenelementen verwendet werden (z. B. IO-Link-Prozessdaten mit mehreren in den Roh-Bytes codierten Elementen). ■ Es können Strukturen erstellt werden, die mehrere Bitfields enthalten (z. B. Sensor mit komplexen Eingangs- und Ausgangsprozessdaten).
Function Block	<ul style="list-style-type: none"> ■ Ein Funktionsblock kann die gleichen Argumente wie eine Procedure haben. ■ Ein Function Block muss instanziiert werden, weshalb mehrere Instanzen desselben Funktionsblocks in einem Projekt existieren können.
HMI Block	<ul style="list-style-type: none"> ■ HMI-Block wird nur auf dem HMI-Block-Bildschirm verwendet. ■ Ein HMI-Block formatiert das Layout des Bildschirms, gibt aber keine Daten auf dem Bildschirm aus. <p>Beispiele für HMI-Blöcke sind die HMI-Tabellen, Zeilenblöcke und Spaltenblöcke.</p>
HMI Control	<ul style="list-style-type: none"> ■ HMI Control wird nur auf dem HMI-Block-Bildschirm verwendet. ■ Ein HMI-Control ist ein Element, das Daten anzeigt oder eine Eingabe von der HMI erhält. <p>Beispiele für HMI-Control sind HMI_Button, HMI_Display_Value, HMI_Dropdown.</p>

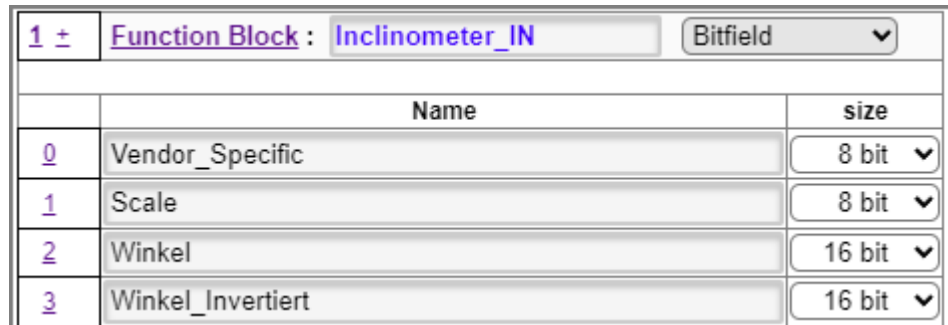
6.3.6 Bitfields und I/O-Mapping

Bitfields

ARGEE unterstützt Bitfields. Mithilfe von Bitfields können in ARGEE einzelne Bits in den Prozessdaten von angeschlossenen Geräten ausgelesen werden.

Im Programmcode können die Bits direkt über die Dezimalpunktschreibweise adressiert werden.

Bitfields werden als Function Blocks angelegt.



Function Block : Inclinometer_IN Bitfield		
	Name	size
0	Vendor_Specific	8 bit
1	Scale	8 bit
2	Winkel	16 bit
3	Winkel_Invertiert	16 bit

Abb. 23: Bitfields in ARGEE

In diesem Beispiel wurden die Prozessdaten eines Neigungssensors in ARGEE übertragen. Nach der Übertragung kann innerhalb der ARGEE-Programmierungsumgebung auf die Daten zugegriffen werden.

I/O-Mapping

Per I/O-Mapping kann auf die in den Bitfields angelegten Prozessdaten zugegriffen werden. Der Vorteil des I/O-Mappings ist, dass die Verknüpfung des Codes und der Bitfields nur einmalig eingerichtet werden muss.

Nach der Einrichtung läuft das I/O-Mapping im Hintergrund.

Weitere Eigenschaften des I/O-Mappings:

- Eine geringere Fehlerquote beim Schreiben des Codes, da die Daten nur einmalig eingerichtet und nicht erneut im Code kopiert oder referenziert werden müssen.
- Der Code wird übersichtlicher, da weniger Zeilen an Code geschrieben werden müssen.

