

Selmo Solution



For
software
that never
lets you
down!

1	Selmo in Use	3
1.1	Anlage strukturieren	6
1.1.1	Schrittfolge einbetten (Sequence)	9
1.1.2	Was passiert in der PLC/HMI	13
1.1.3	Szenario mehrere HWZ	25
1.1.4	Szenario mehrere Sequenzen	28
1.2	Logic anlegen	31
1.2.1	Elemente der Modellierung	32
1.2.2	Cross Sequence	52
1.2.3	Umwandlung in Systemlayer	61
1.3	Signale anlegen	62
1.3.1	Zone	63
1.3.1.1	Zone In	67
1.3.1.2	Zone InOut	85
1.3.1.3	Zone Out	101
1.3.1.4	Zone Mem	106
1.3.2	Bit Controlled	109
1.3.3	CMZ	116
1.3.3.1	Fatal Faults	124
1.3.3.2	Gate/ Fortress Faults	125
1.3.3.3	Warning Faults	126
1.4	Ergebnis nach Schritt 1 - 3	127
1.5	Hand Funktionen	131
1.6	Flexibilisierung über Parameter	135
1.6.1	Parameter Ebenen	136
1.6.2	Szenarien wo können Parameter eingebunden werden	157
1.6.3	OPC UA	162
1.7	Finalisierung	163
1.8	Assembly/driver layer	164
1.9	HMI	173
1.9.1	Sequence Control	174
1.9.2	Alarm Handling	193
1.9.3	Plant	198
2	Hardware Zone	203
	Index	211

Selmo solution

Selmo in Use

1 Selmo in Use

Selmo in Use beschreibt jeden Schritt im Selmo Studio, von der Projekterstellung bis hin zur Anwendung im Betrieb. Dabei liegt der Fokus auf der Anwendung des Selmo Studio, wie Schrittschaltwerke und Signale hier modelliert und definiert werden.

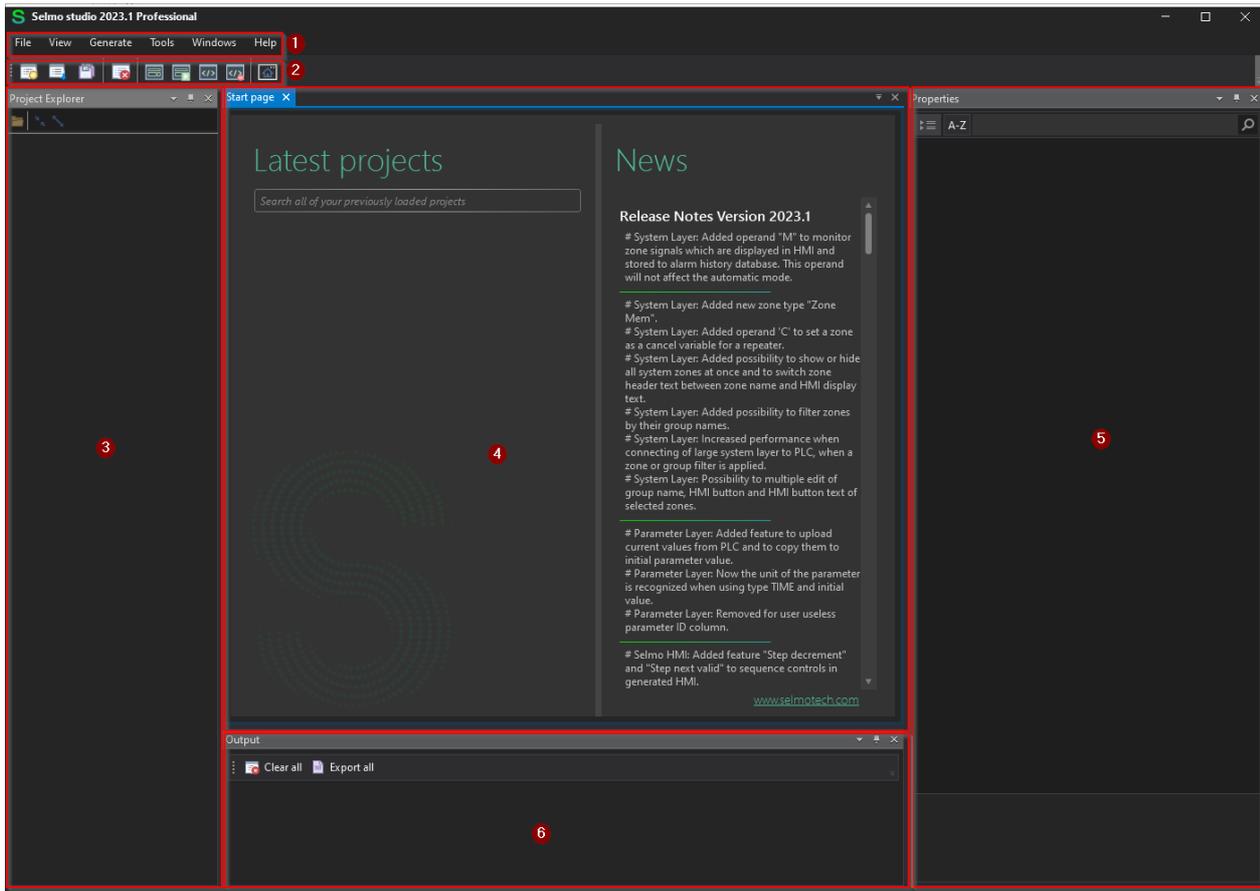
Die Dokumentation bezieht sich auf die Selmo Studio Version 2023.1 SP1.



Studio

Eine Detailbeschreibung aller Features ist im entsprechenden Kapitel Selmo Studio zu finden. Nachfolgend erklärt werden lediglich die einzelnen Schritte, die essenziell sind, um ein Projekt anzulegen. Die Startseite beinhaltet verschiedene Steuerelemente die nachfolgend aufgelistet werden.

1. Menü
2. Schnellzugriff Leiste
3. Projekt Explorer
4. Editor-Fenster
5. Eigenschafts-Fenster
6. Ausgabe-Fenster



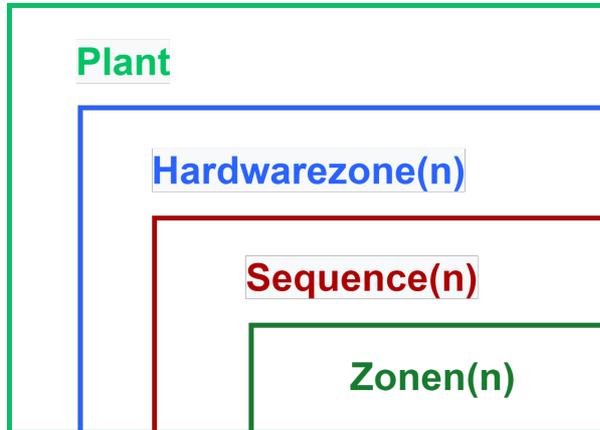
Studio

1.1 Anlage strukturieren

Die Anlagenstruktur beschreibt die hierarchische Gliederung einer technischen Anlage in logische und funktionale Einheiten. Die Anlagenstruktur kann je nach Anwendung unterschiedlich sein, aber typischerweise besteht sie aus drei Ebenen: Plant, Hardwarezone und Sequence. Die Plant-Ebene ist die oberste Ebene der Anlagenstruktur und repräsentiert die gesamte Anlage. Die Plant-Ebene dient der übersichtlichen Darstellung und Organisation der Anlage sowie der Zuordnung von globalen Parametern und Funktionen. Die Hardwarezone-Ebene ist die mittlere Ebene der Anlagenstruktur und unterteilt die Plant-Ebene in kleinere Abschnitte, die jeweils eine bestimmte Hardware-Komponente oder ein Modul enthalten. Die Hardwarezone-Ebene dient der Definition und Konfiguration der Hardware-Schnittstellen und -Eigenschaften sowie der Verwaltung von Ressourcen und Sicherheitsaspekten. Jede Hardwarezone besitzt eine unabhängige Automatik und Manuell Steuerung. Die Sequence-Ebene ist die unterste Ebene der Anlagenstruktur und beschreibt die Abläufe und Prozesse innerhalb einer Hardwarezone. Die Sequence-Ebene dient der Programmierung und Steuerung der logischen Funktionen und Algorithmen sowie der Überwachung und Diagnose des Betriebszustands. Die Sequence-Ebene wird in verschiedenen Layer unterteilt.

Struktur des Selmo Projekts

Der Systemaufbau erfolgt immer nach gleicher Struktur:



Dieser besteht aus verschiedenen Ebenen. Die oberste Ebene wird als Plant bezeichnet und stellt das Gesamtsystem dar. Die Plant enthält immer mindestens eine Subebene, die Hardwarezone(n). Die Hardwarezone kann eine oder mehrere Sequences umfassen. Eine Sequence ist immer nur einer Hardwarezone zugeordnet. Eine Sequence ist ein Teilsystem, das eine logische Prozessbeschreibung definiert. Diese besteht aus einer logischen Prozessbeschreibung in Form einer Schrittkette (Logic Layer), die Definition von Zuständen durch Zonen (System Layer), die Verriegelungen im Handbetrieb (MXIC) und konstant überwachte Zonen (CMZ).

Plant

In der Industrie bezeichnet der Begriff "Plant" in der Regel eine Anlage oder Einrichtung, die zur Herstellung oder Verarbeitung von Produkten oder Rohstoffen eingesetzt wird. Dies kann beispielsweise eine Fabrik, ein Werk, ein Kraftwerk oder eine Raffinerie sein. Eine Plant umfasst oft mehrere Einheiten, die in einer bestimmten Abfolge arbeiten, um das gewünschte Produkt herzustellen oder den Rohstoff zu verarbeiten. Dies können beispielsweise Fertigungsanlagen, Maschinen, chemische Reaktoren, Lager- und Versorgungseinrichtungen oder Kontrollsysteme sein. Plants können sehr groß und komplex sein und erfordern oft spezielle Kenntnisse und Fähigkeiten, um sie zu betreiben und zu warten. Daher arbeiten in der Regel mehrere Fachleute wie Ingenieure, Techniker und Arbeiter zusammen, um eine Plant zu betreiben und optimale Leistungen zu erzielen.

Hardwarezone (HWZ)

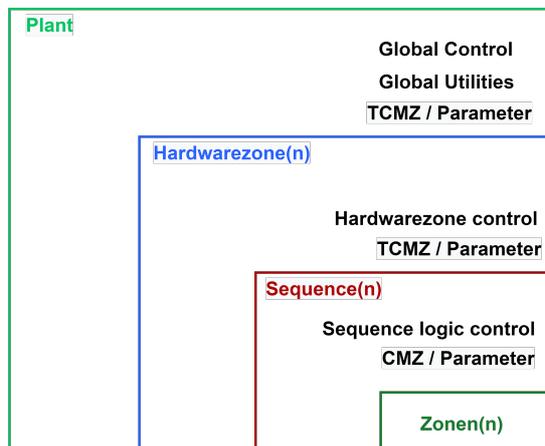
Eine Hardware-Zone (kurz HWZ) ist ein bestimmter Bereich einer Maschine oder Zelle, der über eine separate Automatik- und manuelle Steuerung verfügt. Jede Hardware-Zone hat ihre eigene, spezifische Übersicht. Die Größe einer Hardware-Zone hängt von der Konfiguration der Maschine oder Zelle ab und kann eine oder mehrere Schrittsequenzen enthalten und steuern.

Sequence

Eine Ablaufsteuerung ist eine Art von Steuerungssystem, das eine Folge von vordefinierten Schritten oder Zuständen ausführt. Ein Beispiel für eine Ablaufsteuerung ist ein Schrittschaltwerk, das aus einer Reihe von Speicherelementen besteht, die jeweils einen Schritt der Sequenz repräsentieren. Ein Schrittschaltwerk kann durch externe Eingangssignale oder interne Logik gesteuert werden. Eine Ablaufsteuerung kann verwendet werden, um komplexe Prozesse zu automatisieren oder zu synchronisieren, wie z.B. die Fertigung in einer Fabrik oder die Kommunikation in einem Netzwerk.

Elemente der Plant, Hardwarezone und Sequence

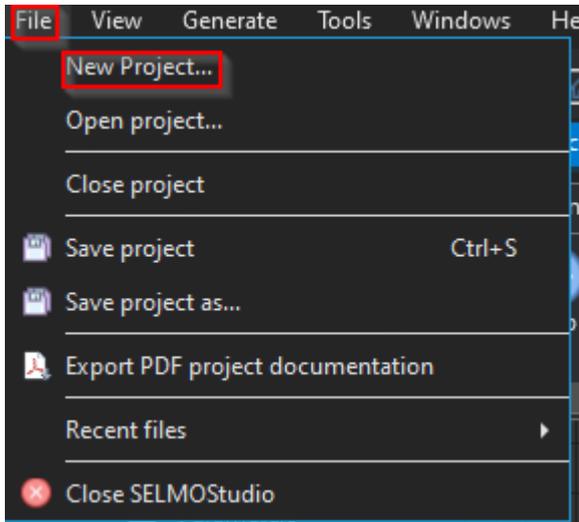
Jede Ebene eines Selmo Projekts hat bestimmte Eigenschaften und Funktionen, die im Wirkbereich gelten. Die Elemente und dessen Wirkbereich werden im folgenden Bild beschrieben:



1.1.1 Schrittkette einbetten (Sequence)

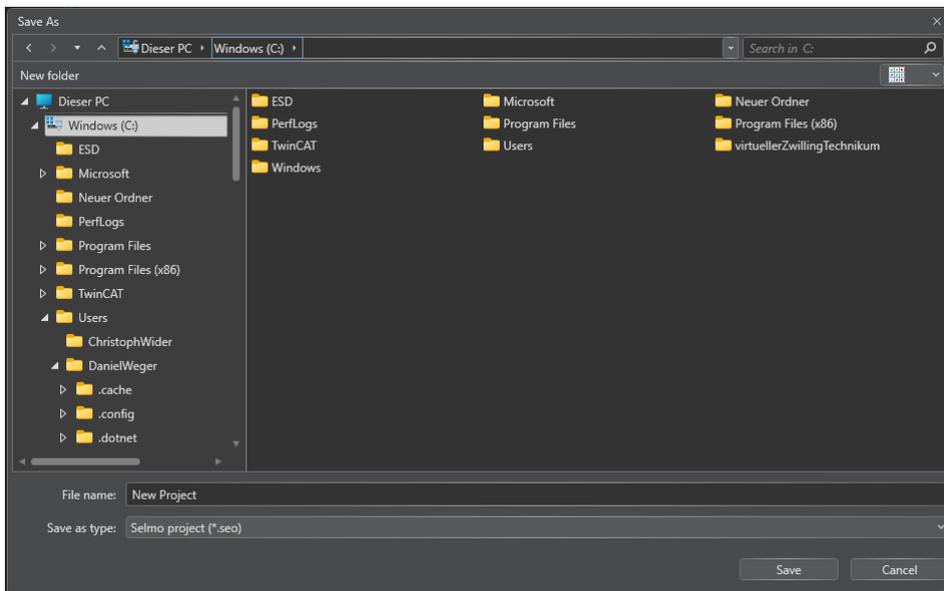
Um eine Schrittkette in eine Anlage einzubetten, müssen Sie zunächst die Anlage (Plant) anlegen. Dazu wählen Sie im Kontextmenü "File" die Option "New Project" und geben einen Namen ein.

Ein neues Project kann unter File, New Project angelegt werden.



Studio

Projektpfad auswählen und einen eindeutigen Namen vergeben. Jedes Projekt wird als Selmo pro-ject (*.seo) angelegt.



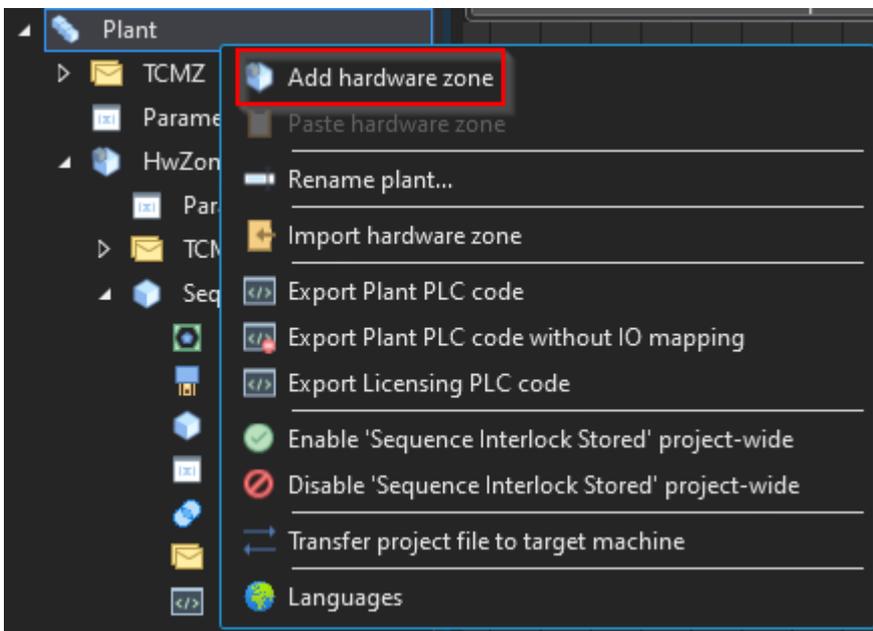
Studio

Danach müssen Sie die Hardwarezone definieren, in der die Schrittkette ausgeführt werden soll.

Eine Hardware-Zone (kurz HWZ) ist ein bestimmter Bereich einer Maschine oder Zelle, der über eine separate Automatik- und manuelle Steuerung verfügt. Jede Hardware-Zone hat ihre eigene, spezifische Übersicht. Die Größe einer Hardware-Zone hängt von der Konfiguration der Maschine oder Zelle ab und kann eine oder mehrere Schrittsequenzen enthalten und steuern.

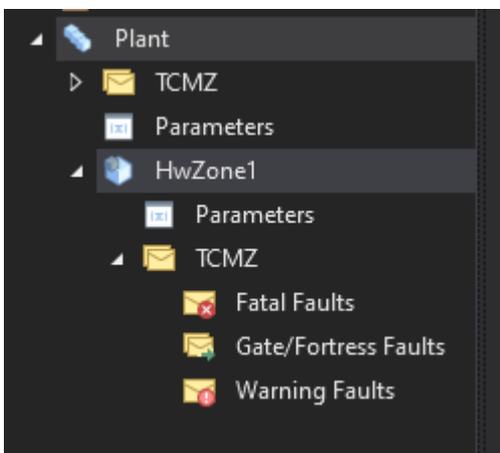
Im Selmo Studio, durch Rechtsklick auf Plant, Add Hardware Zone wird ein neuer Maschinenbereich hinzugefügt.

Es können beliebig viele Hardwarezone zu einer Plant hinzugefügt werden, es ist jedoch von der Performance der Hardware (PLC) abhängig.



Studio

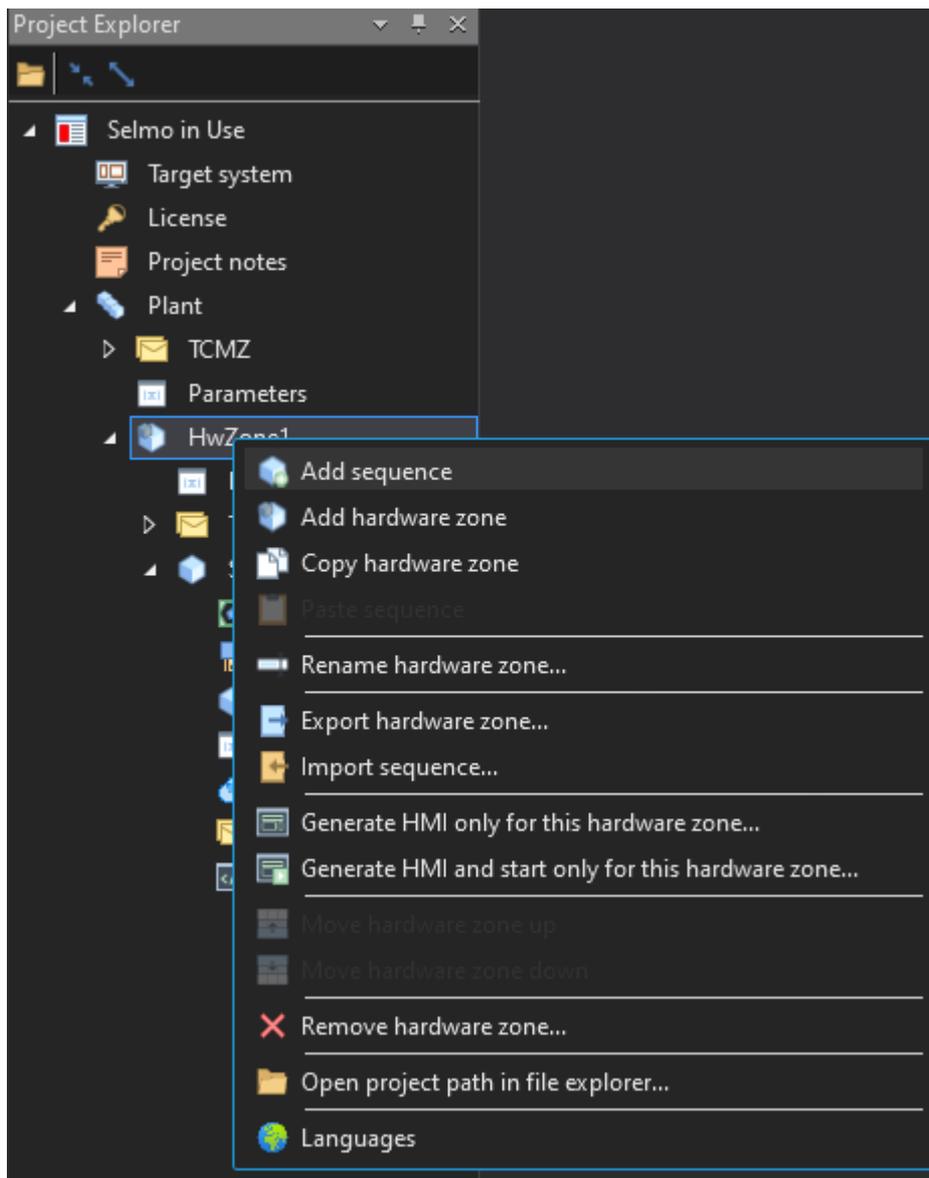
Dadurch entsteht folgende Struktur:



Studio

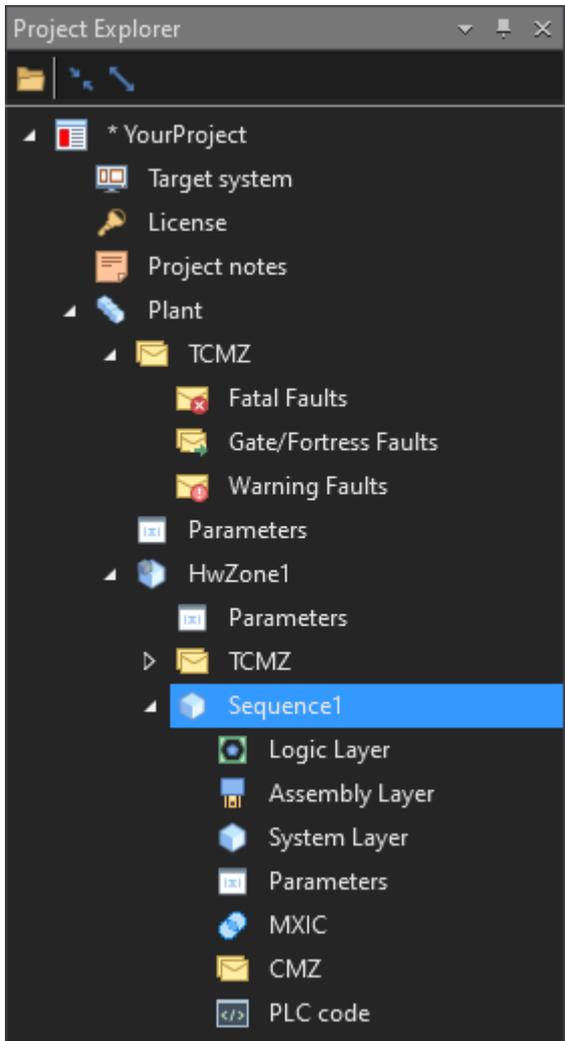
Schließlich müssen Sie die Sequence einfügen, welche die einzelnen Schritte des Ablaufes enthält.