Bitfields einrichten

Im folgenden Abschnitt wird anhand eines angeschlossenen Neigungssensors die Funktionsweise von Bitfields und des I/O-Mappings erläutert.

- ▶ IO-Link-Device an ARGEE-fähiges Gerät anschließen.
- ▶ Prozessdaten über den Webserver oder die Turck Automation Suite (TAS) aufrufen.
- ▶ Prozessdaten bitweise in Function Blocks übertragen.

2 ±		Function Block : <input type="text" value="Inclinometer_IN"/>	Bitfield	▼
		Name	size	
0		Vendor_Specific	8 bit	▼
1		Scale	8 bit	▼
2		Angle	16 bit	▼
3		Angle_Inverted	16 bit	▼
Add Element				
3 ±		Function Block : <input type="text" value="Inclinometer_Out"/>	Bitfield	▼
		Name	size	
0		Output	8 bit	▼
Add Element				
4 ±		Function Block : <input type="text" value="Inclinometer"/>	Structure	▼
		Name	Type	Segment
0		Input	Inclinometer_IN	VARIABLE
1		Output	Inclinometer_Out	VARIABLE
Add Element				
5 ±		Additional Global Variables		
		Name	Type	
0		Inclinometer_1	Inclinometer	
Add Variable				
Additional Global Variables ▼ Add				

Abb. 24: Bitfields Beispiel

I/O-Mapping einrichten

Im I/O-Mapping wird über eine If-Funktion auf die in den Bitfields hinterlegten Prozessdaten zugegriffen.

Procedure - Local_IO_Init()			
0	If		NOT IO_INIT
1	Assignment	Destination:	IO_INIT
		Expression:	1
2	Assignment	Destination:	IOMap_Started
		Expression:	0
3	Call	IOMAP_Start() IOMAP_Start()	
4	Call	IOMAP_Add(slot , sect , map_offset , mapped_arr) IOMAP_Add(2,IOMAP_SECT_INP,0,Inclinometer_1.Input)	
5	Call	IOMAP_Add(slot , sect , map_offset , mapped_arr) IOMAP_Add(2,IOMAP_SECT_outP,0,Inclinometer_1.Output)	
		Call	Add Block

Abb. 25: I/O-Mapping Beispiel

Zur Veranschaulichung des Codes kann über **F3** die I/O-Description des angeschlossenen Geräts geöffnet werden.

Hier können Parameter, Diagnosen, Eingänge und Ausgänge eingesehen werden. Der erste IO-Link-Port befindet sich in diesem Beispiel auf Slot 2.

Der Map-Offset ist der Bit-Offset innerhalb des Slots. Das erste Datenwort am Eingang von Port 1 hat den Offset 0, das nächste den Offset 16.

IO Description - Google Chrome

about:blank

Activate_output_ /	25	1	1 - yes
--------------------	----	---	---------

Slot 2 - IO_Link_Port_1

Input

Name	Bit Offset	Bit Length	Enum Values
Input_data_word_0	0	16	
Input_data_word_1	16	16	
Input_data_word_2	32	16	
Input_data_word_3	48	16	
Input_data_word_4	64	16	
Input_data_word_5	80	16	
Input_data_word_6	96	16	
Input_data_word_7	112	16	
Input_data_word_8	128	16	
Input_data_word_9	144	16	
Input_data_word_10	160	16	
Input_data_word_11	176	16	
Input_data_word_12	192	16	
Input_data_word_13	208	16	
Input_data_word_14	224	16	
Input_data_word_15	240	16	

Output

Name	Bit Offset	Bit Length	Enum Values
Output_data_word_0	0	16	
Output_data_word_1	16	16	
Output_data_word_2	32	16	
Output_data_word_3	48	16	

Abb. 26: ARGEE Map-Offset und Slot Nummer

Die Eingangsdaten, Ausgangsdaten oder Diagnosedaten, die gemappt werden können werden als Section in der I/O-Description definiert.