Eine Schrittkette wird als Sequence bezeichnet und ist die Darstellung eines Prozessablaufs in einzelnen Schritten. Man fügt über einen Rechtsklick auf eine HWZ eine Sequence hinzu. Es besteht auch die Möglichkeit eine Sequence zu importieren oder zu kopieren. Jede Sequence kann nur einer Hardware-Zone zugeordnet werden, jedoch kann man mehrere Sequences einer Hardware-Zone zuordnen.



Studio

Die Struktur einer Sequence ist immer dieselbe und besteht aus mehreren Ebenen wie im folgenden Bild dargestellt ist.



Studio

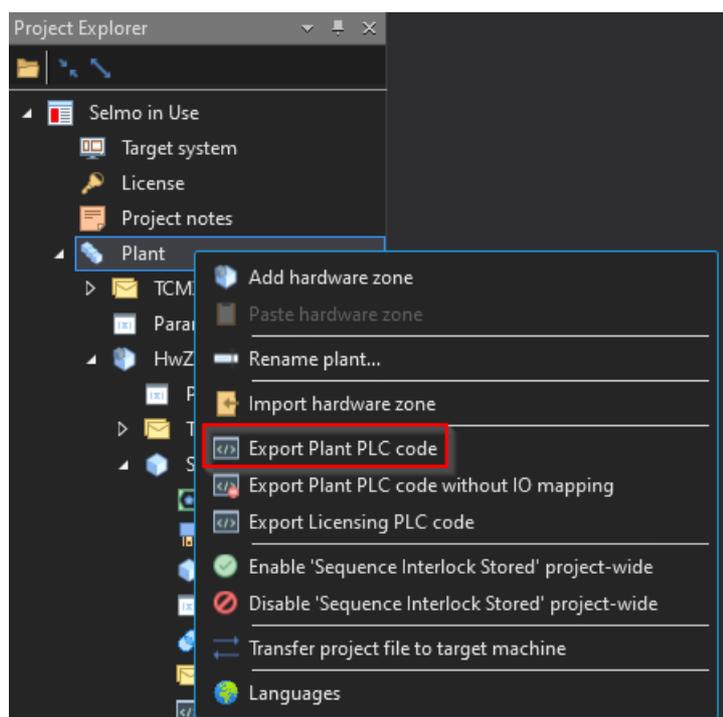
1.1.2 Was passiert in der PLC/HMI

Die Selmo Solution ist eine Softwarelösung, die es ermöglicht, Prozesse von automatisierten Anlagen mit einer grafischen Benutzeroberfläche zu erstellen und zu steuern. Um die Selmo Solution in einen PLC-Code zu konvertieren, der von der Anlage ausgeführt werden kann, muss man folgende Schritte ausführen:

- Öffnen Sie die Selmo Solution in der Selmo Studio Anwendung.
- Wählen Sie die Plant aus, die Sie exportieren möchten, indem Sie sie im Projekt Explorer anklicken.
- Klicken Sie mit der rechten Maustaste auf die Plant und wählen Sie die Option "Export Plant PLC Code" aus dem Kontextmenü.
- Geben Sie einen Namen und einen Speicherort für die exportierte Datei an und klicken Sie auf "Speichern".
- Die Selmo Solution wird nun in einen PLC-Code umgewandelt und in der angegebenen Datei gespeichert.

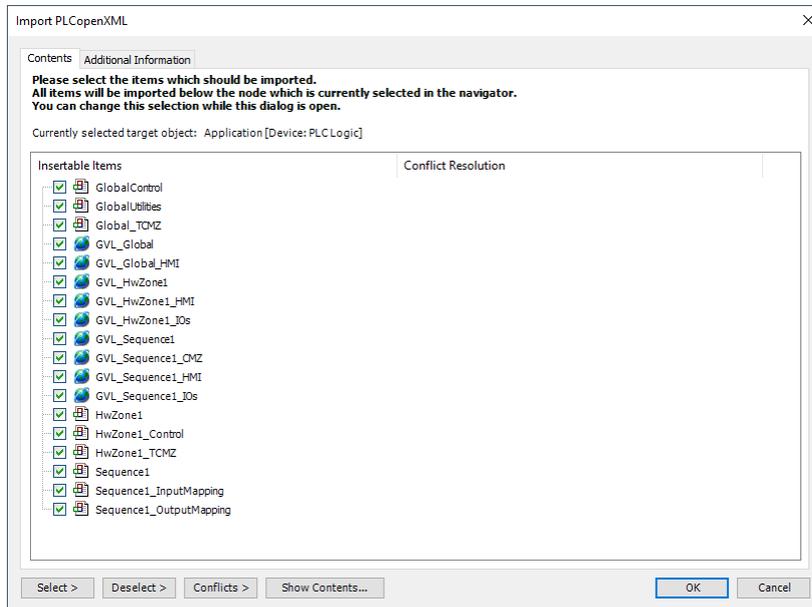
Hinweis:

Mit Selmo wurde die Möglichkeit geschaffen mittels Export Funktion ein PLCopen konformes XML zu generieren und in die gewählte Steuerung zu importieren. Dies bedeutet, dass das Programm, die Bibliotheken und die Projekte, die mit IEC 61131-3 erstellt wurden, in einem standardisierten XML-Format gespeichert werden können, das von verschiedenen Software-Tools gelesen und bearbeitet werden kann. Das PLCopen XML-Format ist Teil der IEC 61131-Normenreihe und wird als IEC 61131-10 bezeichnet. Es bietet eine offene Schnittstelle für den Austausch von Informationen zwischen Entwicklungsumgebungen und anderen Plattformen. Das PLCopen XML-Format ist in der technischen Dokumentation von PLCopen beschrieben.



Studio

Sie können die exportierte Datei dann in Ihrer bevorzugten PLC-Programmierungsumgebung öffnen und auf die Anlage übertragen.



PLC

Dieses PLC Programm beinhaltet folgende Elemente die für die Steuerung und Diagnose notwendig sind:

Die **Plant** Elemente sind die grundlegenden Bausteine für die Konfiguration und Steuerung der Anlage. Sie haben folgende Eigenschaften:

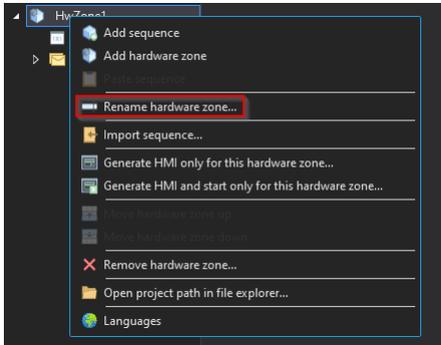
- Sie wirken **GLOBAL**, das heißt, sie beeinflussen alle Hardwarezonen und alle lizenzierten Funktionen der Anlage.
- **Global Control** ist ein Element, das den allgemeinen Aufruf der Hardwarezonen und die Lizenzierung ermöglicht.
- **Global Utilities** ist ein Element, das einige funktionale Elemente zur Verfügung stellt, die für die Diagnose und Wartung der Anlage nützlich sind. Zum Beispiel: Lamp Test, Global Release, Global Reset, usw.
- Die Anlage verfügt über ein **Global TCMZ** (Total Constantly Monitored Zone) System, das alle wichtigen Parameter der Anlage ständig überwacht, wie zum Beispiel die Druckluft- oder Schmiermittellversorgung. Das Global TCMZ-System wirkt auf alle Hardwarezonen der Anlage und kann bei einer Fehler Detektion die Automatik in jeder Hardwarezone unterbrechen. Dies erhöht die Sicherheit und Effizienz der Anlage.
- **GVL_Global_*** ist eine wichtige Ressource für die Programmierung von Anlagen mit verschiedenen Elementen. Sie ermöglicht es, die Variablen, die für die Funktion und Steuerung der Elemente benötigt werden, global zu definieren und zu verwalten. So können die Elemente miteinander kommunizieren und Daten austauschen. Die **GVL_*** enthält die Namen, Typen und Werte der Variablen, die den Elementen zugeordnet sind.

Die **Hardwarezonen** sind für die Steuerung und Überwachung der einzelnen Maschinenbereiche notwendig. Sie ermöglichen es, zwischen den Betriebsarten Hand und Automatik zu wechseln und die definierten Sequences auszuführen.

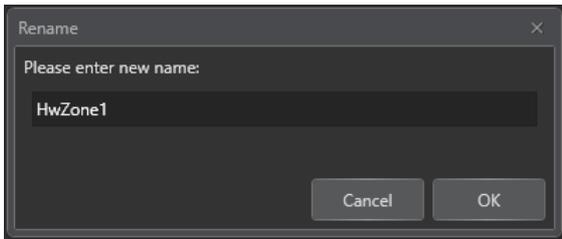
- **<HwZone1>** ist ein Element, das alle Sequences in der Hardwarezone aufruft und steuert.
- **<HwZone1>_Control** ist ein Element, das zusätzliche Funktionen wie Betriebsartwahl, EOC-Mode oder Statusinformationen der Sequences bereitstellt.
- **<HwZone1>TCMZ** ist ein Element, das die Hardwarezone ständig überwacht und bei einem Fehler die Automatik unterbricht.
- **GVL_<HwZone1>_*** ist eine wichtige Ressource für die Programmierung von Maschinenbereichen mit verschiedenen Elementen. Sie ermöglicht es, die Variablen, die für die Funktion und Steuerung der Elemente benötigt werden, global zu definieren und zu verwalten. So können die Elemente miteinander kommunizieren und Daten austauschen. Die GVL_* enthält die Namen, Typen und Werte der Variablen, die den Elementen zugeordnet sind.

Achtung:

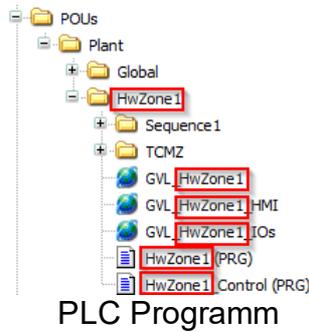
Durch Umbenennen der Hardwarezone ändert sich der Name <HwZone1> der gesamten Struktur inkl. der Globalen Variablenliste, PLC-Programm und HMI (siehe PLC-Code).



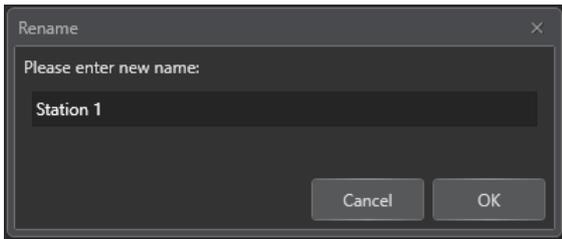
Studio



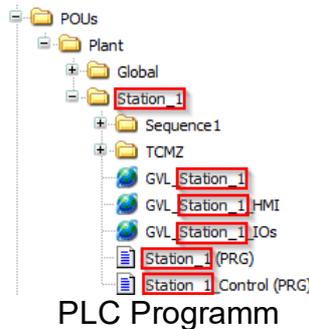
Studio



PLC Programm



Studio



PLC Programm

Die **Sequence** Elemente beschreiben die Abläufe und Prozesse innerhalb einer Hardwarezone. In einer Hardwarezone können beliebig viele Sequences definiert werden und bestehen immer aus den folgenden Elementen:

- **<Sequence1>** ist die Basis einer logischen Prozessabfolge und wird in folgenden Bereichen unterteilt:

- Standard Beginning

- Sequence Logic Control

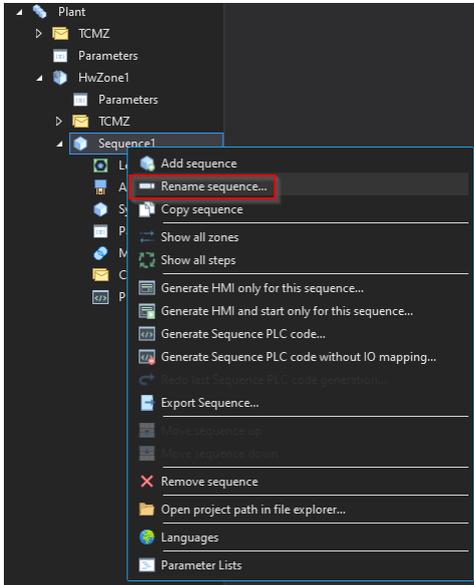
- CMZ

- Standard End

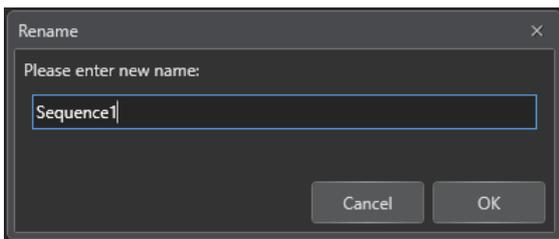
- Im **<Sequence1>_InputMapping** werden Input Signale der realen Welt der Sequence zugewiesen.
- Im **<Sequence1>_OutputMapping** werden Output Signale der realen Welt der Sequence zugewiesen.
- **GVL_<Sequence1>*** ist eine wichtige Ressource für die Programmierung von Logischen Prozessabläufe mit verschiedenen Elementen. Sie ermöglicht es, die Variablen, die für die Funktion und Steuerung der Elemente benötigt werden, global zu definieren und zu verwalten. So können die Elemente miteinander kommunizieren und Daten austauschen. Die **GVL_*** enthält die Namen, Typen und Werte der Variablen, die den Elementen zugeordnet sind.

Achtung:

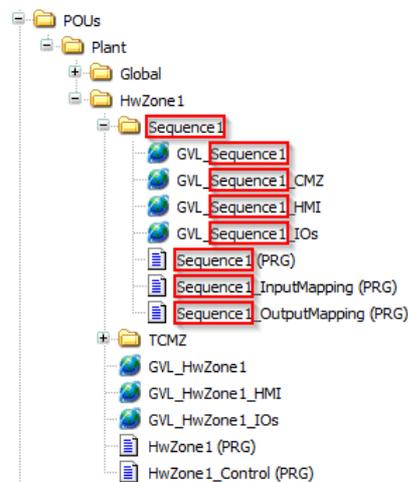
Durch Umbenennen der Sequence ändert sich der Name <Sequence1> der gesamten Struktur inkl. der globalen Variablenliste, PLC Programm und HMI (siehe PLC-Code).



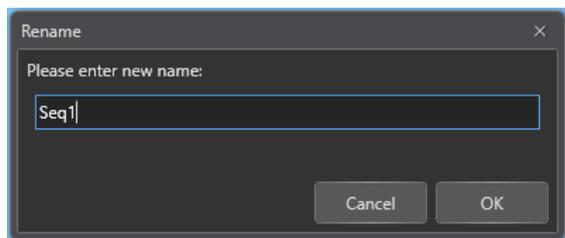
Studio



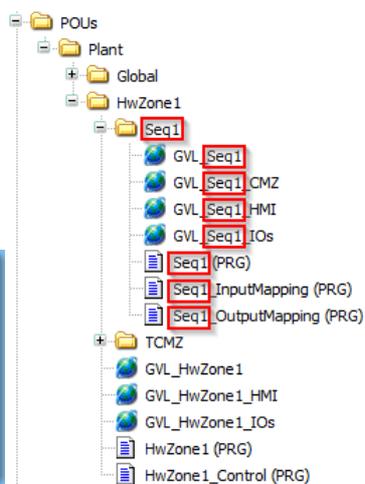
Studio



PLC Programm



Studio



PLC Programm

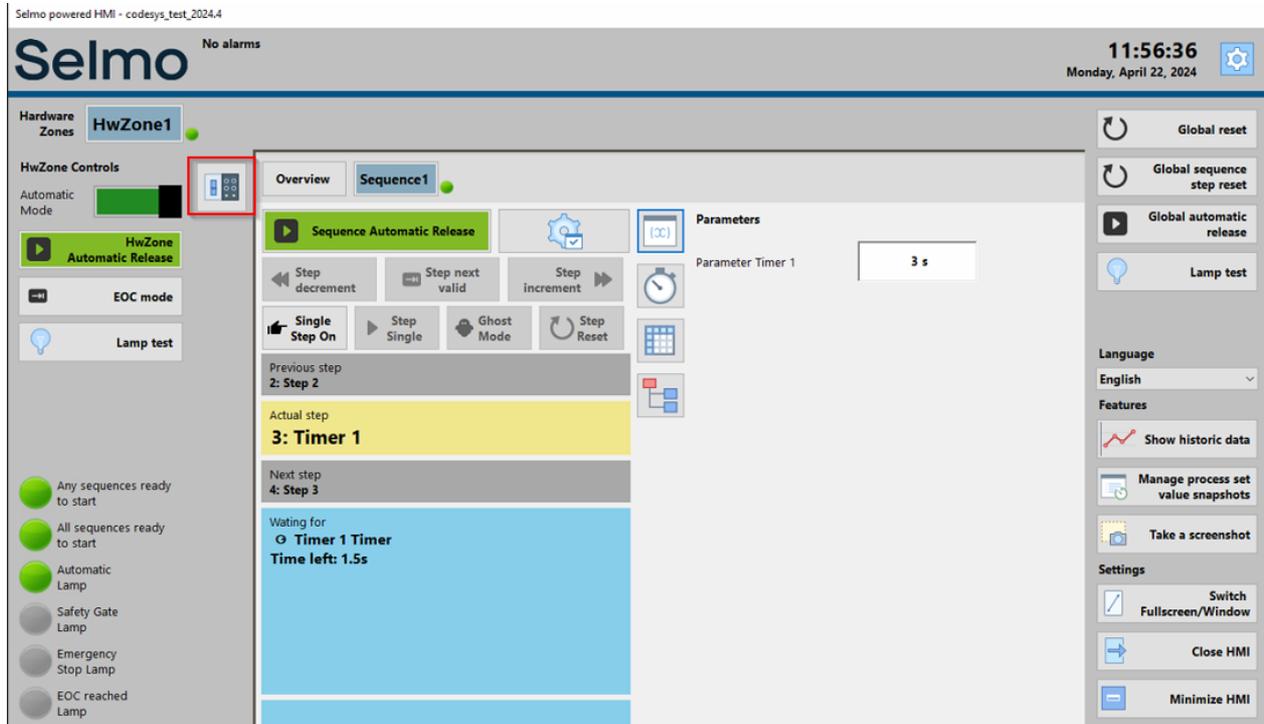
Nachdem der PLC-Code der gesamten Plant exportiert wurde, kann eine HMI generiert werden, welche dieselbe Struktur aufweist wie das PLC Programm:

- Plant: Dies ist die oberste Ebene, welche die gesamte Anlage repräsentiert. Hier kann der Benutzer einen Überblick über den Zustand und die Leistung der Anlage erhalten.

The screenshot displays the Selmo HMI interface for a hardware zone named 'HwZone1'. The top header shows the 'Selmo' logo, 'No alarms', and the time '11:56:36' on 'Monday, April 22, 2024'. The main area is titled 'Sequence1' and features a 'Sequence Automatic Release' button. Below this are navigation buttons for 'Step decrement', 'Step next valid', and 'Step increment'. A 'Parameters' section shows 'Parameter Timer 1' set to '3 s'. The sequence progress is shown with 'Previous step 2: Step 2', 'Actual step 3: Timer 1', and 'Next step 4: Step 3'. A 'Waiting for' section indicates 'Timer 1 Timer' with 'Time left: 1.5s'. The left sidebar contains 'HwZone Controls' with buttons for 'Automatic Mode', 'HwZone Automatic Release', 'EOC mode', and 'Lamp test', along with a status legend. The right sidebar includes 'Global reset', 'Global sequence step reset', 'Global automatic release', 'Lamp test', 'Language' (set to English), 'Features' (Show historic data, Manage process set value snapshots, Take a screenshot), and 'Settings' (Switch Fullscreen/Window, Close HMI, Minimize HMI).

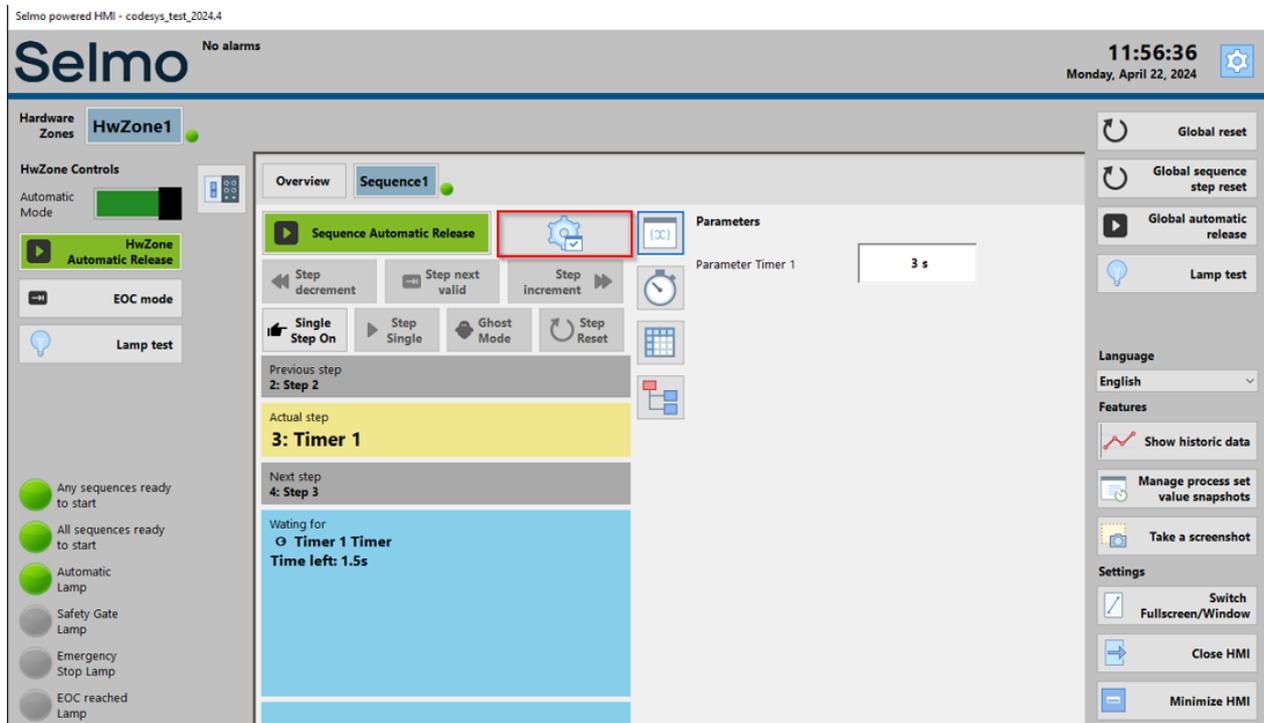
HMI

- Hardwarezonen: Dies sind die untergeordneten Ebenen, welche die verschiedenen Bereiche der Anlage darstellen. Hier kann der Benutzer die einzelnen Komponenten und Geräte der Anlage steuern und überwachen.



HMI

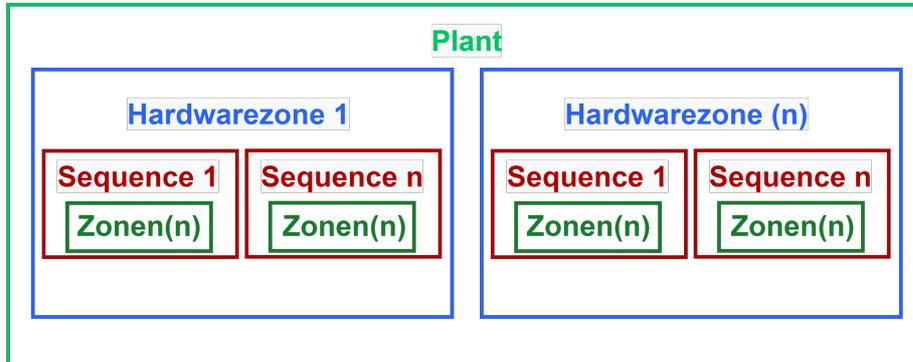
- Sequence: Dies ist die unterste Ebene, welche die Abläufe und Prozesse der Anlage beschreibt. Hier kann der Benutzer die Sequences starten, stoppen, pausieren und ändern, die für die Produktion oder den Betrieb der Anlage erforderlich sind.



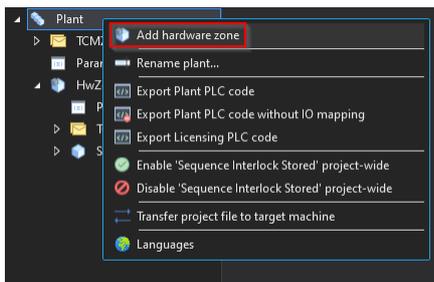
HMI

1.1.3 Szenario mehrere HWZ

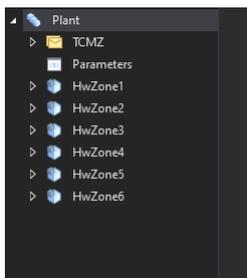
Eine Plant kann mehrere Hardwarezonen enthalten. Diese Hardwarezonen können mehrere Sequences enthalten. In diesen Sequences können Signale mit Zonen verbunden werden. Eine Plant kann dementsprechend folgende Struktur aufweisen:



Wenn mehrere Hardwarezonen eingefügt werden, erweitert sich die Struktur der Plant um die Elemente der zuvor genannten Hardwarezonen. Es gibt keine festgelegte Obergrenze, aber die Leistungsfähigkeit der verwendeten Hardware beeinflusst die Erweiterbarkeit.

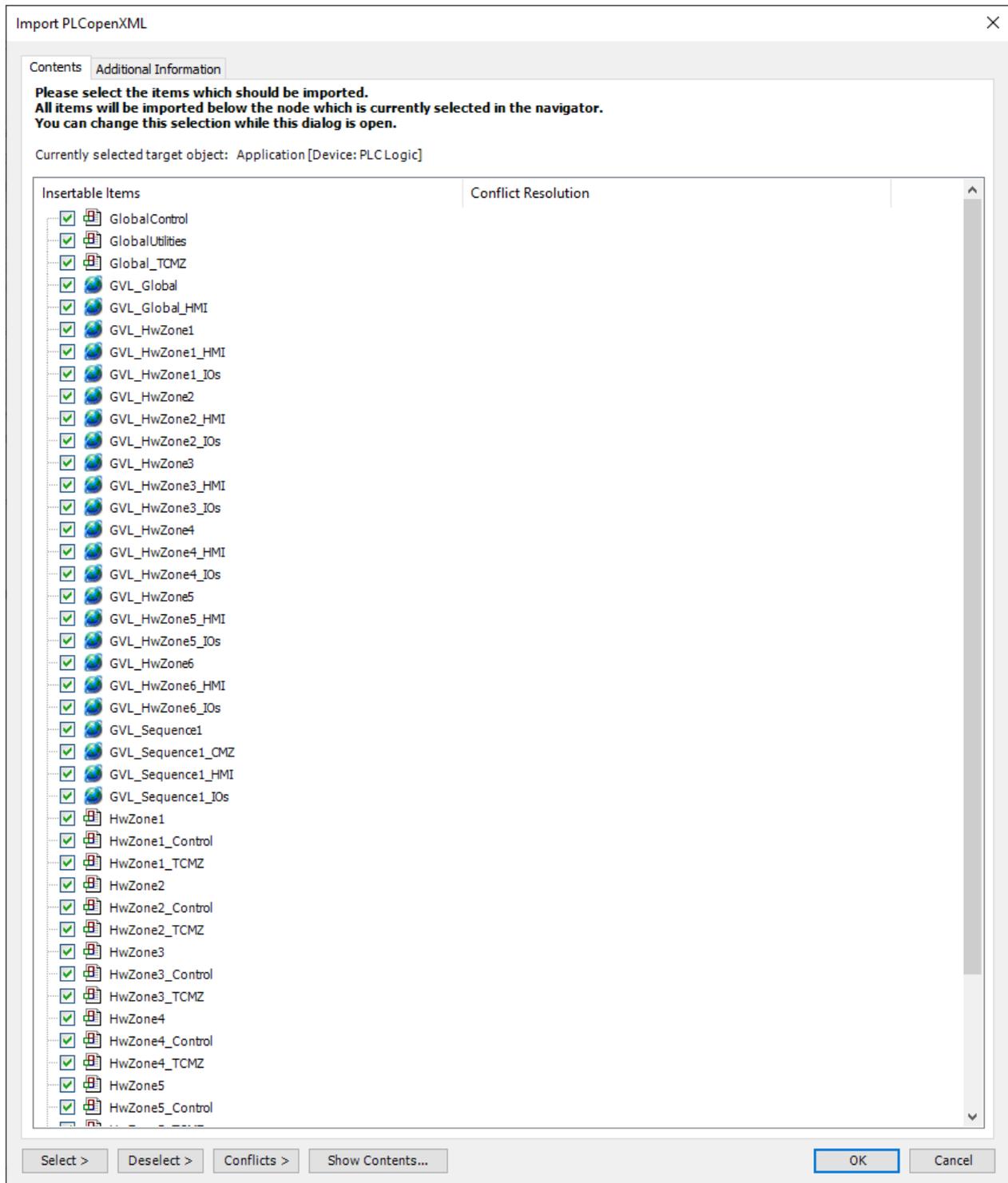


Studio



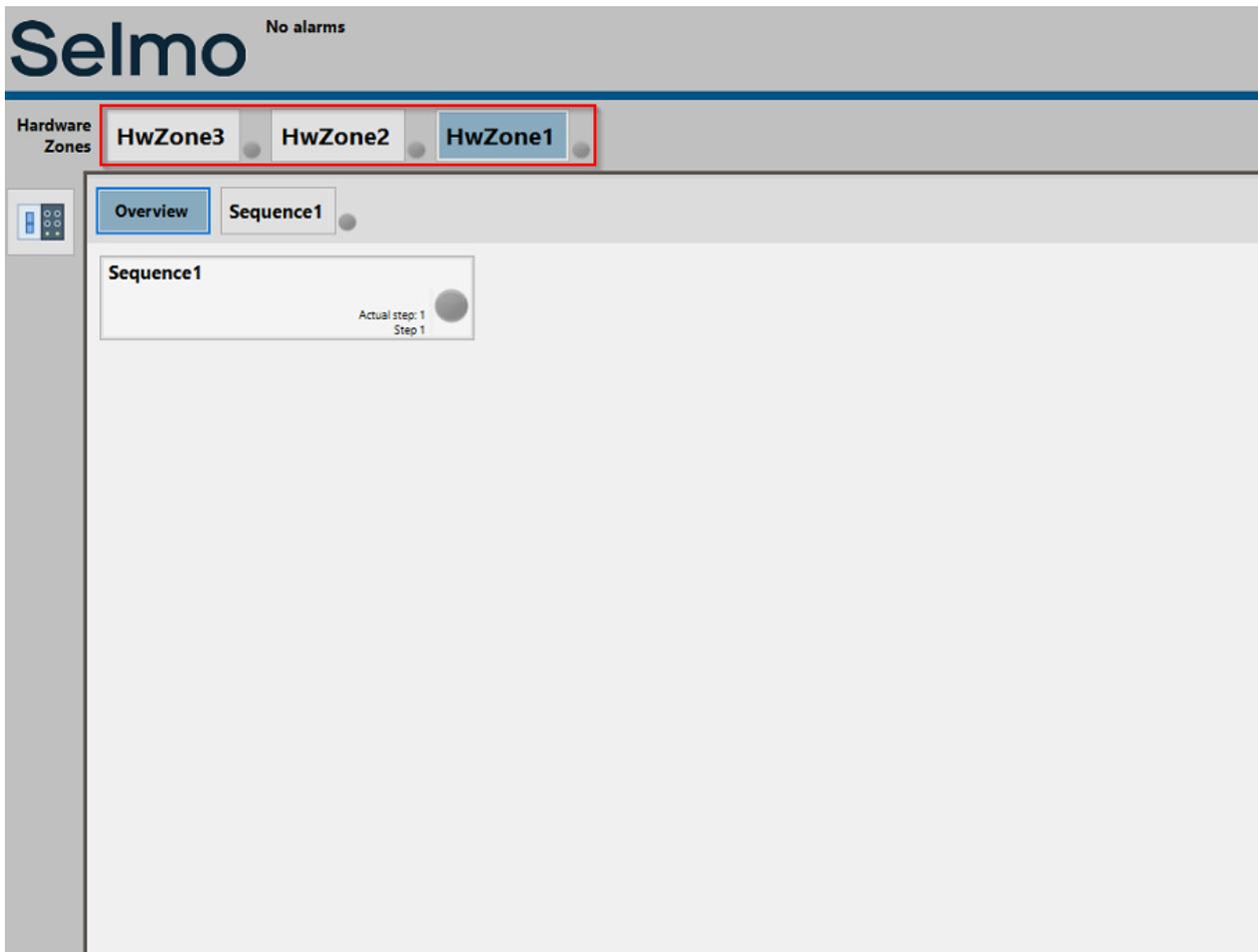
Studio

Folgende PLC Programm Elemente werden durch erneuten Export erstellt:



PLC

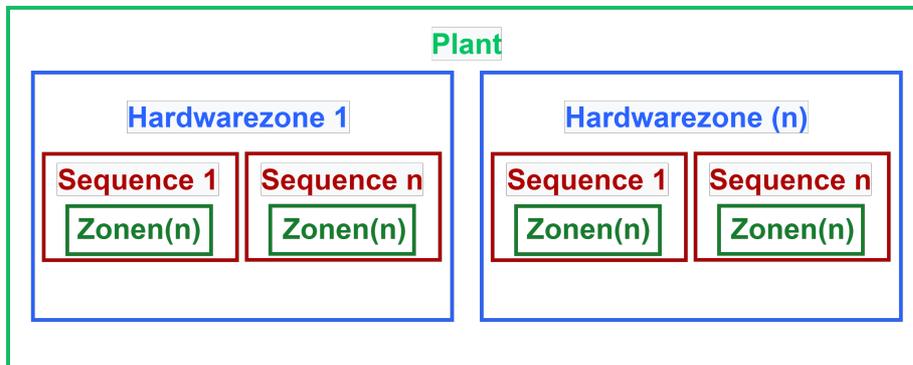
Folgende HMI Elemente werden erzeugt:



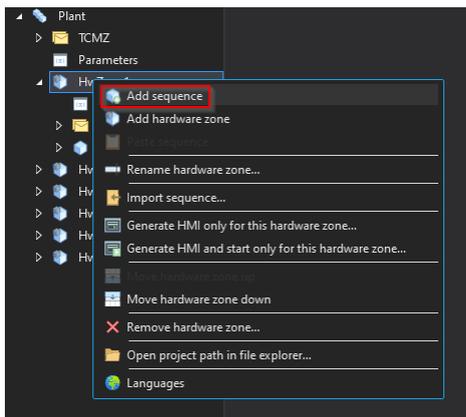
HMI

1.1.4 Szenario mehrere Sequenzen

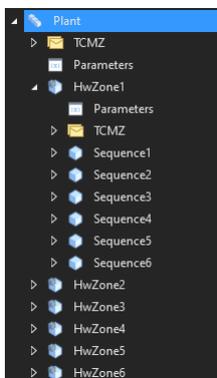
Hardwarezonen können mehrere Sequenzen enthalten. In diesen Sequenzen können Signale mit Zonen verbunden werden. Eine Plant kann dementsprechend folgende Struktur aufweisen:



Wenn mehrere Sequenzen eingefügt werden, erweitert sich die Struktur der Plant um die Elemente der zuvor genannten Sequence. Es gibt keine festgelegte Obergrenze, aber die Leistungsfähigkeit der verwendeten Hardware beeinflusst die Erweiterbarkeit.

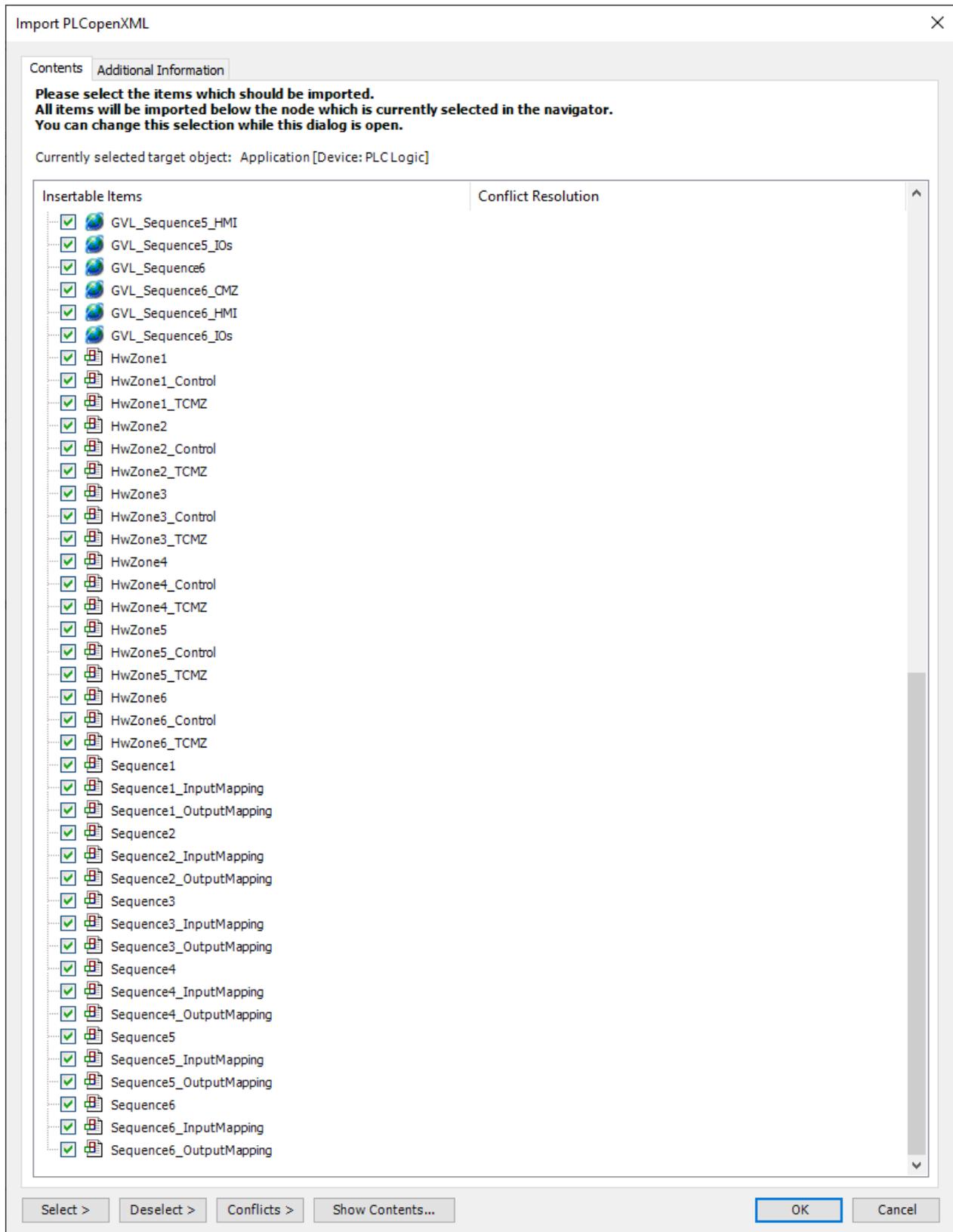


Studio



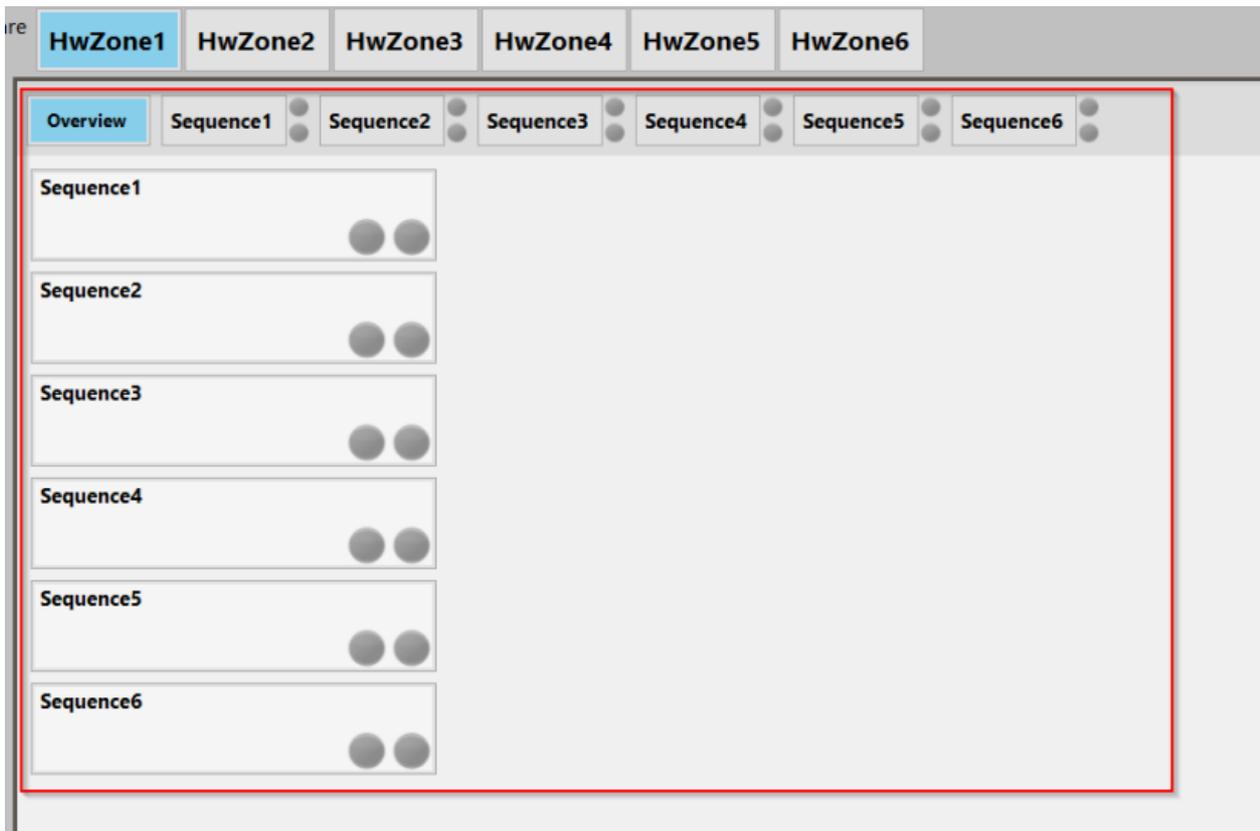
Studio

Folgende PLC Programm Elemente werden durch erneuten Export erstellt:



PLC

Folgende HMI Elemente werden erzeugt:



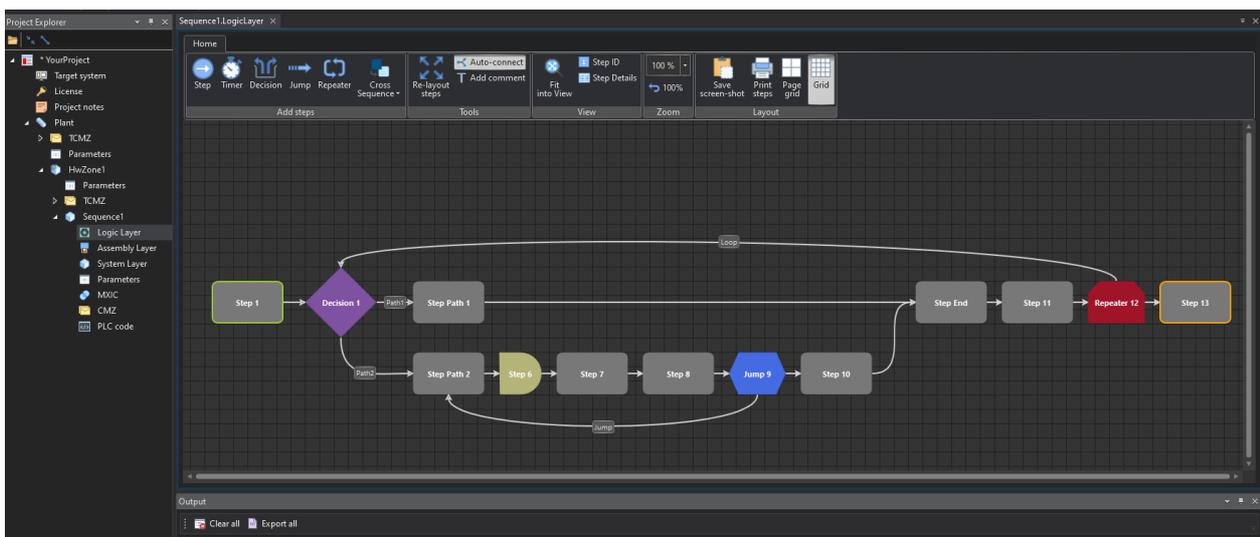
HMI

1.2 Logic anlegen

Ein Prozessmodell ist eine abstrakte Darstellung eines realen oder geplanten Prozesses, der von einer Fertigungsmaschine ausgeführt wird. Ein Prozessmodell besteht aus einer Reihe von Elementen, die die Aktivitäten, Ressourcen, Ereignisse und Entscheidungen beschreiben, die den Prozess ausmachen.

Die Logik einer Fertigungsmaschine kann in zwei Schichten unterteilt werden: den Logic Layer und den System Layer. Der Logic Layer enthält die fachlichen Anforderungen der Maschine. Der System Layer enthält die technischen Aspekte und die Schnittstellen der Maschine zu anderen Systemen.