Die States sind in der Turck Library definiert:

21 ±	States
	Name
0	IO_MAP_SECT_INP
1	IO_MAP_SECT_OUTP
2	IO_MAP_SECT_DIAG

Abb. 27: ARGEE Section

6.3.7 Bibliotheken (Libraries) importieren

Bibliotheken (Libraries) sind Sammlungen von themenbasierten Function Blocks und States.

Bibliotheken können über die Turck Website heruntergeladen oder selbst erstellt werden.

In ARGEE 4 sind bereits die **SYSTEM_Library_Turck** und die **HMI_Library** enthalten.

Weitere Bibliotheken (Libraries) finden Sie auf www.turck.com.

Libraries importieren

- ▶ Bibliothek auf www.turck.com herunterladen.
- ▶ ZIP-Datei extrahieren.
- ▶ Bibliothek (Library) importieren.



Abb. 28: Bibliothek importieren

6.3.8 Program-Blocks

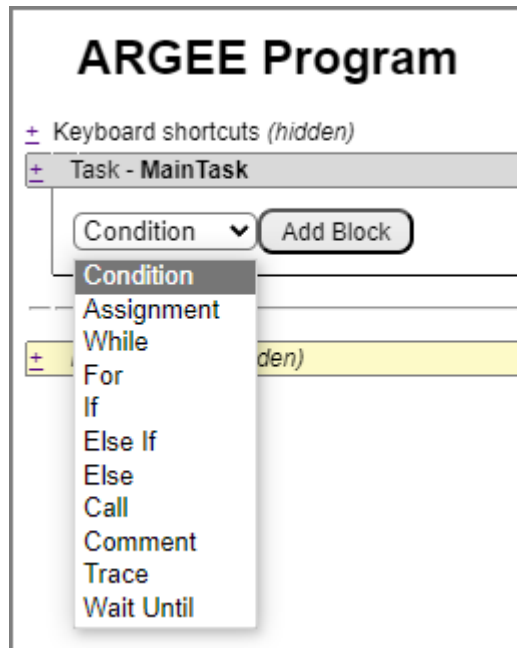


Abb. 29: Program-Blocks

Program-Block Type	Beschreibung
Condition	Condition-Blocks ähneln einem If-Statement und funktionieren wie ein Ladder rung. Wenn die Expression in der Condition wahr ist, werden alle Blöcke darin als wahr ausgeführt. Wenn die Condition falsch ist, werden alle Blöcke in der Condition als falsch ausgeführt. Innerhalb des Condition-Blocks werden zusätzliche Program-Blocks verwendet. [▶ 35]
Assignment	Über ein Assignment kann einer Variable ein Wert zugewiesen werden. Der Wert kann eine Konstante, eine komplexe Expression oder eine Function sein. Der zugewiesene Wert kann eine Variable, ein Output, ein Array oder ein Mitglied einer Structure sein.
While	While-Schleifen (While-loops) führen die Blöcke innerhalb der Schleife so lange aus, wie die Expression wahr bleibt.
For	For-Schleifen (For-loops) werden so lange ausgeführt, bis die Iterator-Variable der For-Schleife dem Endwert entspricht. Start- und Endwerte können Konstanten, Variablen oder Expressions sein.
If	If-Blocks führen die im Block enthaltenen Program-Blocks nur aus, wenn der Ausdruck des If-Statements wahr ist.
Else If	Else If-Blocks können direkt unter If-Blocks platziert werden. Wenn der ursprüngliche If-Block als falsch ausgeführt wird, können so zusätzliche Expressions geprüft werden.
Else	Else-Blocks können direkt unter If-Blocks oder Else-If-Blocks platziert werden. Program-Blocks innerhalb des Else-Blocks werden erst ausgeführt, wenn sowohl der If-Block als auch alle Else-If-Blocks falsch ausgeführt werden.