Im Logic Layer wird der logische Prozess definiert. Um diesen zu beschreiben, stehen fünf Modellierungselemente zu Verfügung. Durch logische Elemente werden die Beziehungen der einzelnen Schritte zueinander beschrieben.

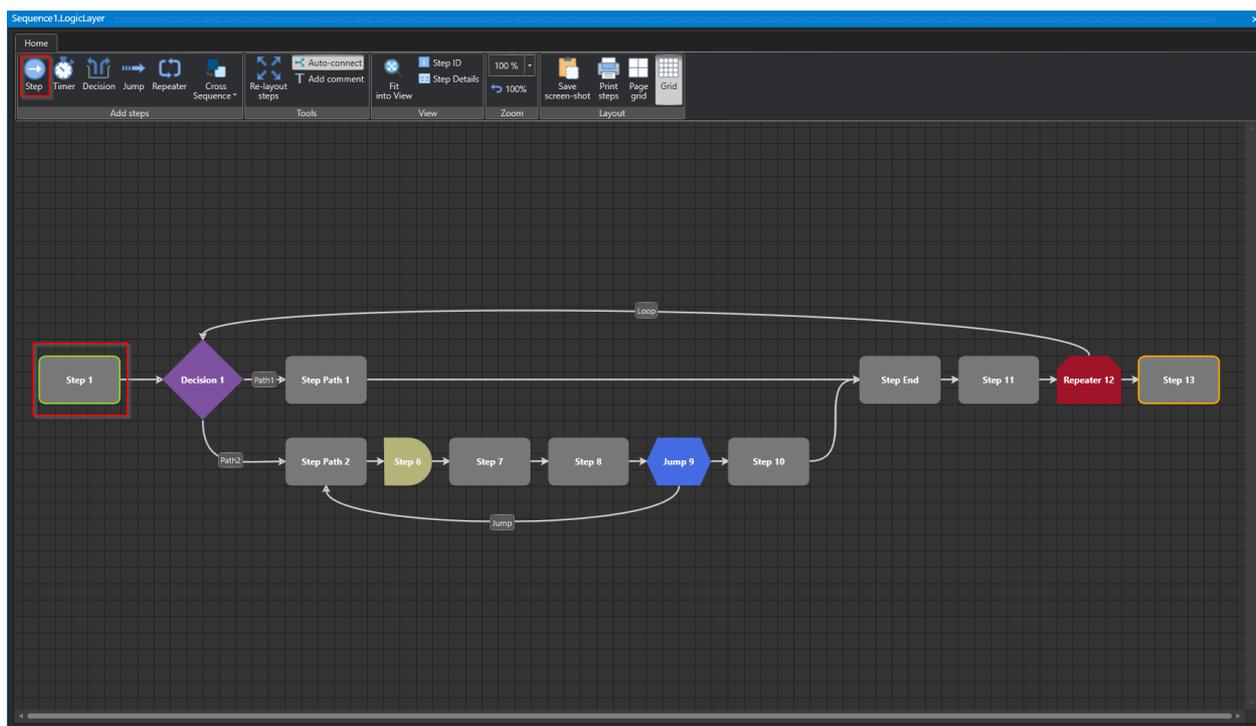


Studio

1.2.1 Elemente der Modellierung

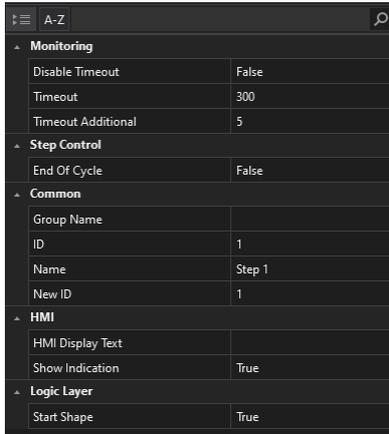
Step

Der Schritt ("Step") bildet das grundlegende Element für die Modellierung, mit dem die einzelnen diskreten Ablaufschritte oder Anlagenzustände beschrieben werden. Jeder Schritt erhält eine eindeutige ID. Ein Schritt wird über die Eigenschaft "Start Shape" als Ausgangszustand mit der ID 1 definiert.



Studio

Properties



The screenshot shows a dark-themed 'Properties' panel in the Selmo Studio. At the top, there is a search bar with 'A-Z' and a magnifying glass icon. Below are several expandable sections, each with a list of properties and their values:

Monitoring	
Disable Timeout	False
Timeout	300
Timeout Additional	5

Step Control	
End Of Cycle	False

Common	
Group Name	
ID	1
Name	Step 1
New ID	1

HMI	
HMI Display Text	
Show Indication	True

Logic Layer	
Start Shape	True

Studio

Monitoring

Die Schrittzeitüberwachung ist eine Funktion, welche die Ausführungszeit jedes Schrittes in einem Programm überwacht. Sie kann über die HMI aktiviert oder deaktiviert werden. Die Überwachung erfolgt zyklusgenau, d.h. die Zeit wird in jedem Zyklus gemessen und mit einem voreingestellten Grenzwert verglichen. Wenn die Zeit den Grenzwert überschreitet, wird ein Alarm ausgelöst. Die Schrittzeitüberwachung wird in einem Fenster in der HMI dargestellt, welches die aktuelle Zeit, den Grenzwert und den Status jedes Schrittes anzeigt. Es kann eine zusätzliche Zeit definiert werden, die bei der Ermittlung des Grenzwertes zur gemessenen Zeit addiert wird. Der Schrittzeitalarm wird ausgelöst, wenn die aktuelle Schrittzeit den Grenzwert, der sich aus der gemessenen Schrittzeit plus der zusätzlichen Schrittzeit bildet, überschreitet.

Disable Timeout

In diesem Abschnitt haben Sie die Möglichkeit, die Schrittzeitüberwachung für den markierten Schritt zu deaktivieren. Beachten Sie jedoch, dass die Schrittzeitüberwachung standardmäßig aktiviert ist.

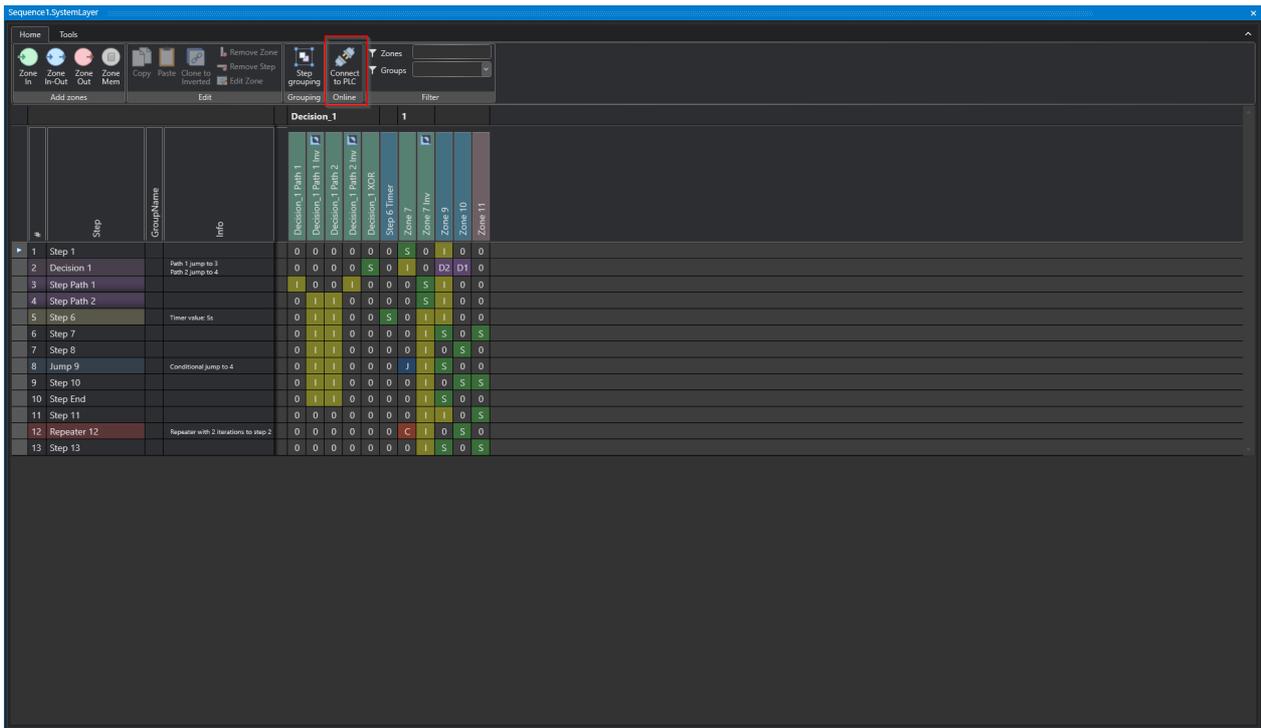
Timeout

In der HMI (Human-Machine Interface) wird eine Warnung angezeigt, wenn innerhalb eines bestimmten Zeitraums keine Zustandsänderung erfolgt. Bitte geben Sie an, ab welcher Zeitspanne ohne Zustandsänderung eine Warnung auf der HMI erscheinen soll.

Timeout Additional

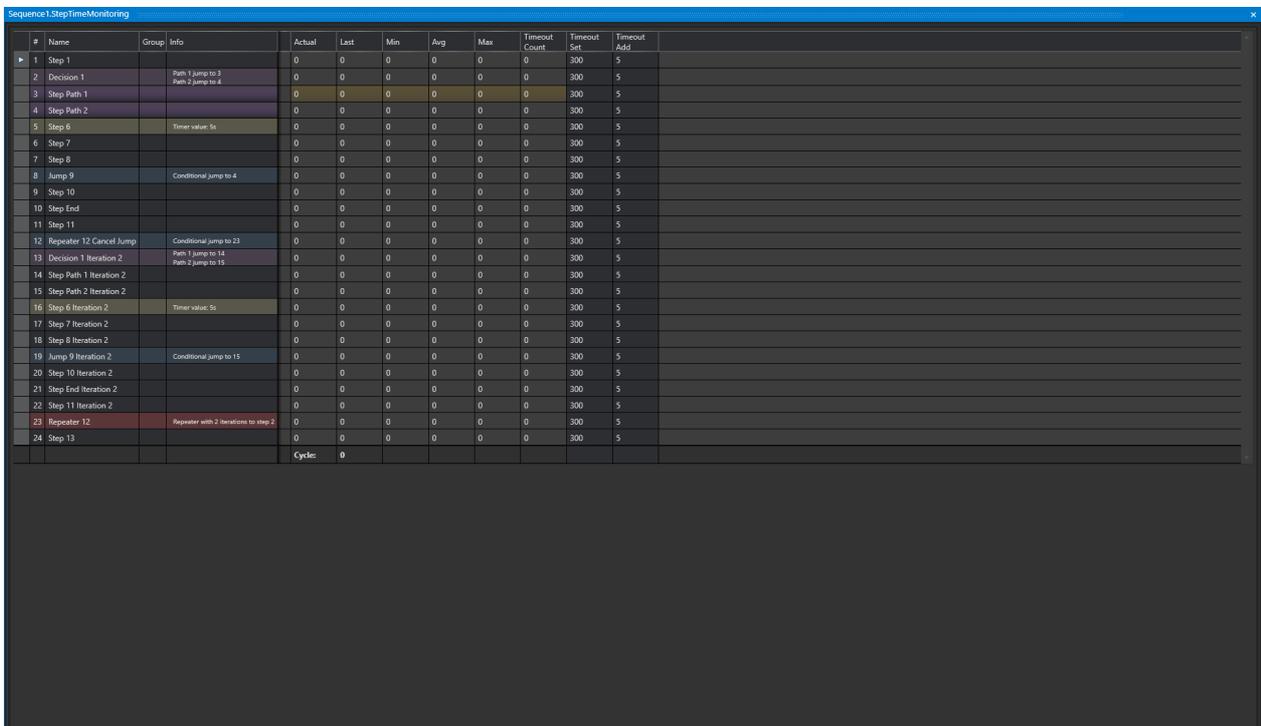
Im Teachmodus können die Schrittzeiten gemessen werden und diese Zeit wird zur gemessenen Zeit hinzugefügt. Falls ein Schritt die gemessene Zeit plus eine festgelegte Toleranzzeit überschreitet, wird eine Warnung auf der HMI ausgegeben. Bitte beachten Sie, dass die genauen Einstellungen und Parameter zur Messung der Schrittzeiten, der Toleranzzeit und der Warnung auf der HMI von der spezifischen Anwendung oder dem System abhängen und entsprechend konfiguriert werden müssen.

Um auf das Online Monitoring im Studio im System Layer zuzugreifen, können Sie die folgenden Symbole verwenden:



Studio

An dieser Stelle werden alle schrittbezogenen Daten in verschiedenen Formaten dargestellt.



Studio

Die Schrittzeitüberwachung ist auch in der HMI verfügbar.

Step Time Monitoring

Enable time monitoring teach mode

	Actual	Last	Min	Avg	Max	Timeout Count	Timeout	Timeout Add	Disable Timeout
Step 1	0.000	0.000	0.000	0.000	0.000	0	300	5	<input type="checkbox"/>
Step 2	0.000	0.000	0.000	0.000	0.000	0	300	5	<input type="checkbox"/>

HMI

nicht aktuell

Step Control

End Of Cycle (EOC) ist ein Betriebsmodus, der die Anlage in einen sicheren Zustand bringt, wenn die Produktion beendet ist. Die Anlage fährt bis zum Erreichen des definierten EOC Schritts, der je nach Anlagentyp variieren kann. Der EOC Modus kann über den Safetykeyswitch an der Bedienkonsole sicher eingeschaltet werden. Die Anzeige auf der HMI zeigt den aktuellen Status des EOC Modus an. Jede Sequence für sich hat einen eigenen EOC Schritt, der die entsprechenden Komponenten abschaltet oder in eine Parkposition bringt.

End Of Cycle

Mit dieser Einstellung kann festgelegt werden, dass die Sequence bei Halt nach Taktende sicher im angegebenen Step anhalten darf.



Studio

Common

Group Name

Es ist möglich, Steps über einen gemeinsamen Gruppennamen zu gruppieren.

ID

Die Step-ID ist die eindeutige ID innerhalb einer Sequence, diese wird durch das System abhängig vom Startschritt automatisch vergeben.

Name

Der Name wird sowohl in der HMI- als auch im System-Layer angezeigt. Wenn kein Text für die Eigenschaft "HMI Display Text" eingegeben wurde, wird dieser Name stattdessen verwendet.



HMI

New ID

HMI

HMI Display Text

Der "HMI Display Text" ist der Text, der in der HMI für diesen Schritt angezeigt wird und als Alternative zum Namen verwendet werden kann.

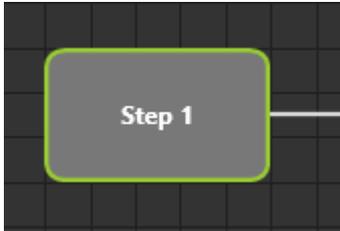


HMI

Show Indication

Logic Layer**Start Shape**

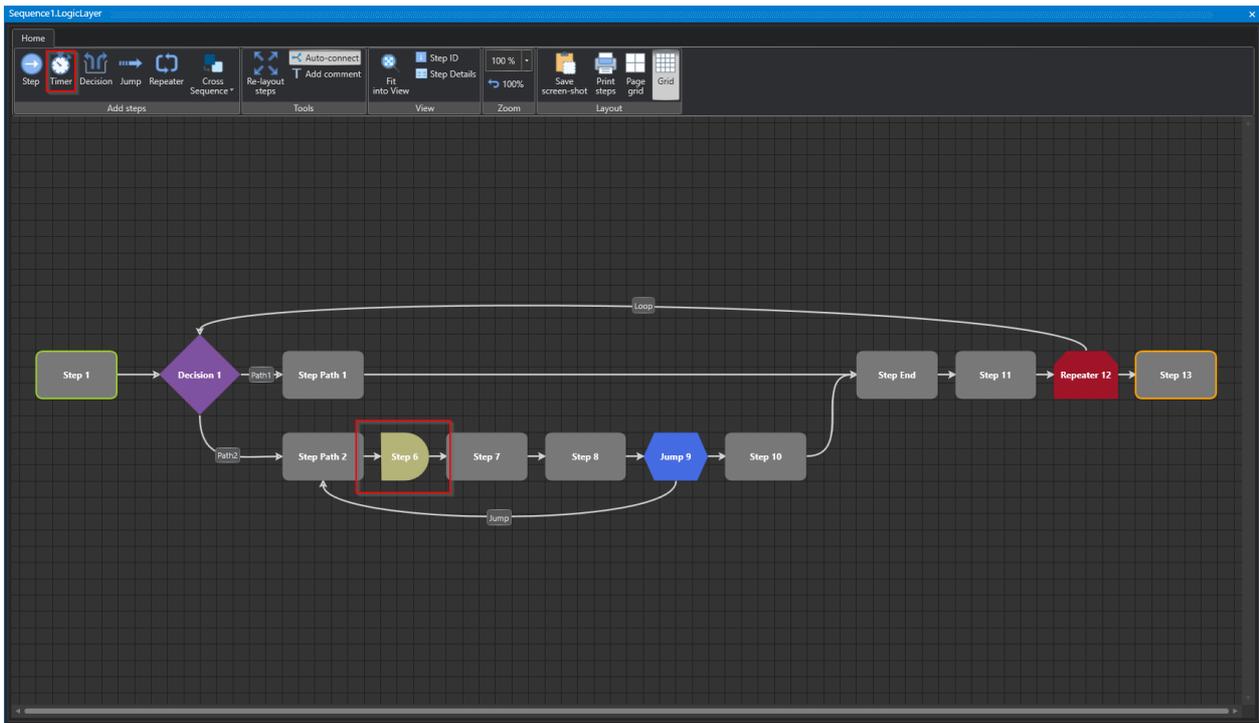
Legt den ersten Schritt der Sequence fest.



Studio

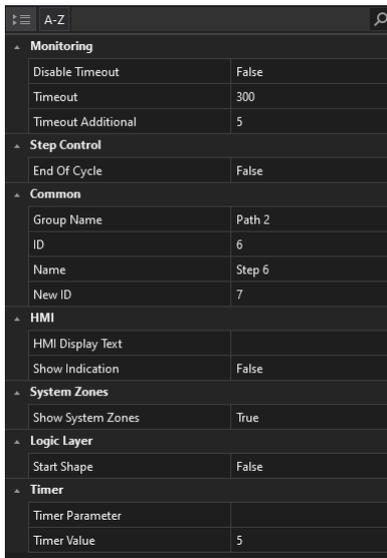
Timer Step

Das Timer-Element ermöglicht die Modellierung von zeitabhängigen Schritten. Ein Timer-Schritt kann erst in den folgenden Schritt wechseln, wenn eine bestimmte Zeit abgelaufen ist. Diese Zeit wird durch einen Timer-Parameter festgelegt. Ein Beispiel hierfür wäre ein zeitgesteuerter Rührvorgang in einem Tank, bei dem der Schritt erst dann zum nächsten Schritt übergeht, wenn die vordefinierte Rührzeit abgelaufen ist.



Studio

Properties



The screenshot shows a dark-themed 'Properties' panel in the Selmo Studio. At the top, there is a search bar with 'A-Z' and a magnifying glass icon. The panel is organized into several expandable sections, each with a small triangle icon to its left. The sections and their contents are as follows:

- Monitoring**
 - Disable Timeout: False
 - Timeout: 300
 - Timeout Additional: 5
- Step Control**
 - End Of Cycle: False
- Common**
 - Group Name: Path 2
 - ID: 6
 - Name: Step 6
 - New ID: 7
- HMI**
 - HMI Display Text: (empty)
 - Show Indication: False
- System Zones**
 - Show System Zones: True
- Logic Layer**
 - Start Shape: False
- Timer**
 - Timer Parameter: (empty)
 - Timer Value: 5

Studio

System Zones

System Zonen sind die Zonen, die automatisch mit einem Schritt generiert werden und für die Funktionsfähigkeit des Schrittes unverzichtbar sind.

Show System Zones

Es ist möglich, die Systemzonen im System Layer anzuzeigen oder zu verbergen.

Timer

Timer Parameter

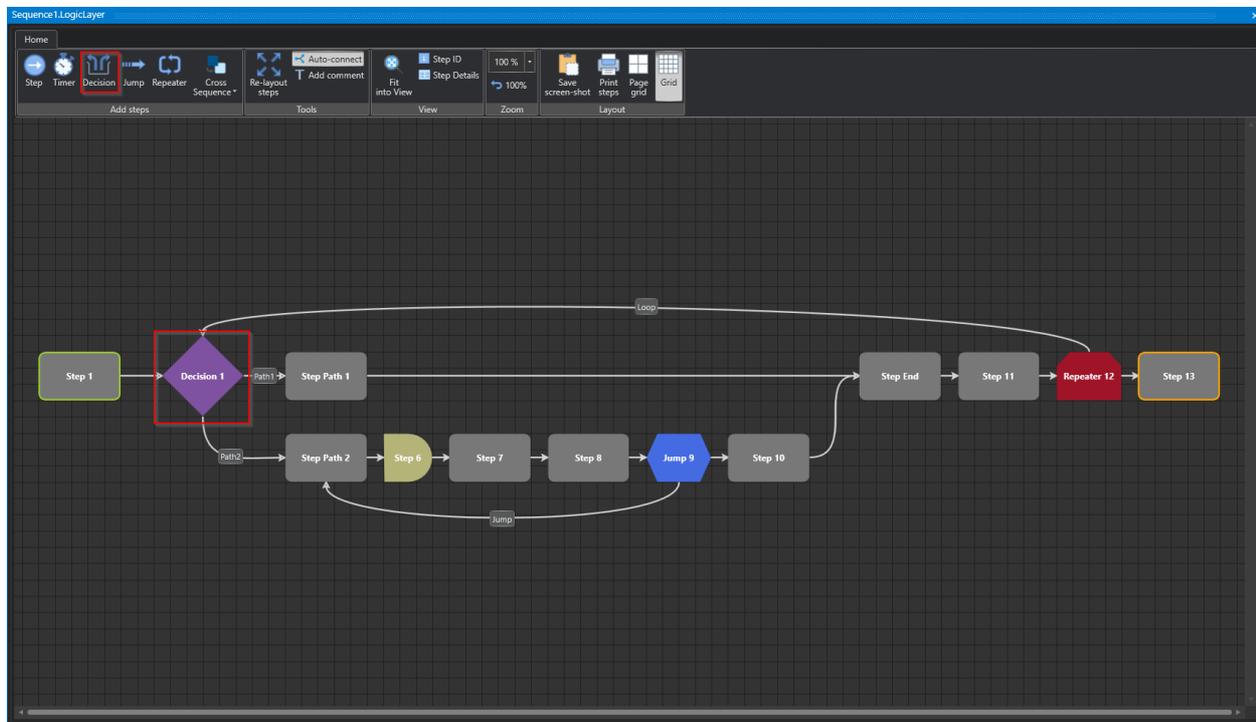
Es ist möglich, einen Parameter, der zuvor unter "Parameters" erstellt wurde, als Zeitbasis zu verwenden.

Timer Value

Es ist möglich, eine feste Zeit für diesen Schritt einzutragen.

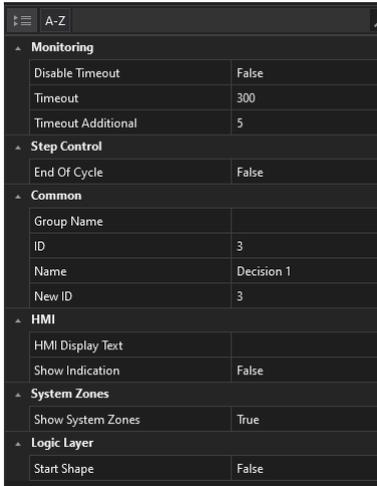
Decision Step

Der Decision-Schritt wird verwendet, um Entscheidungen im Ablauf zu treffen. Je nach Erfüllung eines Kriteriums wird entschieden, welcher Pfad in der Sequence fortgesetzt wird. Es können zwei Pfade ausgewählt werden, wie in der nachfolgenden Abbildung dargestellt.



Studio

Properties



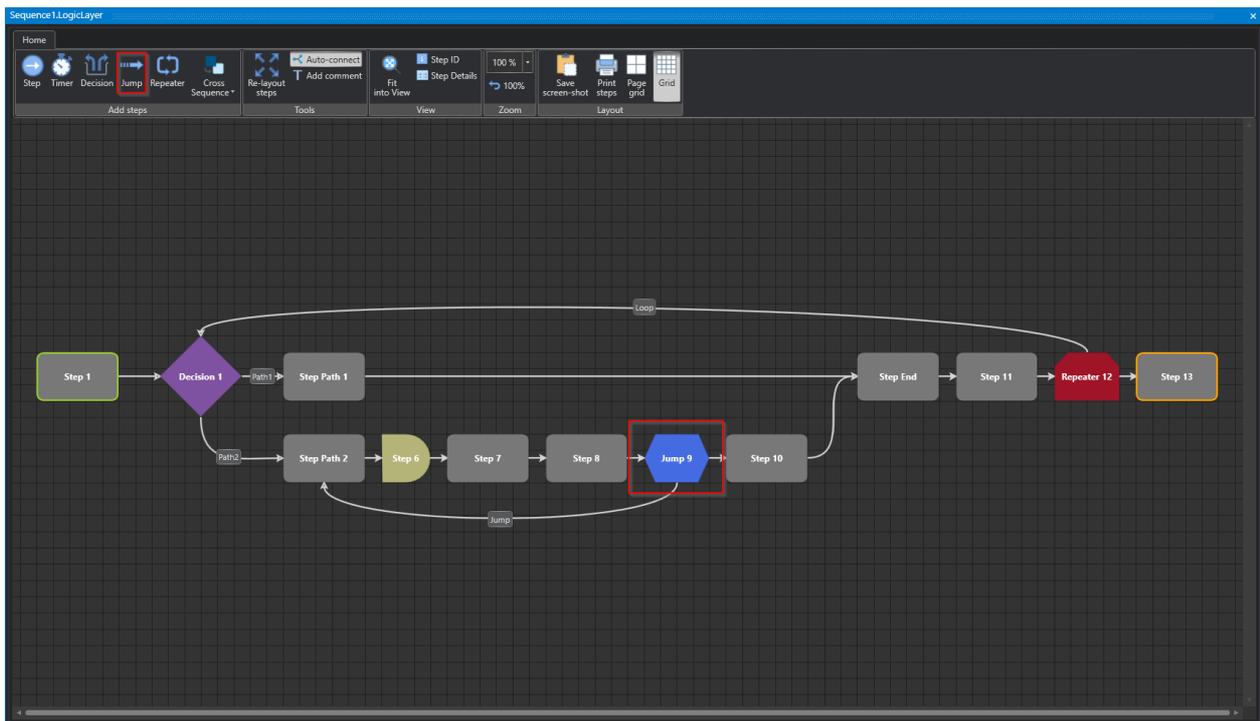
The screenshot shows a dark-themed 'Properties' panel in the Selmo Studio. At the top, there is a search bar with 'A-Z' and a magnifying glass icon. Below the search bar, the properties are organized into several expandable sections, each indicated by a small triangle icon. The sections and their contents are:

- Monitoring**
 - Disable Timeout: False
 - Timeout: 300
 - Timeout Additional: 5
- Step Control**
 - End Of Cycle: False
- Common**
 - Group Name: (empty)
 - ID: 3
 - Name: Decision 1
 - New ID: 3
- HMI**
 - HMI Display Text: (empty)
 - Show Indication: False
- System Zones**
 - Show System Zones: True
- Logic Layer**
 - Start Shape: False

Studio

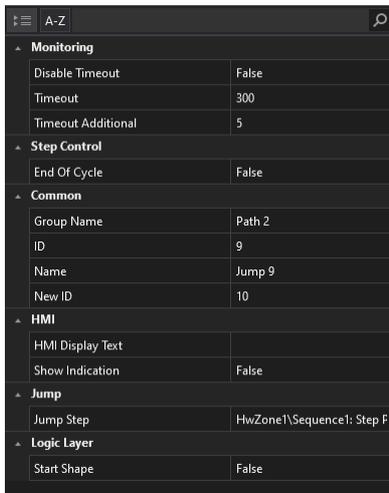
Jump Step

Ein Jump in der Prozess-Modellierung ermöglicht es, von einem bestimmten Schritt zu einem anderen Schritt in der Schrittkette zu springen, um den Programmfluss zu beeinflussen. Jumps können verwendet werden, um bedingte Abläufe zu implementieren oder um bestimmte Schritte zu überspringen oder zu umgehen, basierend auf speziellen Bedingungen oder Ereignissen. Die Verwendung von Jumps erfordert sorgfältige Planung und Modellierung, um sicherzustellen, dass der Ablauf der Schrittkette korrekt und zuverlässig funktioniert. Unsachgemäße Verwendung von Jumps kann zu Fehlern führen und die Programmlogik unvorhersehbar machen.



Studio

Properties



The screenshot shows a 'Properties' window in the Selmo Studio. The window has a title bar with 'A-Z' and a search icon. The content is organized into several expandable sections, each with a small triangle icon to its left. The sections and their properties are:

- Monitoring**
 - Disable Timeout: False
 - Timeout: 300
 - Timeout Additional: 5
- Step Control**
 - End Of Cycle: False
- Common**
 - Group Name: Path 2
 - ID: 9
 - Name: Jump 9
 - New ID: 10
- HMI**
 - HMI Display Text: (empty)
 - Show Indication: False
- Jump**
 - Jump Step: HwZone1\Sequence1: Step F
- Logic Layer**
 - Start Shape: False

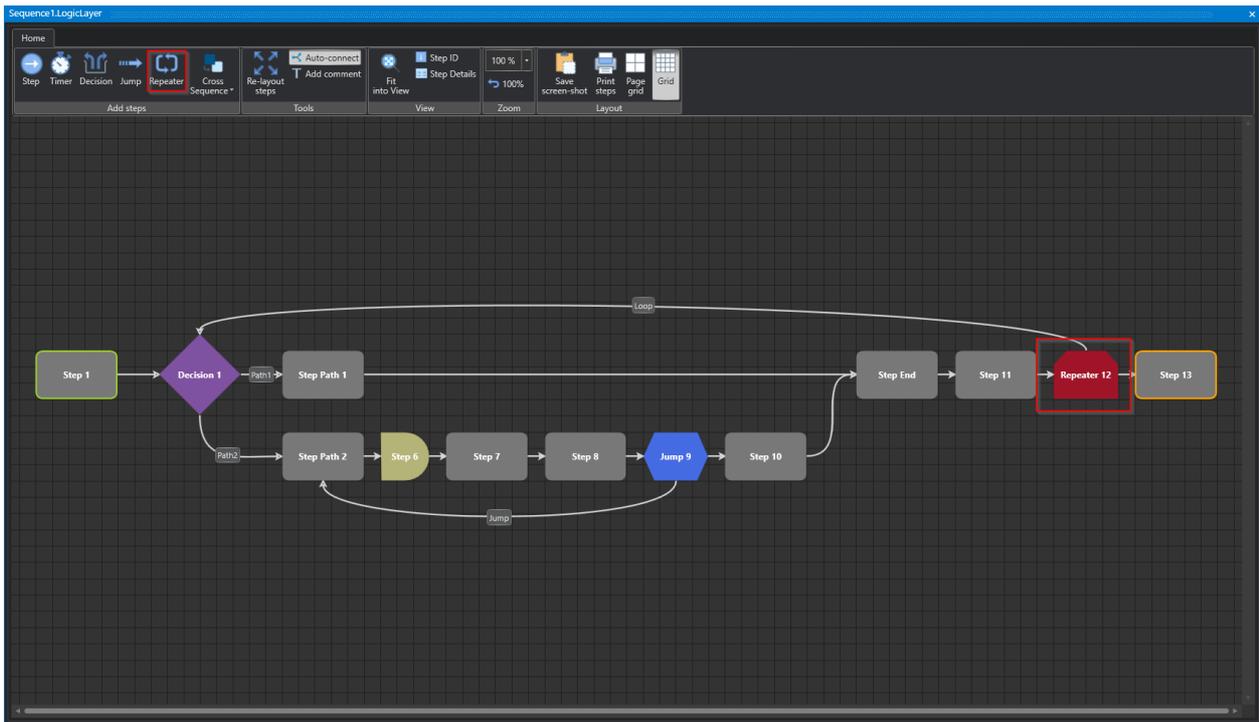
Studio

Jump**Jump Step**

Bestimmung des Sprungziels, also des Schritts, zu dem gesprungen wird, aufgrund von spezifischen Bedingungen, die im System hinterlegt sind.

Repeater Step

Der Repeater ermöglicht das wiederholte Ausführen von spezifischen Schritten, die zuvor modelliert wurden, indem er die in der Schleife definierten Schritte bis zur angegebenen Anzahl von Repeater-Iterationen kopiert. Dabei handelt es sich um eine endliche Abfolge von Iterationen und nicht um eine herkömmliche Schleifenprogrammierung. Der Repeater bietet außerdem die Möglichkeit, die Schleifenabarbeitung vor Erreichen der maximalen Iteration durch Verwendung einer Abbruchvariable zu beenden.



Studio

Properties

A-Z	
Monitoring	
Disable Timeout	False
Timeout	300
Timeout Additional	5
Step Control	
End Of Cycle	False
Common	
Group Name	
ID	13
Name	Repeater 12
New ID	14
HMI	
HMI Display Text	
Show Indication	False
Repeater	
Repeater Iterations	2
Repeater Step	HwZone1\Sequence1: Decision 1
System Zones	
Show System Zones	True
Logic Layer	
Start Shape	False

Studio

Repeater**Repeater Iterations**

Bestimmt die Anzahl der Duplikationen der Schritte in der Schleife.

Repeater Step

Bestimmung des Zielschritts, zu dem gesprungen wird.

1.2.2 Cross Sequence

Die Cross Sequence setzt sich aus Cross Sequence Sender und Cross Sequence Receiver zusammen. Die Cross Sequence wird verwendet, um die Freigaben zwischen zwei Sequences zu organisieren. Durch diese Freigaben ist es möglich Sequences zu synchronisieren, anzuordnen oder zu timen. Im folgenden Beispiel wird ein Cross Sequence Receiver in Sequence 2 durch einen Cross Sequence Sender freigegeben. Wird der Cross Sequence Sender erstellt werden zwei System Zonen angelegt, welche die Freigabe steuern. Dazu muss der Cross Sequence Receiver einem Master zugeordnet werden. Dazu wählt man die Eigenschaften des Cross Sequence Receiver an und weist einen Cross Sequence Sender aus. In diesem Beispiel wurde der Cross Sequence Receiver der Sequence 1 ausgewählt. Die Freigabe des Cross Sequence Receiver Getter erfolgt von dem ausgewählten Cross Sequence Sender, siehe nachfolgendes Bild. Es gibt noch die Möglichkeit einen Cross Sequence Multi Sender für die Freigabe mehrere Cross Sequence Sender zu erstellen. Im System Layer wird im Infofeld die entsprechende Zuordnung eingetragen.

Wichtige Grundlagen

Jeder **Receiver Getter** ist mit einem Receiver Setter automatisch verknüpft.

Jeder **Sender Setter** ist mit einem Sender Getter automatisch verknüpft.

Ein **Receiver Getter** kann (nur) einen Sender Setter zur Freigabe auswählen. Automatisch wird die Rückmeldung des Receiver Setter den Sender Getter zugeordnet.

Einem **Sender Setter** können (mehrere) Receiver Getter zur Freigabe zugeordnet werden. Automatisch wird die Rückmeldung des Receiver Setter den Sender Getter zugeordnet.

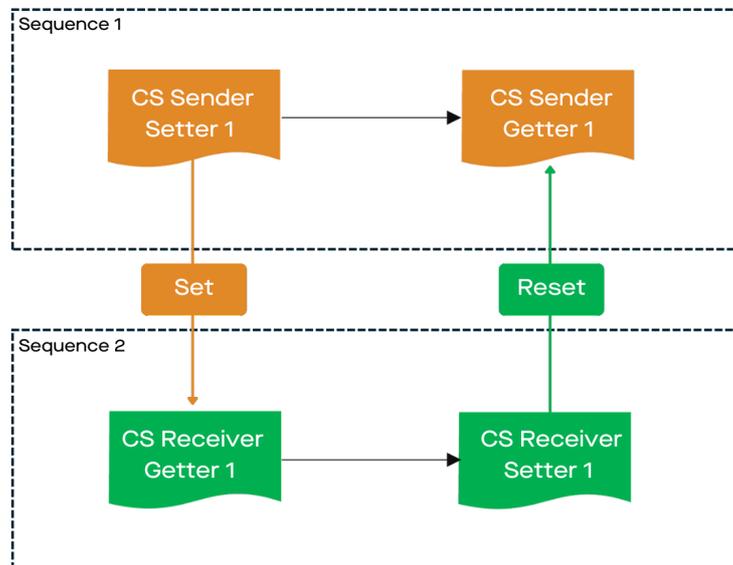
Jeder **Multi Sender Setter** ist mit einem Multi Sender Getter automatisch verknüpft.

Ein **Multi Sender Setter** kann (nur) einen Sender Setter zur Freigabe auswählen. Automatisch wird die Rückmeldung des Receiver Setter den Sender Getter zugeordnet. Der Sender Getter meldet den Multi Sender Getter zurück.

Ein **Sender Setter** kann mehrfach durch den **Multi Sender Setter** freigegeben werden.

Cross Sequence Sender-Receiver Freigabe

Die Cross Sequence Sender-Receiver Freigabe wird anhand folgendem Bild dargestellt:



Der Ablauf im Detail erfolgt wie in folgenden Schritten erklärt:

Schritt 1 in Sequence 1: Sender Setter 1 gibt den zugeordneten Receiver Getter 1 frei. Schrittweilerschaltung.

Schritt 1 in Sequence 2: Warten auf Freigabe von Sender Setter 1.

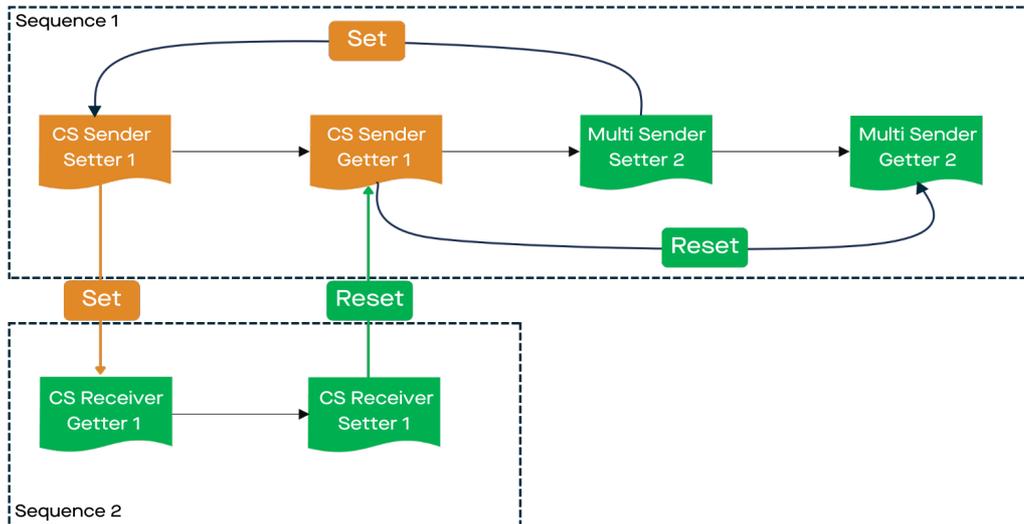
Schritt 1 in Sequence 2: Freigabe von Sender Setter 1 erhalten. Schrittweilerschaltung.

Schritt 2 in Sequence 1: Warten auf Freigabe Receiver Getter 1 der Sequence 2.

Schritt 2 in Sequence 2: Rückmeldung von Receiver Setter 1 an Sender Getter 1. Schrittweilerschaltung.

Schritt 2 in Sequence 1: Freigabe von Receiver Getter 1 erhalten. Schrittweilerschaltung.

Cross Sequence Multi Sender-Receiver Freigabe



(Gekürzte Schrittfolge) Der Verlauf der beiden Cross Sequences mit Multi Sender Freigabe sieht wie folgt aus:

Schritt 1 & 2 in Sequence 1 & 2 erfolgen, wie es im Bild Cross Sequence Multi Sender-Receiver Freigabe beschrieben ist.

Schritt 3 in Sequence 1: Multi Sender Setter 1 gibt den zugeordneten Sender Setter 1 frei. Schrittweilerschaltung.

Es beginnt die Schrittkette der Sender Receiver Freigabe erneut.

Rückmeldung von Receiver Setter 1 an Sender Getter 1. Rückmeldung von Sender Getter 1 an Multi Sender Getter 2. Schrittweilerschaltung

Schritt 4 in Sequence 1: Warten auf Freigabe von Sender Getter 1. Schrittweilerschaltung

(Vollständige Schrittfolge) Der Verlauf der beiden Cross Sequences mit Multi Sender Freigabe sieht wie folgt aus:

Schritt 1 in Sequence 1: Sender Setter 1 gibt den zugeordneten Receiver Getter 1 frei. Schrittweilerschaltung.

Schritt 1 in Sequence 2: Warten auf Freigabe von Sender Setter 1. Freigabe von Sender Setter 1 erhalten. Schrittweilerschaltung.

Schritt 2 in Sequence 1: Warten auf Freigabe Receiver Getter 1 der Sequence 2. Freigabe von Receiver Getter 1 erhalten. Schrittweilerschaltung.

Schritt 2 in Sequence 2: Rückmeldung von Receiver Setter 1 an Sender Getter 1. Schrittweilerschaltung.

Schritt 3 in Sequence 1: Multi Sender Setter 1 gibt den zugeordneten Sender Setter 1 frei. Schrittweilerschaltung.

Schritt 1 in Sequence 1: Sender Setter 1 gibt den zugeordneten Receiver Getter 1 frei. Schrittweilerschaltung.

Schritt 1 in Sequence 2: Warten auf Freigabe von Sender Setter 1. Freigabe von Sender Setter 1 erhalten. Schrittweilerschaltung.

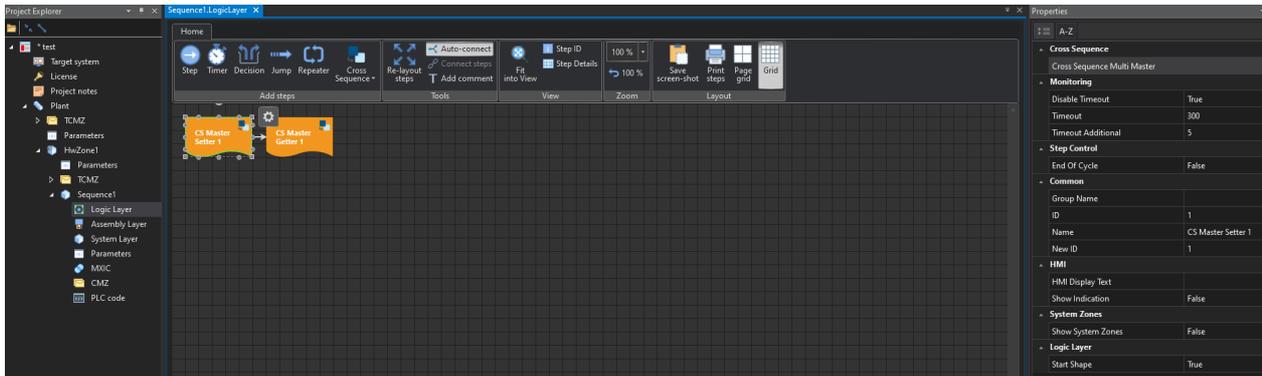
Schritt 2 in Sequence 1: Warten auf Freigabe Receiver Getter 1 der Sequence 2. Freigabe von Receiver Getter 1 erhalten. Schrittweilerschaltung.

Schritt 2 in Sequence 2: Rückmeldung von Receiver Setter 1 an Sender Getter 1. Rückmeldung von Sender Getter 1 an Multi Sender Getter 2. Schrittweilerschaltung.

Schritt 4 in Sequence 1: Warten auf Freigabe von Sender Getter 1. Schrittweilerschaltung.

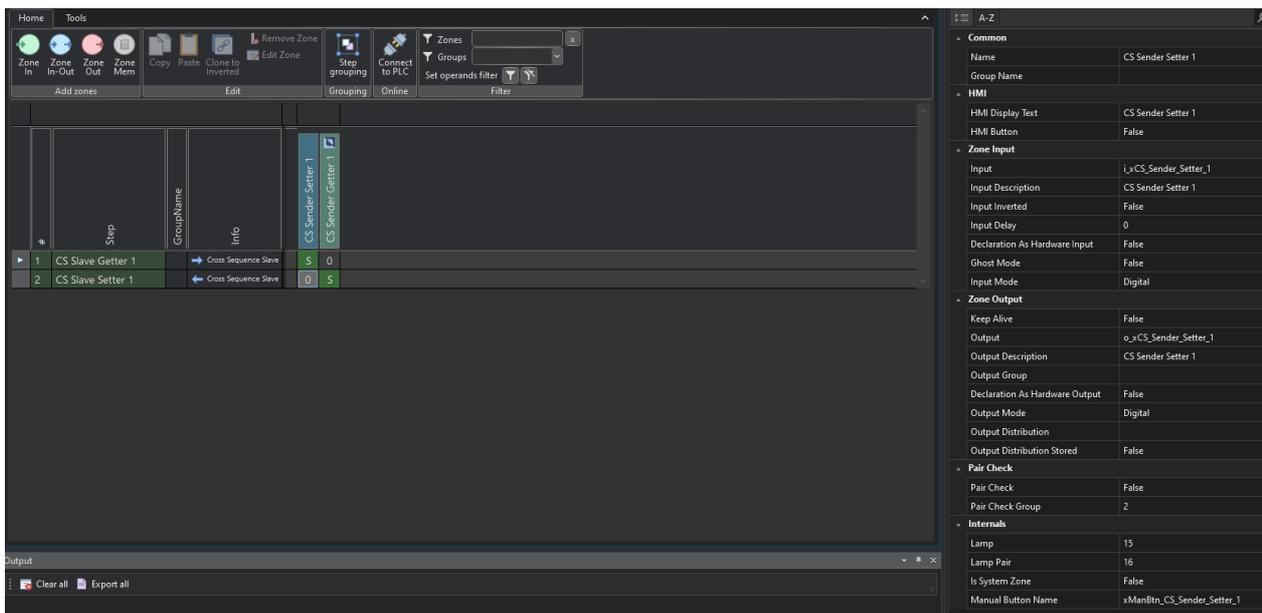
Die Cross Sequence setzt sich aus Cross Sequence Sender und Cross Sequence Receiver zusammen. Die Cross Sequence wird verwendet, um die Freigaben zwischen zwei Sequenzen zu organisieren. Durch diese Freigaben ist es möglich Sequenzen zu synchronisieren, anzuordnen oder zu timen.