Program-Block Type	Beschreibung
Call	Call-Blocks führen Functions, Procedures oder Function-Blocks aus. Dabei kann es sich um integrierte oder benutzerdefinierte Functions handeln. Mit der Tastenkombination Strg+Q werden die verfügbaren Funktionen angezeigt. Mögliche Arguments für eine Function werden oberhalb der Expression-Box in einer kurzen Hilfezeile angezeigt.
Comment	Comment-Blocks können verwendet werden, um Kommentare über oder unter anderen Program-Blocks einzufügen. Kommentare können innerhalb der Program-Blocks unterhalb der If-Blocks, Else If-Blocks und Else-Blocks verwendet werden. Wenn Comment-Blocks zur Trennung von If-Blocks, Else-If-Blocks oder Else-Blocks eingesetzt werden, führt das zu einem Kompilierfehler.
Trace	Trace-Statements können zur Fehlersuche verwendet werden. Wenn ein Programm einen Trace-Block erreicht, wird der Wert der Expression an die im Trace-Puffer gespeicherte Präfix-Zeichenkette mit einem Zeitstempel angehängt. Der ringförmige Trace-Puffer kann die letzten 50 Traces speichern, die auf dem Debug-Bildschirm angezeigt werden.
Wait Until	Wait Until ist ein blockierendes Statement, das die Ausführung der Task anhält, bis das Statement in der Expression wahr wird. Andere Tasks (Siehe Task [▶ 27]) im Projekt werden wie gewohnt ausgeführt.

Folgende Program-Blocks sind nur innerhalb des Condition-Blocks wählbar:

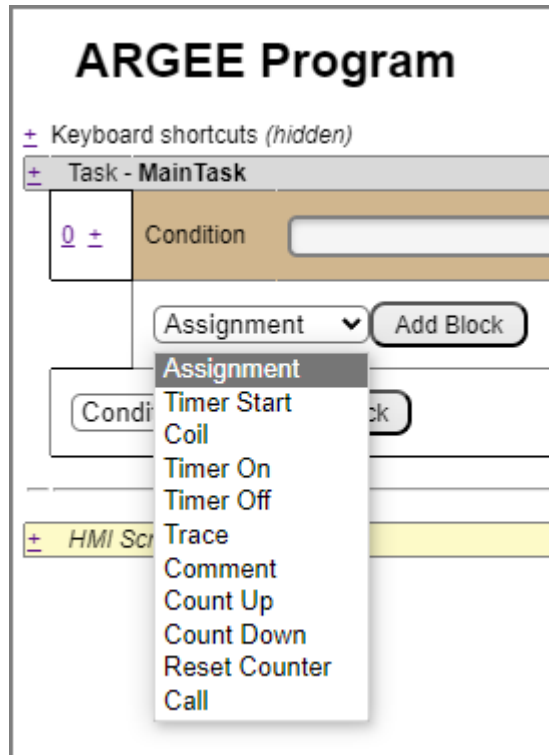


Abb. 30: Program-Blocks Condition

Program-Block Type	Beschreibung
Timer Start	Timer Start startet einen Timer, wenn der übergeordnete Condition-Block wahr ist. Solange die Condition wahr bleibt, startet der Timer in den nachfolgenden Programmzyklen wieder bei 0.
Coil	Coil kann an einen physischen Ausgang oder eine Programmvariable gebunden werden. Coil bleibt aktiviert, solange der übergeordnete Condition-Block wahr ist. Wenn die Condition falsch ist, wird der Coil deaktiviert.
Timer On	Timer On ist ein On-Delay-Timer. Der Timer startet, wenn die Condition wahr ist. Der Timer zählt so lange hoch, wie die Condition wahr bleibt. Wenn die Condition falsch wird, wird der Timer zurückgesetzt.
Timer Off	Timer Off ist ein Off-Delay-Timer. Der Timer startet, wenn die Condition falsch ist. Der Timer zählt so lange hoch, wie die Condition falsch bleibt. Wenn die Bedingung wahr wird, wird der Timer zurückgesetzt.
Count Up	Count Up erhöht stufenweise den Zählerwert bei einer steigenden Flanke der Condition.
Count Down	Count Down verringert stufenweise den Zählerwert bei einer steigenden Flanke der Condition.
Reset Counter	Reset Counter setzt den Zählerwert bei einer steigenden Flanke der Condition auf 0 zurück.