In folgendem Beispiel wird ein Cross Sequence Receiver in Sequence 2 durch einen Cross Sequence Sender freigegeben.



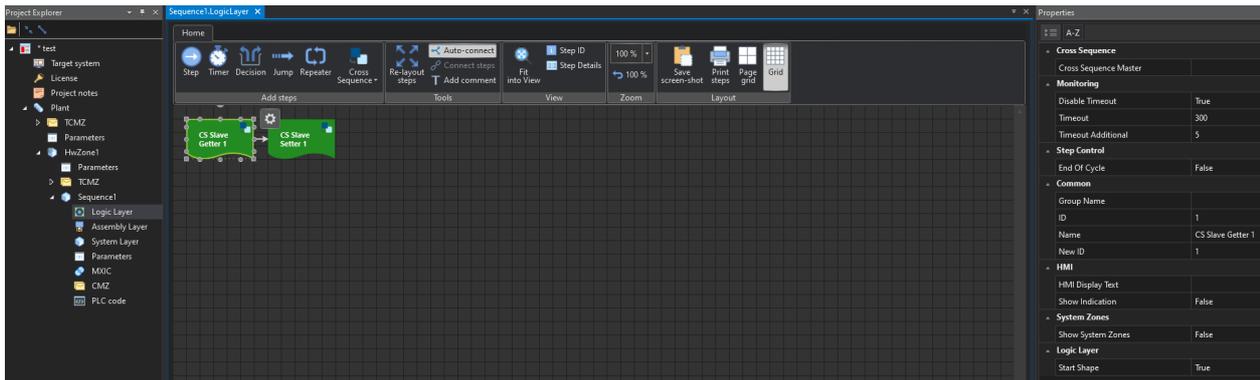
Studio

Wird der Cross Sequence Sender erstellt, werden zwei System Zonen angelegt, welche die Freigabe steuern.



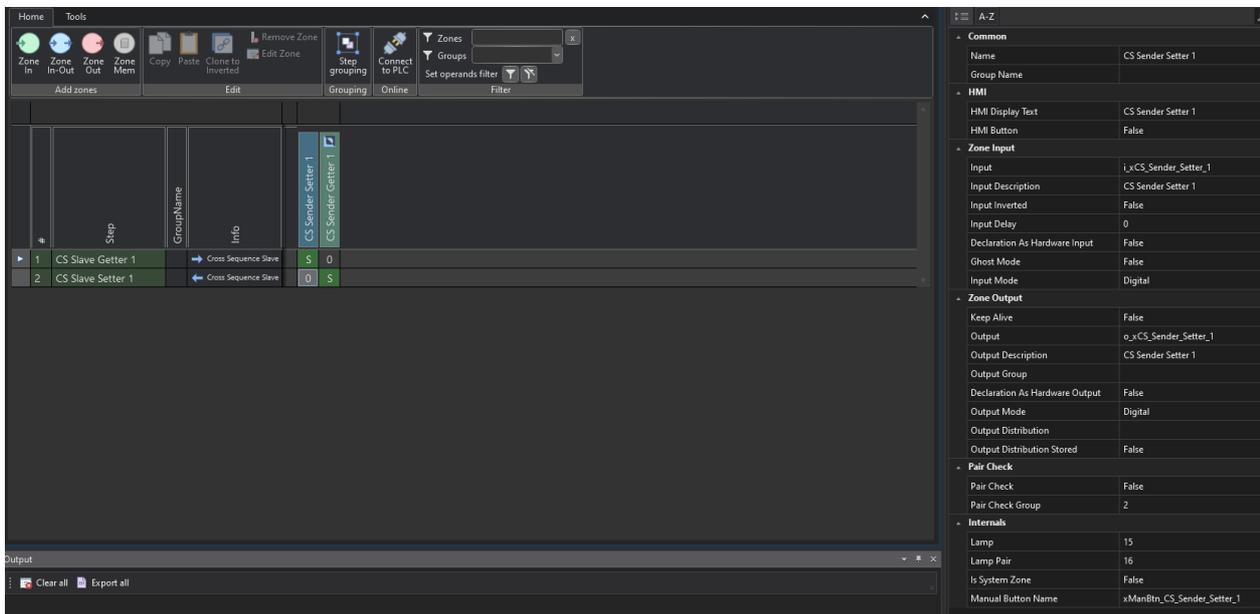
Studio

Dazu muss der Cross Sequence Receiver einem Sender zugeordnet werden. Dazu wählt man die Eigenschaften des Cross Sequence Receiver an und weist einen Cross Sequence Sender zu. In diesem Beispiel wurde der Cross Sequence Sender der Sequence 1 zugewiesen.

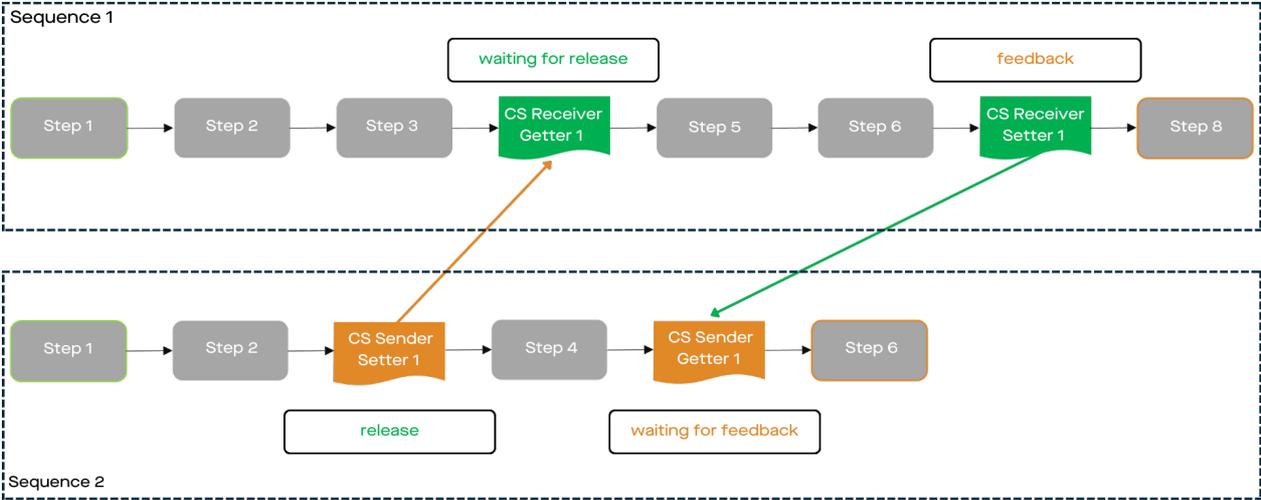


Studio

Die Freigabe des Cross Sequence Receiver Getter erfolgt von dem ausgewählten Cross Sequence Senders, siehe nachfolgendes Bild.



Studio



Studio

Die Cross Sequence wird über Setter/Getter Logik gesteuert:

- **Cross Sequence Sender Setter**

setzt das Freigabe Bit

- **Cross Sequence Sender Getter**

wartet, bis das Freigabe Bit vom Receiver zurückgesetzt wird

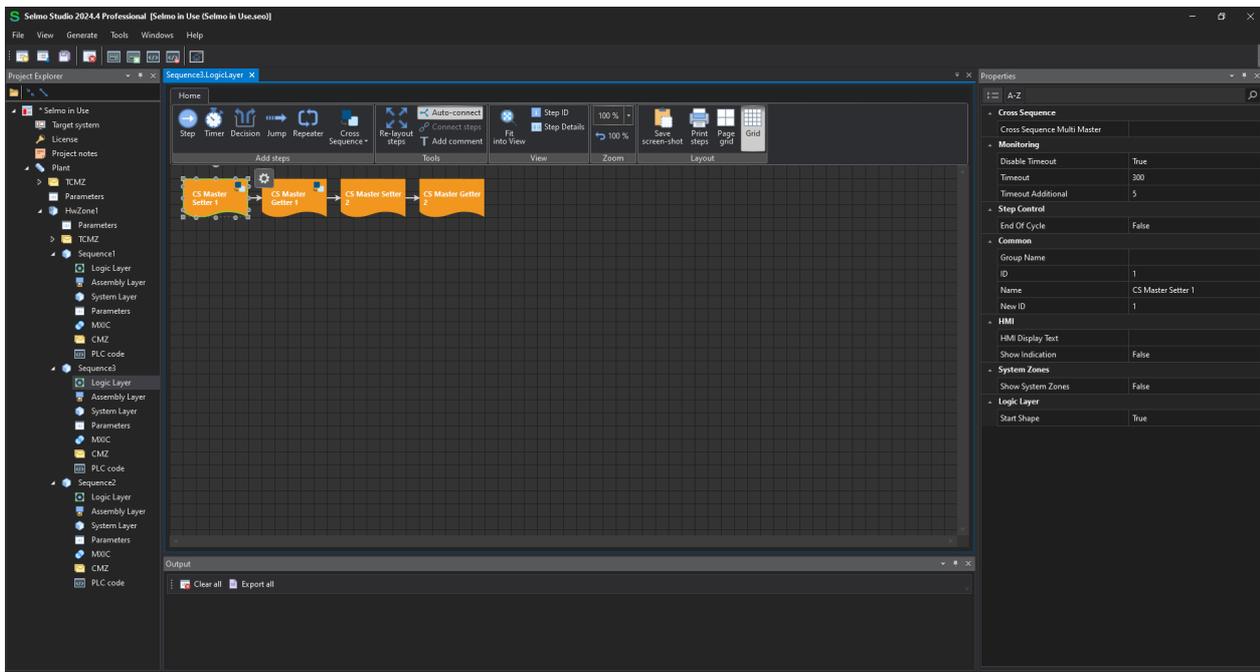
- **Cross Sequence Receiver Getter**

wartet auf das Freigabe Bit vom Sender

- **Cross Sequence Receiver Setter**

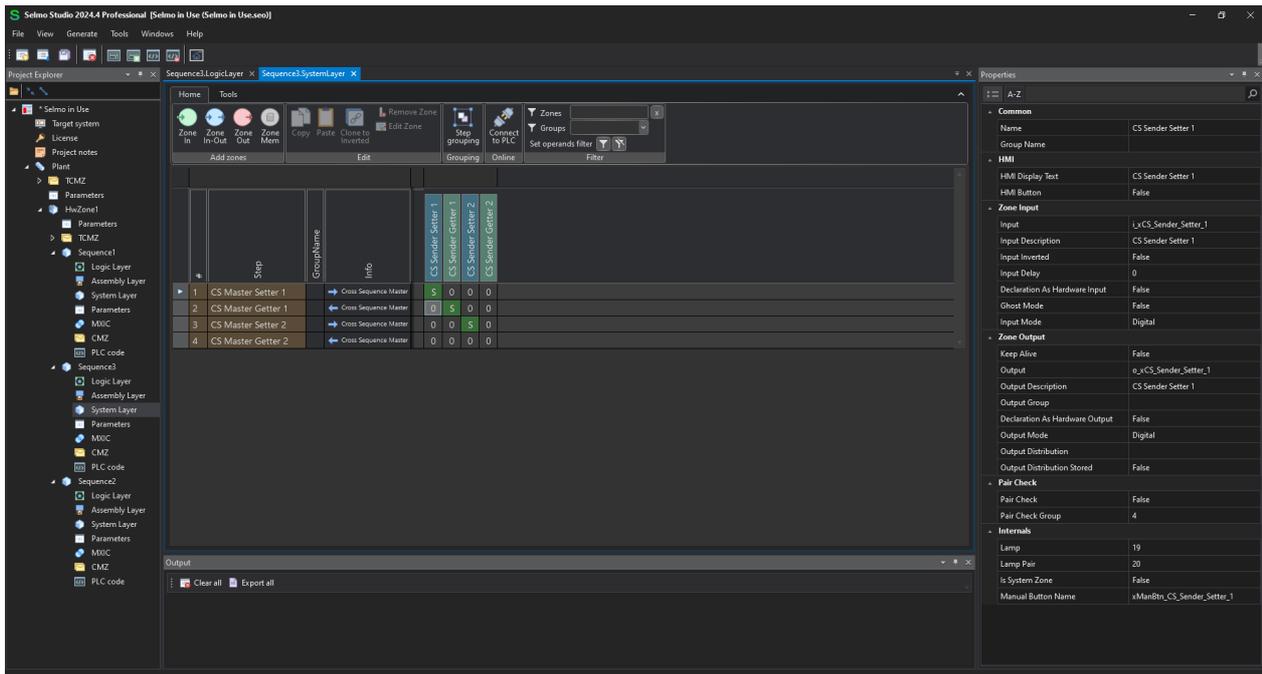
setzt das Freigabe Bit zurück

Es gibt noch die Möglichkeit einen Cross Sequence Multi Sender für die Freigabe mehrerer Cross Sequence Sender zu erstellen.



Studio

Im System Layer wird im Infobereich die entsprechende Zuordnung eingetragen.

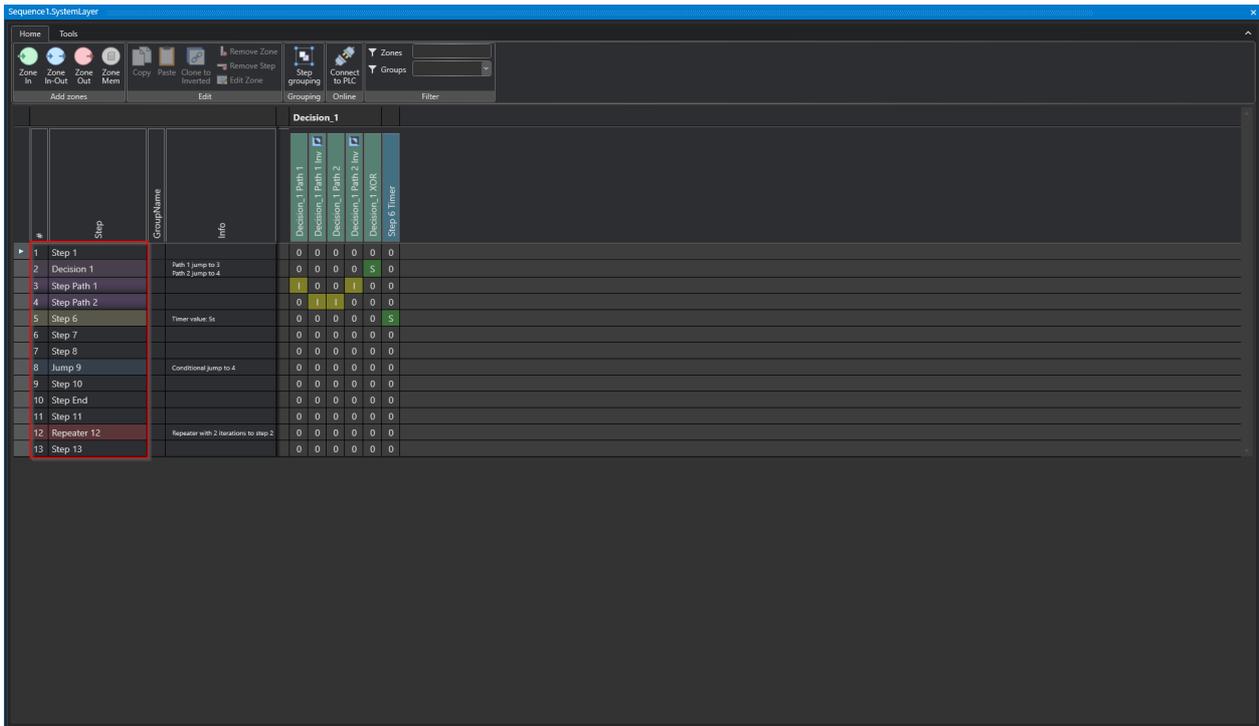


Studio

1.2.3 Umwandlung in Systemlayer

Der Logic-Layer wird im System-Layer als Spalte links dargestellt. Im System-Layer wird das Programm modelliert, die Zonen definiert und die Maschinenzustände bzw. Überwachungen festgelegt.

In jedem Schritt werden die Zustände der Zonen mit den Operanden '0', 'I', 'S', 'M', 'D1', 'D2', 'J', 'C' beschrieben.



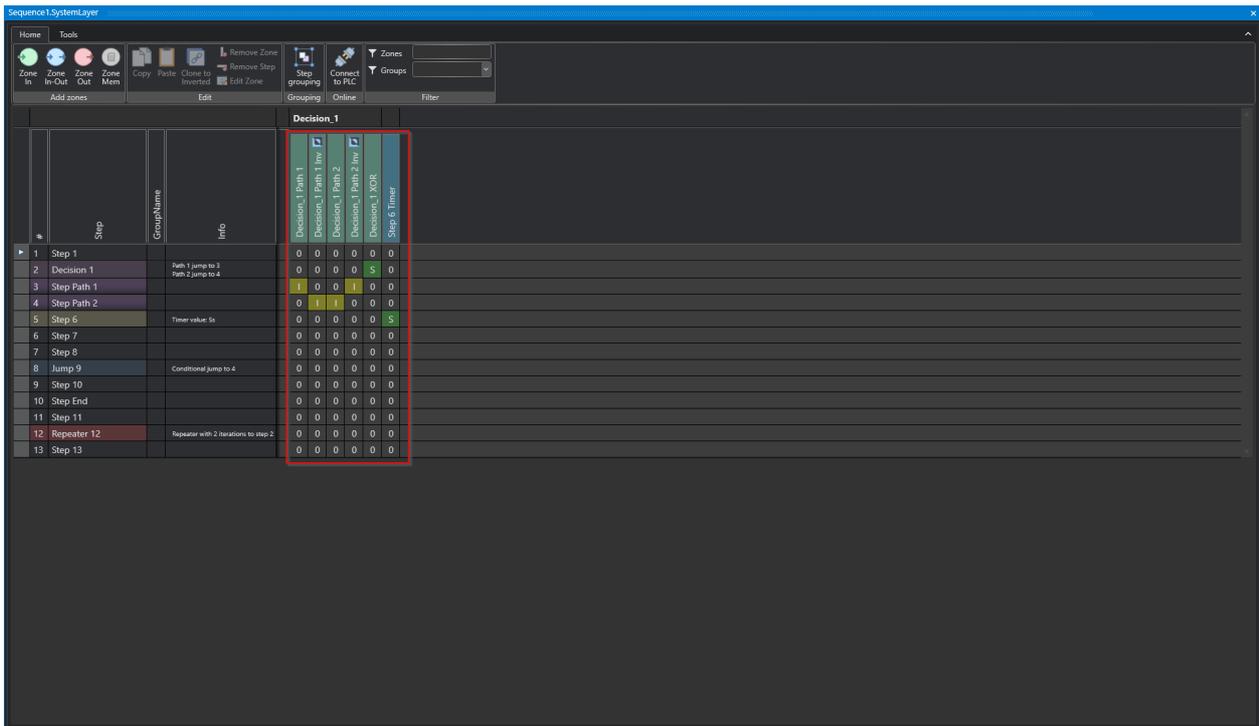
The screenshot shows the 'Sequence1 System Layer' window in the Selmo Studio. The interface includes a toolbar with various actions like 'Zone In', 'Zone Out', 'Zone Mem', 'Copy', 'Paste', 'Clone to Inverted', 'Edit Zone', 'Remove Zone', 'Remove Step', 'Step grouping', 'Connect to PLC', 'Zones', and 'Groups'. Below the toolbar is a table with the following columns: #, Step, GroupName, Info, Decision_1 Path 1, Decision_1 Path 2, Decision_1 Path 3, Decision_1 XOR, and Step 6 Timer. The table contains 13 rows of data, with the first row highlighted in red.

#	Step	GroupName	Info	Decision_1 Path 1	Decision_1 Path 2	Decision_1 Path 3	Decision_1 XOR	Step 6 Timer
1	Step 1			0	0	0	0	0
2	Decision 1		Path 1 jump to 3 Path 2 jump to 4	0	0	0	0	S
3	Step Path 1			1	0	0	1	0
4	Step Path 2			0	1	1	0	0
5	Step 6		Timer value 5s	0	0	0	0	S
6	Step 7			0	0	0	0	0
7	Step 8			0	0	0	0	0
8	Jump 9		Conditional jump to 4	0	0	0	0	0
9	Step 10			0	0	0	0	0
10	Step End			0	0	0	0	0
11	Step 11			0	0	0	0	0
12	Repeater 12		Repeater with 2 iterations to step 2	0	0	0	0	0
13	Step 13			0	0	0	0	0

Studio

1.3 Signale anlegen

Der System-Layer ist die Ebene, die für die Verarbeitung und Steuerung von Prozesssignalen zuständig ist. Die Prozesssignale sind binäre Werte, die den Zustand von Sensoren oder Aktoren repräsentieren. Der System-Layer verwendet bitgesteuerte Logikfunktionen, um die Prozesssignale zu verknüpfen und entsprechende Ausgänge zu steuern. Der System-Layer ist in Zonen unterteilt, die jeweils einen Teil des Gesamtprozesses überwachen oder steuern.



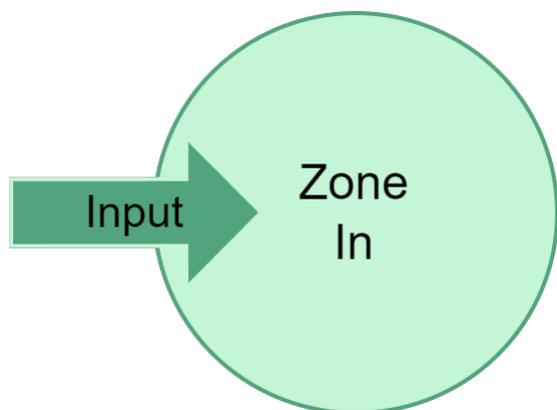
Studio

Eine mögliche Unterteilung von Signalen in einer automatisierten Anlage ist die Unterscheidung zwischen ständig Überwachten Signalen (Constantly Monitored Zone CMZ) und Prozesssignalen. Die CMZs umfassen alle Signale, die für die Sicherheit und den Schutz der Anlage relevant sind, wie z.B. Sicherungen, Überwachungen, Not-Aus-Schalter etc. Diese Signale werden ständig überwacht und ausgewertet, um bei Bedarf eine Abschaltung oder eine Warnung auszulösen. Die Prozesssignale hingegen sind alle Signale, die den eigentlichen Ablauf des Prozesses steuern oder regeln, wie z.B. Sensoren, Aktoren, Messwerte etc. Die Unterteilung von Signalen in CMZ und Prozesssignale hilft, die Anforderungen an die Sensorik, die Signalverarbeitung und die Kommunikation zu definieren und zu vereinfachen.

1.3.1 Zone

Zonen ermöglichen es, verschiedene Bereiche eines Systems zu definieren und zu kontrollieren. Zonen können je nach Anwendung und Anforderung unterschiedlich konfiguriert werden. Dabei ist es wichtig, die Funktionen und Eigenschaften der verschiedenen Zonentypen zu verstehen. In diesem Abschnitt werden die Zonentypen In, InOut, Out und Mem erklärt und ihre Anwendungsmöglichkeiten aufgezeigt. Außerdem werden die Operanden beschrieben, die in einer Zone verwendet werden können, um bestimmte Aktionen auszulösen oder zu überprüfen.

Sensor einbinden



Um einen Sensor in Ihre Anwendung einzubinden, müssen Sie die Zone-In verwenden. Der Sensor sendet nur ein Input-Signal an das System, dieses wird mit der Zone-In und der negierten Zone-In auf high und low überwacht.

Beispiele für Sensoren die mit einer Zone-In eingebunden werden können:



Induktiver Sensor



Schalter



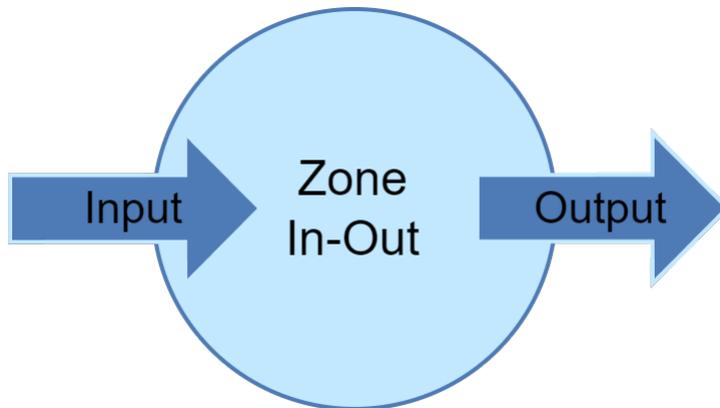
Taster



Endlagensensor Lichtschrank



Aktor mit Rückmeldung einbinden



Die Zone-InOut schaltet einen Ausgang und erwartet sich eine Rückmeldung, dass die durch den Ausgang angesteuerte Funktion ausgeführt wurde.

Beispiele:

- Zylinder Ventile "Ausfahren" wird geschaltet: Die Zone-InOut erwartet sich als Input den Endlagensensor des Zylinders
- Servomotor "Fahre auf Position 1" wird geschaltet: Die Zone-InOut erwartet sich als Input das Signal, dass sich die Achse in Position 1 befindet
- Frequenzumrichter "Geschwindigkeit 1" wird geschaltet: Zone-InOut erwartet sich als Input das Signal, dass der Frequenzumrichter die Geschwindigkeit erreicht hat

Beispiele für Sensoren die mit einer Zone-InOut eingebunden werden können:



Frequenzumrichter



Zylinder



Servomotor

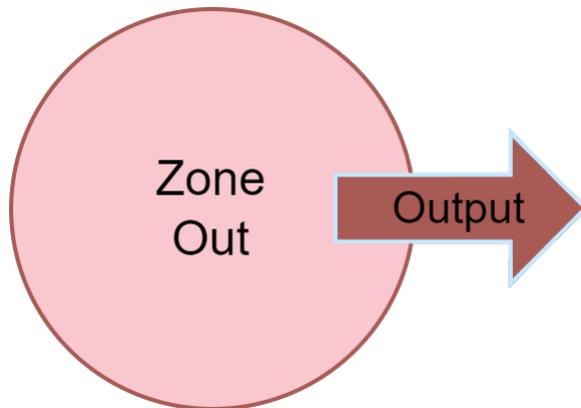


Schrittmotor



Regelung

Aktor ohne Rückmeldung einbinden



Die Zone-Out schaltet Aktoren und erwartet kein Feedback, sprich sie bedient reine Schaltlogik. Dieser Zonen-Typ wird hauptsächlich für Leuchten und Signale verwendet, jedoch auch für Aktoren, die kein Feedback besitzen, dies sollte aber nach Möglichkeit durch technische Möglichkeiten vermieden werden, da dies zu Unsicherheiten im Prozess führt.

Beispiele für Sensoren die mit einer Zone-Out eingebunden werden können:



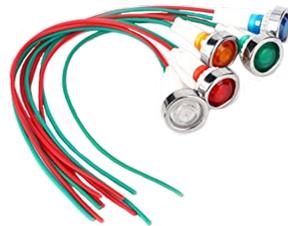
Alarmleuchte



Signalhorn



Stableuchte



Signallampen



Zylinder ohne Rückmeldung

Merker einbinden

Die Zone-Mem wird als Merker verwendet, um sich ausgeführte Prozessschritte zu merken. Sie besteht immer aus zwei Zonen, die Erste setzt und überprüft den Merker auf high, die Zweite setzt den Merker zurück und überprüft den Merker auf low.

1.3.1.1 Zone In

Die **Zone-In** kann verwendet werden, um Eingangsinformationen von z.B. Sensoren zu verarbeiten. Ist in einem Schritt der Sequence-Check in einer Zone-In gesetzt, so wird ein Zustandswechsel dieser Zone-In diesem Schritt erwartet. Ein Interlock-Check einer Eingangszone stellt sicher, dass der erwartete Zustand dieser Zone-In diesem Schritt erhalten bleibt. In der Praxis kann z.B. ein Schalter durch eine Zone-In überwacht werden. Wird ein Schaltvorgang gefordert, muss das Signal des Schalters mit der Zone-In verknüpft werden und im entsprechenden Schritt ein Sequence-Check gesetzt sein.

Zone-In anlegen

Um eine Zone-In anzulegen, muss auf das entsprechende Symbol geklickt werden.

The screenshot shows the Selmo Studio software interface. At the top, there is a toolbar with various tools including 'Zone In', 'Zone In-Out', 'Zone Out', 'Zone Mem', 'Copy', 'Paste', 'Clone to Inverted', 'Edit Zone', 'Remove Zone', 'Step grouping', 'Connect to PLC', and 'Filter'. Below the toolbar is a ladder logic diagram with a table of steps and their associated logic elements.

#	Step	GroupName	Info	Decision_1	1	Group 1	Mem 1:
1	Step 1			0	0	0	0
2	Step 14			0	0	0	0
3	Decision 1		Path 1 jump to 4 Path 2 jump to 5	0	0	0	0
4	Step Path 1	Path 1		I	0	0	I
5	Step Path 2	Path 2		0	I	I	0
6	Step 6	Path 2	Timer value: 5s	0	I	I	0
7	Step 7	Path 2		0	I	I	0
8	Step 8	Path 2		0	I	I	0
9	Jump 9	Path 2	Conditional jump to 5	0	I	I	0
10	Step 10	Path 2		0	I	I	0
11	Step End			0	I	I	0
12	Step 11			0	0	0	0
13	Repeater 12		Repeater with 2 iterations to step 3	0	0	0	0
14	Step 13			0	0	0	0

Studio

Invertierte Zone-In

Um ein Signal vollständig abzusichern kann eine invertierte Zone-In eingefügt werden. Die invertierte Zone überwacht den sicheren Übergang des mit der Zone verknüpften Signals von true auf false. Beispielsweise werden Taster, Sensoren etc. damit überwacht. Dadurch kann eine Fehlbedienung verhindert werden. Mithilfe des Buttons Clone to inverted wird eine invertierte Zone der ausgewählten Zone-In eingefügt.

The screenshot shows the Selmo Studio interface. At the top, there is a 'Tools' menu with several icons. The 'Clone to Inverted' icon, which shows a blue square with a white circle and a diagonal line, is highlighted with a red box. Below the menu is a table representing the ladder logic logic.

#	Step	GroupName	Info	Decision_1	1	Group 1	Mem 1:										
				Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone 11	Zone Mem 12	Zone Mem 12	Zone 14
1	Step 1			0	0	0	0	0	0	S	0	0	0	0	0	0	0
2	Step 14			0	0	0	0	0	0	I	0	0	0	0	0	0	0
3	Decision 1		Path 1 jump to 4 Path 2 jump to 5	0	0	0	0	S	0	I	0	D2	D1	0	0	0	0
4	Step Path 1	Path 1		I	0	0	I	0	0	0	S	I	0	0	S	0	0
5	Step Path 2	Path 2		0	I	I	0	0	0	0	S	I	0	0	0	S	0
6	Step 6	Path 2	Timer value: 5s	0	I	I	0	0	S	0	I	I	0	0	0	I	0
7	Step 7	Path 2		0	I	I	0	0	0	0	I	S	0	S	0	I	0
8	Step 8	Path 2		0	I	I	0	0	0	0	I	0	S	0	0	I	0
9	Jump 9	Path 2	Conditional jump to 5	0	I	I	0	0	0	J	I	S	0	0	0	I	0
10	Step 10	Path 2		0	I	I	0	0	0	0	I	0	S	S	0	I	0
11	Step End			0	I	I	0	0	0	0	I	S	0	0	0	0	0
12	Step 11			0	0	0	0	0	0	0	I	I	0	S	0	0	0
13	Repeater 12		Repeater with 2 iterations to step 3	0	0	0	0	0	0	C	I	0	S	0	0	0	0
14	Step 13			0	0	0	0	0	0	0	I	S	0	S	0	0	0

Studio

Operanden der Zone-In

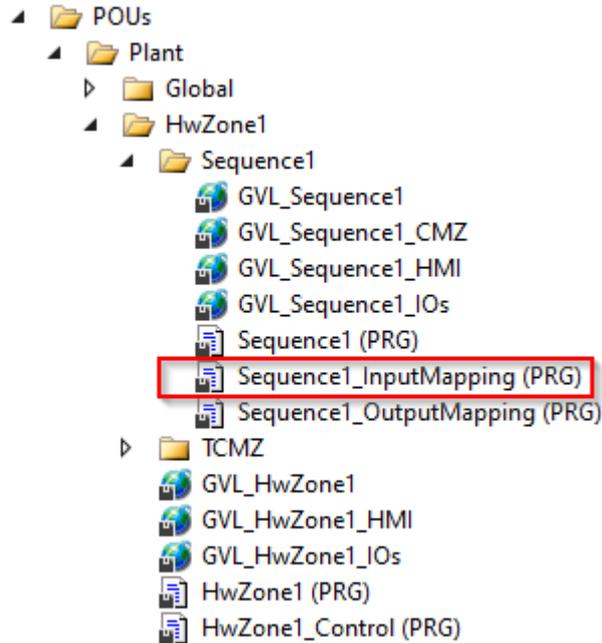
Soll eine Zone-In im Schritt überwacht werden, ist es notwendig den Operanden 'I' einzutragen. Mit den Operanden 'S' wird auf einen Zustandswechsel der Zone gewartet. Die Zone-In wird verwendet, um Eingangssignale zu verarbeiten wie z.B. Taster, Sensoren usw.

Home				Tools														
Zone In Zone In-Out Zone Out Zone Mem Add zones				Copy Paste Clone to Inverted Edit Zone Edit				Step grouping Connect to PLC Grouping Online		Zones: [dropdown] x Groups: [dropdown] Set operands filter [dropdown] [dropdown]								
#	Step	GroupName	Info	Decision_1					1	Group 1				Mem 1:				
				Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone 11	Zone Mem 12	Zone Mem 12	Zone 14	
1	Step 1			0	0	0	0	0	0	S	0	0	0	0	0	0	0	0
2	Step 14			0	0	0	0	0	0	I	0	0	0	0	0	0	0	0
3	Decision 1		Path 1 jump to 4 Path 2 jump to 5	0	0	0	0	S	0	I	0	D2	D1	0	0	0	0	
4	Step Path 1	Path 1		I	0	0	I	0	0	0	S	I	0	0	S	0	0	
5	Step Path 2	Path 2		0	I	I	0	0	0	0	S	I	0	0	0	S	0	
6	Step 6	Path 2	Timer value: 5s	0	I	I	0	0	S	0	I	I	0	0	0	I	0	
7	Step 7	Path 2		0	I	I	0	0	0	0	I	S	0	S	0	I	0	
8	Step 8	Path 2		0	I	I	0	0	0	0	I	0	S	0	0	I	0	
9	Jump 9	Path 2	Conditional jump to 5	0	I	I	0	0	0	J	I	S	0	0	0	I	0	
10	Step 10	Path 2		0	I	I	0	0	0	0	I	0	S	S	0	I	0	
11	Step End			0	I	I	0	0	0	0	I	S	0	0	0	0	0	
▶ 12	Step 11			0	0	0	0	0	0	0	I	I	0	S	0	0	0	
13	Repeater 12		Repeater with 2 iterations to step 3	0	0	0	0	0	0	C	I	0	S	0	0	0	0	
14	Step 13			0	0	0	0	0	0	0	I	S	0	S	0	0	0	

Studio

Zuordnung der Signale zu der Zone-In

Weiteres wird der Input der Zone-In im Code erzeugt und kann mit einem Signal verlinkt werden, siehe nachfolgenden Code:

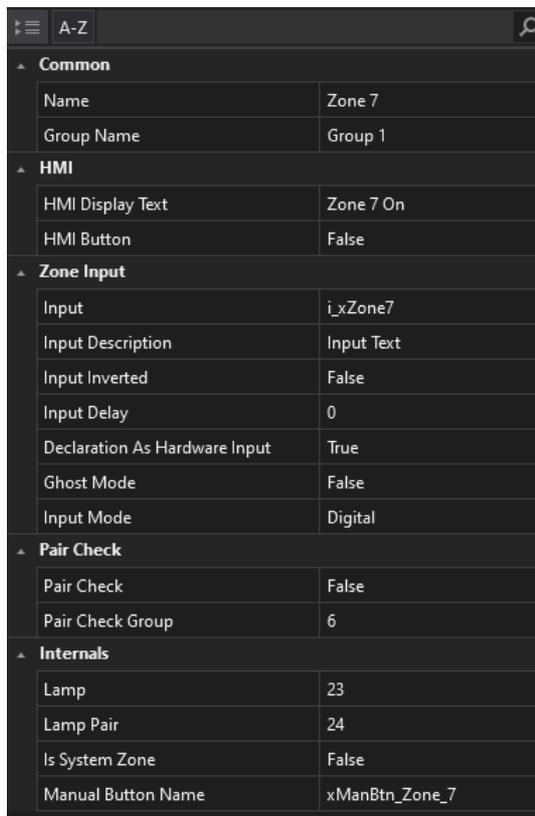


PLC

```
1 // copyright SELMO Technology GmbH by SELMStudio
2 // Version 2023.1.1.7920
3 // This function has been automatically generated.
4 {attribute 'symbol' := 'none'}
5 PROGRAM Sequence1_InputMapping
6 VAR
7 END_VAR
8
9 {region "Description Input Mapping"}
10 (*
11 All Step Sequence Zone Inputs that cannot be directly linked are connected to the real world hardware in this section.
12 This must be done manually.
13 *)
14 {endregion}
15
16 (*
17 GVL_Sequence1_IOs.i_xZone7 := ;
18 GVL_Sequence1_IOs.i_xZone9 := ;
19 GVL_Sequence1_IOs.i_xZone10 := ;
20 *)
```

PLC

Properties



The screenshot shows the 'Properties' panel in the Selmo Studio. At the top, there is a search bar with 'A-Z' and a magnifying glass icon. The panel is organized into several expandable sections, each with a small triangle icon to its left. The sections and their contents are as follows:

- Common**
 - Name: Zone 7
 - Group Name: Group 1
- HMI**
 - HMI Display Text: Zone 7 On
 - HMI Button: False
- Zone Input**
 - Input: i_xZone7
 - Input Description: Input Text
 - Input Inverted: False
 - Input Delay: 0
 - Declaration As Hardware Input: True
 - Ghost Mode: False
 - Input Mode: Digital
- Pair Check**
 - Pair Check: False
 - Pair Check Group: 6
- Internals**
 - Lamp: 23
 - Lamp Pair: 24
 - Is System Zone: False
 - Manual Button Name: xManBtn_Zone_7

Studio

		Decision_1															
		1													Mem 1:		
		Group 1															
#	Step	GroupName	Info	Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone 11	Zone Mem 12	Zone Mem 12	Zone 14
1	Step 1			0	0	0	0	0	0	S	0	0	0	0	0	0	0
2	Step 14			0	0	0	0	0	0	I	0	0	0	0	0	0	0
3	Decision 1		Path 1 jump to 4 Path 2 jump to 5	0	0	0	0	S	0	I	0	D2	D1	0	0	0	0
4	Step Path 1	Path 1		I	0	0	I	0	0	0	S	I	0	0	S	0	0
5	Step Path 2	Path 2		0	I	I	0	0	0	0	S	I	0	0	0	S	0
6	Step 6	Path 2	Timer value: 5s	0	I	I	0	0	S	0	I	I	0	0	0	I	0
7	Step 7	Path 2		0	I	I	0	0	0	0	I	S	0	S	0	I	0
8	Step 8	Path 2		0	I	I	0	0	0	0	I	0	S	0	0	I	0
9	Jump 9	Path 2	Conditional jump to 5	0	I	I	0	0	0	J	I	S	0	0	0	I	0
10	Step 10	Path 2		0	I	I	0	0	0	0	I	0	S	S	0	I	0
11	Step End			0	I	I	0	0	0	0	I	S	0	0	0	0	0
12	Step 11			0	0	0	0	0	0	0	I	I	0	S	0	0	0
13	Repeater 12		Repeater with 2 iterations to step 3	0	0	0	0	0	0	C	I	0	S	0	0	0	0
14	Step 13			0	0	0	0	0	0	0	I	S	0	S	0	0	0

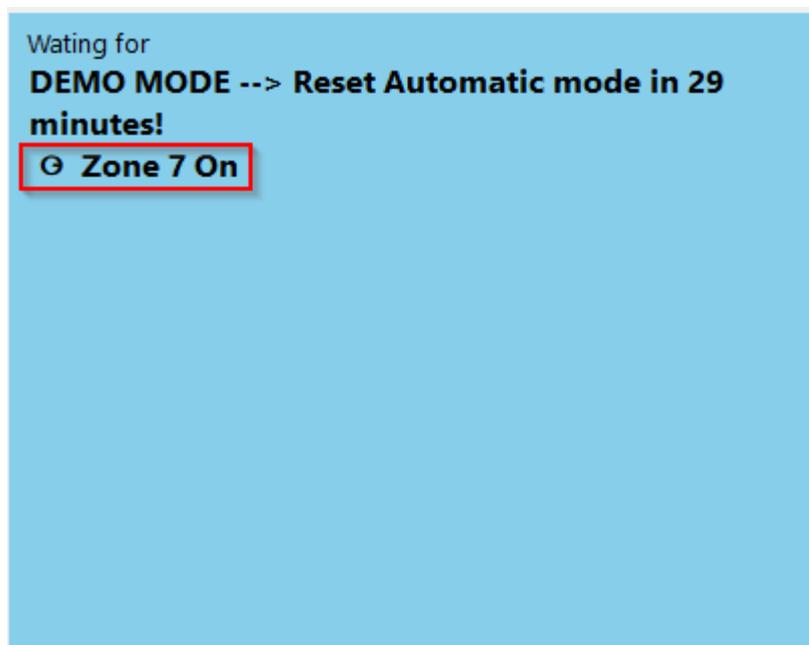
Studio

HMI Display Text

Wird hier ein Text eingetragen wirkt der Text nur in der HMI und wird als Anzeige Text übernommen. Die Eigenschaft Name wird in der HMI überschrieben.

Common	
Name	Zone 7
GroupName	Group 1
HMI	
HMI Display Text	Zone 7 On
HMI Button	False
Zone Input	

Studio



HMI

Input

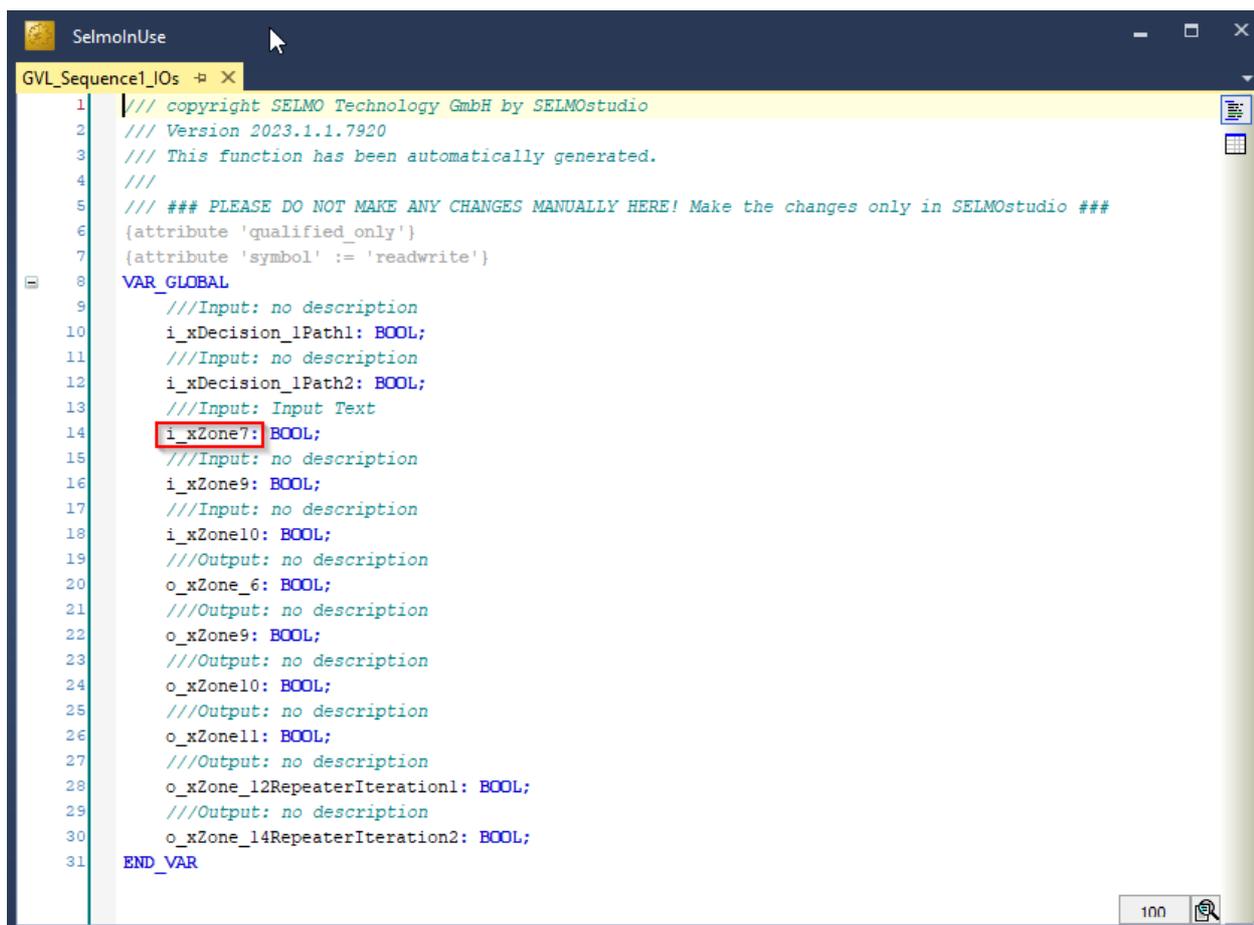
Im Kontext auf die PLC-Programmierung bezieht sich der Begriff "Name der Input-Variable" auf die Bezeichnung oder den Namen einer Eingangsvariable, die im Programmcode des programmierbaren Logikcontrollers (PLC) definiert und verwendet wird. Eine Eingangsvariable kann beispielsweise ein Signal von einem Sensor oder einer anderen Quelle sein, das von der PLC verarbeitet wird, um eine bestimmte Aktion auszuführen. Der Name der Input-Variable im PLC-Programm ist ein wichtiger Aspekt bei der Programmierung des Controllers, da er dazu beiträgt, dass der Code leichter lesbar, verständlicher und wartbarer wird. Der Name sollte daher sorgfältig gewählt und beschreibend sein, um die Funktion der Eingangsvariable zu verdeutlichen und die Lesbarkeit des Codes zu erhöhen.