6.4 Programm ausführen

6.4.1 Programm kompilieren

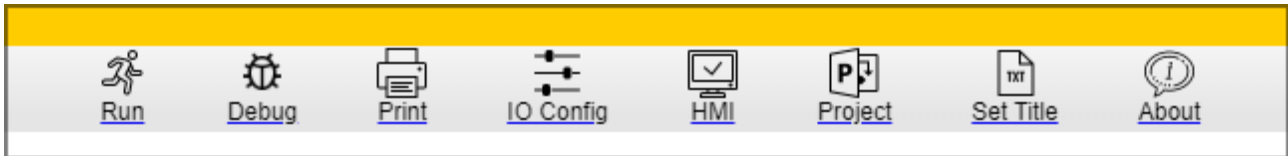


Abb. 31: ARGEE-Kopfzeile

Mit einem Klick auf **Run** wird der Code kompiliert.

1. Beim Klick auf **Run** wird der Code auf Fehler überprüft.
2. Wenn der Code Fehler enthält, wird eine Fehlermeldung ausgegeben und beschrieben, in welchem Abschnitt des Codes sich der Fehler befindet. Außerdem wird die Zeile, die den Fehler enthält, rot markiert.
3. Wenn der Code keine Fehler aufweist, lädt ARGEE den Code auf das ARGEE-fähige Gerät.
4. ARGEE berechnet den benötigten Speicherplatz und den noch frei verfügbaren Speicherplatz für den Code.
5. ARGEE wechselt zum **Debug**-Screen.

Beispiel Fehlermeldung

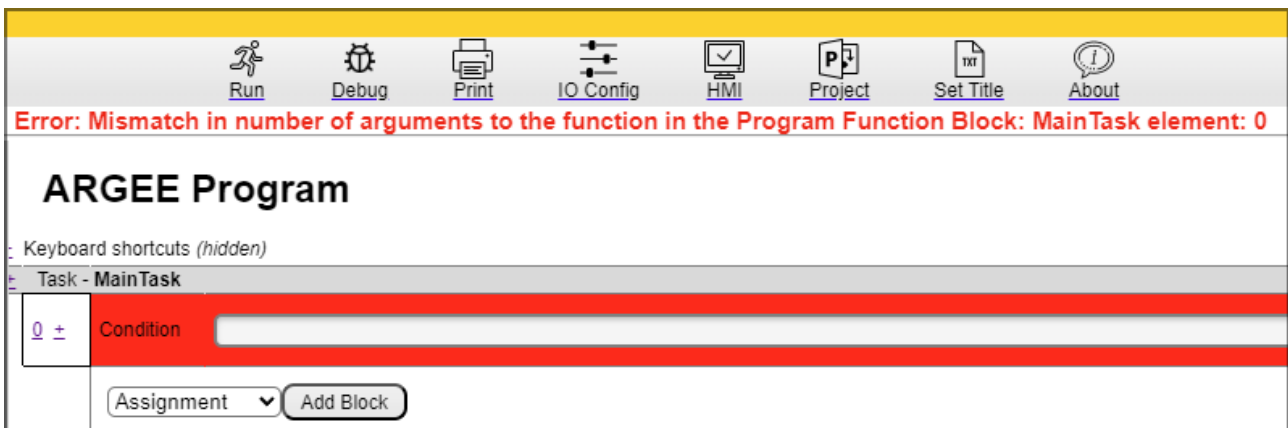


Abb. 32: ARGEE Error

6.4.2 Debug Optionen

Im Debug-Menü kann der kompilierte Code auf Fehler überprüft werden.

Funktionen des Debug-Menüs:

- Breakpoints
- Trace
- Funktionen der Kopfzeile

Breakpoints

Mithilfe von Breakpoints kann das Programm an einem gewünschten Statement angehalten werden. Das kompilierte Programm läuft von Anfang bis zum ersten Breakpoint und wird angehalten.

Breakpoints können mit einem Klick auf die Zahl des anzuhaltenden Statements angeschaltet und ausgeschaltet werden.

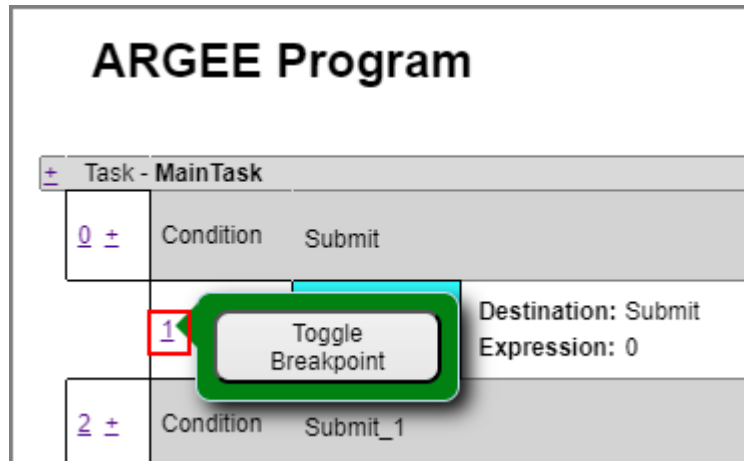


Abb. 33: Breakpoint setzen

Debug Hauptmenü

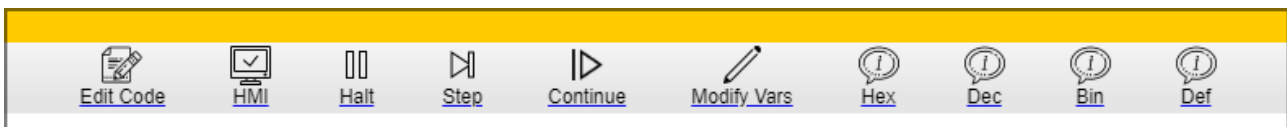


Abb. 34: Debug-Menü Kopfzeile

Funktion	Beschreibung
Edit Code	Über Edit Code öffnet sich die ARGEE Program -Maske.
HMI	Über HMI kann ein zuvor programmierter HMI-Screen angezeigt werden.
Halt	Hält das Programm an.
Step	Um Step verwenden zu können, muss das Programm vorher über Halt angehalten worden sein. Mit Step kann der Programmcode Zeile für Zeile durchlaufen werden. Ein Klick auf Step entspricht einer Zeile im Programmcode.
Continue	Wenn das ARGEE-Programm zuvor über Halt angehalten wurde, kann es über Continue wieder normal ausgeführt werden.
Modify Vars	Modify Vars steht für Modify Variables. Über Modify Vars können Variablen im Runtime Status -Fenster geändert werden. Kürzlich geänderte Variablen werden gelb dargestellt. Modify Vars kann dazu verwendet werden ARGEE im Debug-Modus manuell zu steuern.
Hex	Ändert die Zahlendarstellung auf die Hexadezimal-Schreibweise.
Dec	Ändert die Zahlendarstellung auf die Dezimal-Schreibweise.
Bin	Ändert die Zahlendarstellung auf die Binär-Schreibweise.
Def	Ändert die Zahlendarstellung auf den Default-Wert.

Trace

Trace-Statements dienen dazu, das Laufzeitverhalten eines Programms zu messen.

Trace-Statements müssen manuell im Programmcode gesetzt werden.

Mithilfe der Trace Funktion kann nachvollzogen werden, wie lange jeder Zustand andauert und welche Statements in welcher Reihenfolge besucht worden sind.

Während das Programm weiterläuft, kann eine Liste der letzten 50 Zustände mit einem Klick auf **Pause/Resume** generiert werden.

Die Liste zeigt auf die Millisekunde genau an, in welcher Zeile des Codes, welche Daten übertragen worden sind.

Über die Schaltfläche **Clear Trace** können die Daten aus der Liste gelöscht werden.

Trace Beispiel

Runtime Status

TRACE

Pause/Resume Clear Trace

Time	line	data
7797221	2	Count reg1:7
7795271	2	Count reg1:6
7793443	2	Count reg1:5
7791259	2	Count reg1:4
7789145	2	Count reg1:3
7786593	2	Count reg1:2
7784267	2	Count reg1:1

PROG CYCLE TIME : 2
 PLC CONNECTED : 0

ALIAS VARIABLES

- MainTask

reg1 : 7

ARGEE Program

Task - MainTask

0	If	R_Trig(IO_Basic_Input_Input_value_0)
1	Assignment	Destination: reg1 Expression: reg1+1
2	Trace	Prefix String: Count reg1 Expression: reg1

Abb. 35: Trace Beispiel

Wenn Input 0 von Off zu On wechselt, wird reg1 vom Programm um 1 erhöht.

Anschließend wird der Wert von reg1 im Trace Buffer aufgeführt. Der Trace zeigt in der linken Spalte (Time) den Zeitstempel in ms und rechts (data) den Präfix-String („Count reg1“) an. Der Präfix-String ist an den Wert von reg1 angehängt.

Zwischen der ersten Einschaltung (7784267 ms) und zweiten Einschaltung (7786593 ms) des Inputs sind hier 2326 ms (2,3 s) vergangen.

6.4.3 HMI

In ARGEE können verschiedene HMI-Screens programmiert werden, die zur Visualisierung und Bedienung von ARGEE-Programmen verwendet werden können. Unter dem HMI-Screens-Reiter lassen sich vier verschiedene HMI-Screen-Typen und eine Kommentarfunktion anwählen:

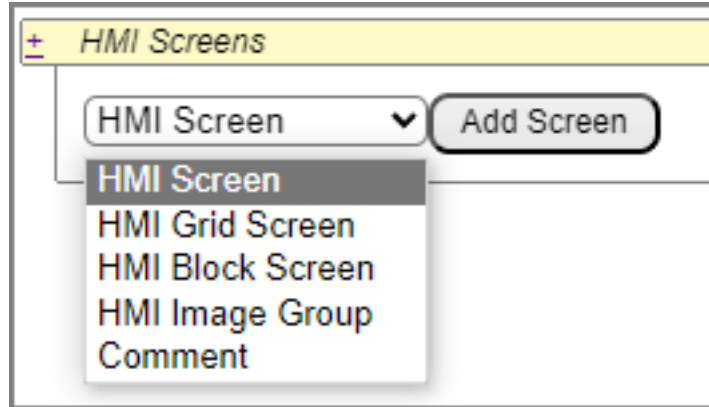


Abb. 36: Auswahl ARGEE HMI Screens

Typ	Beschreibung
HMI Screen	Daten (z. B. grundlegende Bedieneingaben) können einspaltig von oben nach unten dargestellt werden.
HMI Grid Screen	Daten können in Tabellen, Gittern und Spalten dargestellt werden. HMI Grid Screen unterstützt die Darstellung von Bildern.
HMI Block Screen	Über die Option HMI Block Screen kann der HMI-Bildschirm vollständig vom Benutzer angepasst werden. Die einzelnen Elemente des HMI-Bildschirms werden als Funktionsblöcke geschrieben und in der HMI-Bibliothek abgespeichert. Die Funktionsblöcke werden in Java Script erstellt. Durch die zusätzlich erstellten Blöcke können zusätzliche Funktionen für den HMI-Screen eingefügt werden.
HMI Image Group	Dient als Speichercontainer für Bilder, die im HMI-Screen verwendet werden.
Comment	Mithilfe von Comment können Kommentare im HMI-Screens-Abschnitt erstellt werden.

7 Turck-Niederlassungen – Kontaktdaten

Deutschland	Hans Turck GmbH & Co. KG Witzlebenstraße 7, 45472 Mülheim an der Ruhr www.turck.de
Australien	Turck Australia Pty Ltd Building 4, 19-25 Duerdin Street, Notting Hill, 3168 Victoria www.turck.com.au
Belgien	TURCK MULTIPROX Lion d'Orweg 12, B-9300 Aalst www.multiprox.be
Brasilien	Turck do Brasil Automação Ltda. Rua Anjo Custódio Nr. 42, Jardim Anália Franco, CEP 03358-040 São Paulo www.turck.com.br
China	Turck (Tianjin) Sensor Co. Ltd. 18,4th Xinghuazhi Road, Xiqing Economic Development Area, 300381 Tianjin www.turck.com.cn
Frankreich	TURCK BANNER S.A.S. 11 rue de Courtalin Bat C, Magny Le Hongre, F-77703 MARNE LA VALLEE Cedex 4 www.turckbanner.fr
Großbritannien	TURCK BANNER LIMITED Blenheim House, Hurricane Way, GB-SS11 8YT Wickford, Essex www.turckbanner.co.uk
Indien	TURCK India Automation Pvt. Ltd. 401-403 Aurum Avenue, Survey. No 109 /4, Near Cummins Complex, Baner-Balewadi Link Rd., 411045 Pune - Maharashtra www.turck.co.in
Italien	TURCK BANNER S.R.L. Via San Domenico 5, IT-20008 Bareggio (MI) www.turckbanner.it
Japan	TURCK Japan Corporation ISM Akihabara 1F, 1-24-2, Taito, Taito-ku, 110-0016 Tokyo www.turck.jp
Kanada	Turck Canada Inc. 140 Duffield Drive, CDN-Markham, Ontario L6G 1B5 www.turck.ca
Korea	Turck Korea Co, Ltd. A605, 43, Iljik-ro, Gwangmyeong-si 14353 Gyeonggi-do www.turck.kr
Malaysia	Turck Banner Malaysia Sdn Bhd Unit A-23A-08, Tower A, Pinnacle Petaling Jaya, Jalan Utara C, 46200 Petaling Jaya Selangor www.turckbanner.my

Mexiko	Turck Comercial, S. de RL de CV Blvd. Campestre No. 100, Parque Industrial SERVER, C.P. 25350 Arteaga, Coahuila www.turck.com.mx
Niederlande	Turck B. V. Ruiterlaan 7, NL-8019 BN Zwolle www.turck.nl
Österreich	Turck GmbH Graumanngasse 7/A5-1, A-1150 Wien www.turck.at
Polen	TURCK sp.z.o.o. Wroclawska 115, PL-45-836 Opole www.turck.pl
Rumänien	Turck Automation Romania SRL Str. Siriului nr. 6-8, Sector 1, RO-014354 Bucuresti www.turck.ro
Schweden	Turck AB Fabriksstråket 9, 433 76 Jonsered www.turck.se
Singapur	TURCK BANNER Singapore Pte. Ltd. 25 International Business Park, #04-75/77 (West Wing) German Centre, 609916 Singapore www.turckbanner.sg
Südafrika	Turck Banner (Pty) Ltd Boeing Road East, Bedfordview, ZA-2007 Johannesburg www.turckbanner.co.za
Tschechien	TURCK s.r.o. Na Brne 2065, CZ-500 06 Hradec Králové www.turck.cz
Türkei	Turck Otomasyon Ticaret Limited Sirketi Inönü mah. Kayisdagi c., Yesil Konak Evleri No: 178, A Blok D:4, 34755 Kadiköy/ Istanbul www.turck.com.tr
Ungarn	TURCK Hungary kft. Árpád fejedelem útja 26-28., Óbuda Gate, 2. em., H-1023 Budapest www.turck.hu
USA	Turck Inc. 3000 Campus Drive, USA-MN 55441 Minneapolis www.turck.us

TURCK

Your Global Automation Partner



Over 30 subsidiaries and
60 representations worldwide!

100048239 | 2024/06



www.turck.com