Zone Input	
Input	i_xZone7
Input Description	Input Text
Input Inverted	False
Input Delay	0
Declaration As Hardware Input	False
Input Mode	Digital

Studio

- ▲ POUs
 - ▲ Plant
 - ▶ Global
 - ▲ HwZone1
 - ▲ Sequence1
 - GVL_Sequence1
 - GVL_Sequence1_CMZ
 - GVL_Sequence1_HMI
 - GVL_Sequence1_IOs**
 - Sequence1 (PRG)
 - Sequence1_InputMapping (PRG)
 - Sequence1_OutputMapping (PRG)
 - ▶ TCMZ
 - GVL_HwZone1
 - GVL_HwZone1_HMI
 - GVL_HwZone1_IOs
 - HwZone1 (PRG)
 - HwZone1_Control (PRG)

PLC



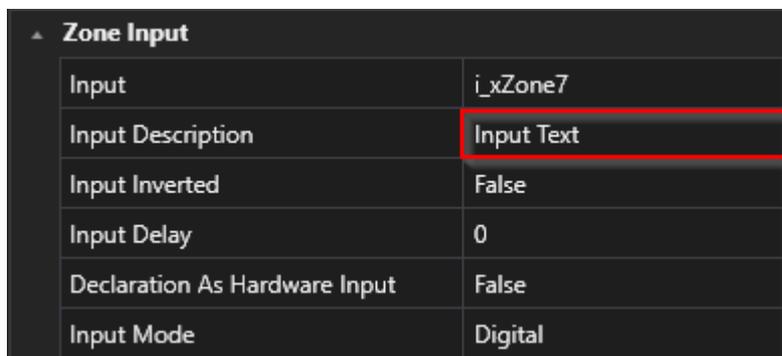
```
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  /// 
5  /// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  {attribute 'qualified_only'}
7  {attribute 'symbol' := 'readwrite'}
8  VAR_GLOBAL
9      /// //Input: no description
10     i_xDecision_lPath1: BOOL;
11     /// //Input: no description
12     i_xDecision_lPath2: BOOL;
13     /// //Input: Input Text
14     i_xZone7: BOOL;
15     /// //Input: no description
16     i_xZone9: BOOL;
17     /// //Input: no description
18     i_xZone10: BOOL;
19     /// //Output: no description
20     o_xZone_6: BOOL;
21     /// //Output: no description
22     o_xZone9: BOOL;
23     /// //Output: no description
24     o_xZone10: BOOL;
25     /// //Output: no description
26     o_xZone11: BOOL;
27     /// //Output: no description
28     o_xZone_l2RepeaterIteration1: BOOL;
29     /// //Output: no description
30     o_xZone_l4RepeaterIteration2: BOOL;
31 END_VAR
```

PLC

Input Description

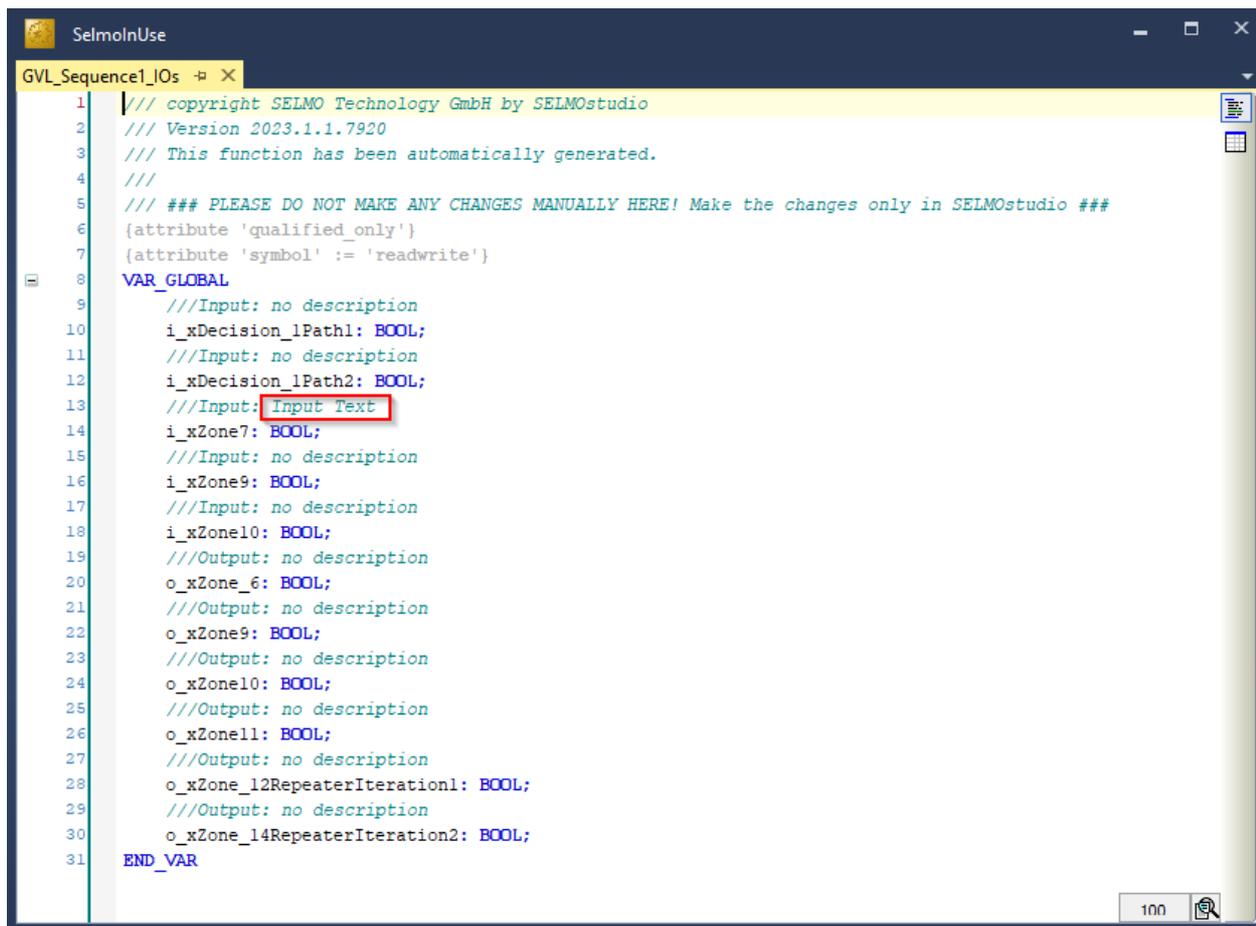
In der Programmierung von PLCs ist es wichtig, dass jede Eingangsvariable im Code eine aussagekräftige Beschreibung erhält. Eine solche Beschreibung hilft anderen Programmierern oder Wartungspersonal, den Code leichter zu verstehen und zu bearbeiten. Die Beschreibung der Eingangsvariable sollte idealerweise den Zweck und die Funktionsweise der Variable beschreiben.

Durch eine klare und präzise Beschreibung der Eingangsvariable kann auch sichergestellt werden, dass sie richtig konfiguriert und kalibriert ist, bevor sie in das PLC-Programm integriert wird. Außerdem hilft sie bei der Fehlersuche und Fehlerbehebung im Code, indem sie es dem Programmierer ermöglicht, schnell festzustellen, welche Eingangsvariablen betroffen sind. Die Beschreibung der Eingangsvariable sollte daher als wichtiger Bestandteil der Dokumentation des PLC-Programms angesehen werden, um die Effizienz, Wartbarkeit und Fehlerfreiheit des Codes zu erhöhen.



Zone Input	
Input	i_xZone7
Input Description	Input Text
Input Inverted	False
Input Delay	0
Declaration As Hardware Input	False
Input Mode	Digital

Studio



```
1  ///  
2  ///  
3  ///  
4  ///  
5  ///  
6  ///  
7  ///  
8  VAR_GLOBAL  
9  ///  
10 i_xDecision_lPath1: BOOL;  
11 ///  
12 i_xDecision_lPath2: BOOL;  
13 ///  
14 i_xZone7: BOOL;  
15 ///  
16 i_xZone9: BOOL;  
17 ///  
18 i_xZone10: BOOL;  
19 ///  
20 o_xZone_6: BOOL;  
21 ///  
22 o_xZone9: BOOL;  
23 ///  
24 o_xZone10: BOOL;  
25 ///  
26 o_xZone11: BOOL;  
27 ///  
28 o_xZone_l2RepeaterIteration1: BOOL;  
29 ///  
30 o_xZone_l4RepeaterIteration2: BOOL;  
31 END_VAR
```

PLC

Input Inverted

Bitte legen Sie fest, ob das Eingangssignal invertiert ist oder nicht. Das bedeutet, dass Sie entscheiden müssen, ob das Signal, das in ein bestimmtes System eingeht, in seiner Polarität umgekehrt ist oder nicht. Eine invertierte Signalpolarität bedeutet, dass das Signal in Bezug auf seine positive und negative Polarität umgekehrt ist. Es ist wichtig, diese Entscheidung zu treffen, da sie Auswirkungen auf die Art und Weise hat, wie das Signal im System verarbeitet wird.

Input Delay

Bitte geben Sie an, um wie viele Millisekunden das Eingangssignal verzögert werden soll. Die Verzögerung bezieht sich auf den Zeitunterschied zwischen dem Zeitpunkt, zu dem das Signal empfangen wird, und dem Zeitpunkt, zu dem es im System verarbeitet wird. Eine Verzögerung kann beabsichtigt sein, um das Signal auf eine bestimmte Weise zu modifizieren oder um sicherzustellen, dass es synchron mit anderen Signalen verarbeitet wird. Die genaue Zeitdauer, um die das Signal verzögert werden soll, hängt von den Anforderungen des Systems und der Art des Signals ab, das verarbeitet wird.

Declaration as Hardware Input

Wenn Sie den Input als "true" deklarieren, wird sie als Hardware-Eingang deklariert und mit dem "AT %I*" Attribut in der Programmierungslogik eingebunden. Dies bedeutet, dass die Variable ein Signal oder einen Wert von einem physikalischen Eingang des Systems empfängt, wie beispielsweise von einem Sensor oder einem Schalter.

Zone Input	
Input	i_xZone7
Input Description	Input Text
Input Inverted	False
Input Delay	0
Declaration As Hardware Input	True
Ghost Mode	False
Input Mode	Digital

Studio

```
SelmoInUse
GVL_Sequence1_I0s -> X
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  ///
5  /// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  {attribute 'qualified_only'}
7  {attribute 'symbol' := 'readwrite'}
8  VAR_GLOBAL
9  ///Input: no description
10 i_xDecision_1Path1: BOOL;
11 ///Input: no description
12 i_xDecision_1Path2: BOOL;
13 ///Input: Input Text
14 i_xZone7 AT %I*: BOOL;
15 ///Input: no description
16 i_xZone9: BOOL;
17 ///Input: no description
18 i_xZone10: BOOL;
19 ///Output: no description
20 o_xZone_6: BOOL;
21 ///Output: no description
22 o_xZone9: BOOL;
23 ///Output: no description
24 o_xZone10: BOOL;
25 ///Output: no description
26 o_xZone11: BOOL;
27 ///Output: no description
28 o_xZone_12RepeaterIteration1: BOOL;
29 ///Output: no description
30 o_xZone_14RepeaterIteration2: BOOL;
31 END_VAR
```

PLC

Input Mode

Die Art des Signals wird durch den Modus des Eingangs bestimmt. Dabei kann es sich entweder um ein digitales oder analoges Signal oder um einen Parameter handeln. Der Modus des Eingangs gibt somit an, welche Art von Signal erwartet wird und wie dieses Signal interpretiert werden soll. Wenn der Eingangsmodus beispielsweise auf "digital" eingestellt ist, erwartet das System ein Signal, das aus diskreten Werten besteht, während bei einem analogen Eingangsmodus ein kontinuierliches Signal erwartet wird. Bei einem Parametermodus hingegen wird ein Wert erwartet, der einen bestimmten Parameter repräsentiert. Insgesamt hängt die Art des Signals, das ein System empfängt und verarbeitet, somit maßgeblich vom eingestellten Eingangsmodus ab.

AnalogValue

Zone Input	
Input	i_xZone7
Input Description	Input Text
Input Inverted	False
Input Delay	0
Declaration As Hardware Input	True
Ghost Mode	False
Input Mode	AnalogValue
Input Analog Setpoint	0
Input Analog Function	Equals

Im Falle der Auswahl von 'AnalogValue' wartet die Zone intern auf ein digitales Signal, während extern ein analoger Wert ausgewählt wird. Die Weiterleitung erfolgt, sobald diese Funktion aktiviert ist und eine spezifische Bedingung erfüllt ist. Diese Funktion ist im PLC-Code unter der entsprechenden Zone zu finden.

AnalogParameter

Zone Input	
Input	i_xZone7
Input Description	Input Text
Input Inverted	False
Input Delay	0
Declaration As Hardware Input	True
Ghost Mode	False
Input Mode	AnalogParameter
Input Analog Setpoint Parameter	
Input Analog Function	Equals

Das Prinzip entspricht dem von 'AnalogValue', jedoch unterscheidet sich der analoge Schritt dadurch, dass er nicht mehr ein statischer Wert ist, sondern ein Parameter. Unter 'Input Analog Setpoint Parameter' wird dann der entsprechende Name des ausgewählten Parameters angezeigt. Diese Funktion ist ebenfalls im PLC-Code unter der entsprechenden Zone zu finden.

ParameterList

Zone Input	
Input	i_xZone7
Input Description	Input Text
Input Inverted	False
Input Delay	0
Declaration As Hardware Input	True
Ghost Mode	False
Input Mode	ParameterList
Input Parameter List	
Input Analog Function	Equals

Die 'ParameterList' wird mit einem Eingangssignal verglichen, wobei wiederum ein analoger Wert eingelesen wird. Der Unterschied besteht jedoch darin, dass pro Schritt mehrere Vergleiche durchgeführt werden können. Um diese Funktion nutzen zu können, muss bereits eine Parameterliste vorhanden sein oder erstellt werden um diese in der gewünschten Zone auswählen zu können. Die ausgewählte Parameter Liste steht dann in 'Input Parameter List'. Dort wo man in der Parameter Liste einen Vergleich machen möchte muss im System Layer ein 'Sequence Check' gesetzt werden.

ParameterInput

Zone Input	
Input Description	Input Text
Input Inverted	False
Input Delay	0
Ghost Mode	False
Input Mode	ParameterInput
Input Parameter	
Input Analog Setpoint Parameter	
Input Analog Function	Equals

Bei dieser Funktion entfällt das Eingangssignal, stattdessen werden zwei Parameter miteinander verglichen. Im PLC-Code wird unter der entsprechenden Zone nicht mehr die IO-Stelle der Zone verwendet, sondern auch hier wird ein Parameter angegeben.

PairCheck

Wenn 'PairCheck' aktiv ist, wird die Zone mit anderen Zonen überprüft, um sicherzustellen, dass bestimmte Bedingungen erfüllt sind, wie beispielsweise das Vorhandensein von Signal 1 und Signal 2, die nicht zeitgleich auftreten dürfen.

PairCheckGroup

Die PairCheckGroup-Nummer wird verwendet, um festzulegen, ob eine Zone-In einer Prüfung mit anderen Zonen derselben Gruppe zusammengefasst werden soll. Diese Gruppierung ermöglicht es, bestimmte Prüfungen auf mehrere Zonen anzuwenden, die dieselbe PairCheckGroup-Nummer aufweisen, um sicherzustellen, dass die Ergebnisse konsistent sind.

Internals

Internals	
Lamp	23
Lamp Pair	24
Is System Zone	False
Manual Button Name	xManBtn_Zone_7

Lamp

Internals	
Lamp	23
Lamp Pair	24
Is System Zone	False
Manual Button Name	xManBtn_Zone_7

'Lamp' bezieht sich auf den Index des jeweiligen Arrays. Die Zoneninformation wird über diesen Index im Array gespeichert. Mit diesem Index wird dann über die HMI auf das Array zugegriffen, und die Texte werden im Fenster 'Waiting for' ausgegeben.

Lamp Pair

Internals	
Lamp	23
Lamp Pair	24
Is System Zone	False
Manual Button Name	xManBtn_Zone_7

Gleiche Funktion wie 'Lamp'.

Is System Zone

Internals	
Lamp	23
Lamp Pair	24
Is System Zone	False
Manual Button Name	xManBtn_Zone_7

Wenn die Funktion auf True gesetzt ist, dient dies als Indikator dafür, dass es sich um eine Systemzone handelt, die automatisch erstellt wird (z. B. Timer)..

Manual Button Name

Internals	
Lamp	23
Lamp Pair	24
Is System Zone	False
Manual Button Name	xManBtn_Zone_7

Hierbei handelt es sich um eine textbasierte Information, wie der Manual Button in der PLC benannt wurde.

1.3.1.2 Zone InOut

Die **Zone-InOut** kombiniert eine Eingangs- und eine Ausgangszone. Um den Ausgang zu steuern, ist es notwendig, den Operanden 'S' einzugeben, damit der Ausgang der Zone solange gesteuert wird, bis der Eingang der Zone den logischen Wert 1 meldet. Damit ist die Bedingung erfüllt und es kann zum nächsten Schritt übergegangen werden. Soll der Eingang der Zone überwacht werden, ist dies durch Eingabe des Operanden 'I' möglich. Zum Beispiel wird mit dem Ausgang der Zone ein Ventil eines Pneumatikzylinders angesteuert und mit dem entsprechenden Eingang der Zone die Endlage des Zylinders überwacht.

Zone-InOut erstellen

Um einen Zone-InOut zu erstellen, klicken Sie auf das entsprechende Symbol.

The screenshot shows the Selmo Studio interface for 'Sequence1.SystemLayer'. The toolbar includes 'Add zones' (Zone In, Zone In-Out, Zone Out, Zone Mem), 'Edit' (Copy, Paste, Clone to Inverted, Edit Zone, Remove Zone, Remove Step), 'Grouping' (Step grouping), 'Online' (Connect to PLC), and 'Filter' (Zones, Groups). The main workspace displays a ladder logic diagram for 'Decision_1' with steps 1 through 13. A red box highlights the 'Zone 9 Inv' element in the diagram.

#	Step	GroupName	Info	Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone 11
1	Step 1			0	0	0	0	0	0	S	0	0	0	0
2	Decision 1	Path 1 jump to 3 Path 2 jump to 4		0	0	0	0	S	0	I	0	D2	D1	0
3	Step Path 1			I	0	0	I	0	0	0	S	I	0	0
4	Step Path 2			0	I	I	0	0	0	0	S	I	0	0
5	Step 6	Timer value: 5s		0	I	I	0	0	S	0	I	I	0	0
6	Step 7			0	I	I	0	0	0	0	I	S	0	S
7	Step 8			0	I	I	0	0	0	0	I	0	S	0
8	Jump 9	Conditional jump to 4		0	I	I	0	0	0	J	I	S	0	0
9	Step 10			0	I	I	0	0	0	0	I	0	S	S
10	Step End			0	I	I	0	0	0	0	I	S	0	0
11	Step 11			0	0	0	0	0	0	0	I	I	0	S
12	Repeater 12	Repeater with 2 iterations to step 2		0	0	0	0	0	0	C	I	0	S	0
13	Step 13			0	0	0	0	0	0	0	I	S	0	S

Studio

Operanden des Zone-InOut

Das Zone-InOut ist definiert als ein Ausgang mit entsprechender Rückmeldung über einen Eingang. So können Ausgänge gesteuert werden, bis eine entsprechende Rückmeldung über den Eingang erfolgt. Damit kann z.B. ein Ventil mit Positionserfassung angesteuert werden. Die Positionserfassung des Ventils ist so ausgelegt, dass die Stellung offen oder geschlossen erkannt werden kann. Um die Aktion (Ventil öffnen) zu starten, wird eine Ablaufkontrolle gesetzt. Nun wird der Ausgang des mit dem Zone-InOut verknüpften Ventils so lange angesteuert, bis die Zielposition des Ventils erreicht ist und die Rückmeldung über den verknüpften Eingang der Positionserkennung erfolgt. Mit der Verriegelungskontrolle kann überprüft werden, ob die mit dem Zone-InOut verknüpften Eingänge in diesem Zustand bleiben. Wird über den Wert 0 des Operanden in einem Zone-InOut ein "don't care" gesetzt, wird die Information des verknüpften Eingangs ignoriert und der verknüpfte Ausgang nicht verwendet.

Wenn ein Zone-InOut im Schritt überwacht werden soll, ist es notwendig, den Operanden 'I' einzugeben. Mit dem Operanden 'S' wird ein Zustandswechsel von logisch 0 nach logisch 1 abgewartet. Der Zone-InOut dient der Verarbeitung von Eingangs- und Ausgangssignalen wie Zylinder, Frequenzumrichter, Schütze usw.

Sequence1.SystemLayer

Home Tools

Zone In Zone In-Out Zone Out Zone Mem

Copy Paste Clone to Inverted Remove Zone Remove Step Edit Zone

Step grouping Connect to PLC

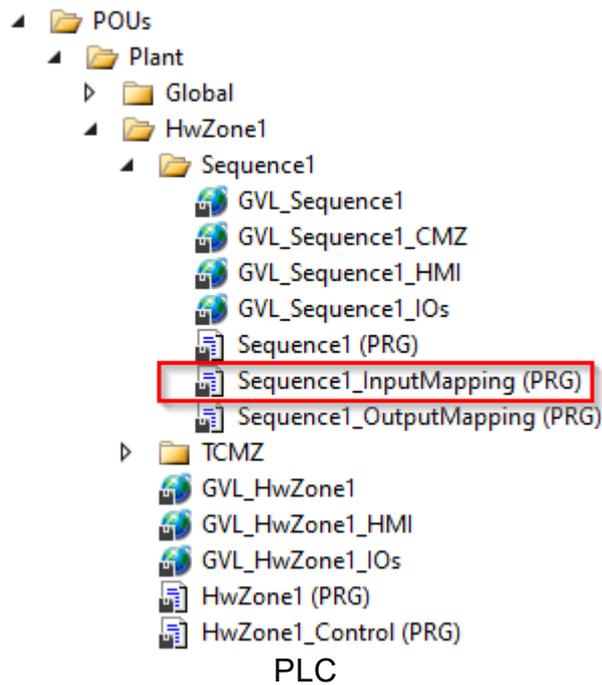
Zones Groups Filter

#	Step	GroupName	Info	Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone 11
1	Step 1			0	0	0	0	0	0	S	0	0	0	0
2	Decision 1		Path 1 jump to 3 Path 2 jump to 4	0	0	0	0	S	0	I	0	D2	D1	0
3	Step Path 1			I	0	0	I	0	0	0	S	I	0	0
4	Step Path 2			0	I	I	0	0	0	0	S	I	0	0
5	Step 6		Timer value: 5s	0	I	I	0	0	S	0	I	I	0	0
6	Step 7			0	I	I	0	0	0	0	I	S	0	S
7	Step 8			0	I	I	0	0	0	0	I	0	S	0
8	Jump 9		Conditional jump to 4	0	I	I	0	0	0	J	I	S	0	0
9	Step 10			0	I	I	0	0	0	0	I	0	S	S
10	Step End			0	I	I	0	0	0	0	I	S	0	0
11	Step 11			0	0	0	0	0	0	0	I	I	0	S
12	Repeater 12		Repeater with 2 iterations to step 2	0	0	0	0	0	0	C	I	0	S	0
13	Step 13			0	0	0	0	0	0	0	I	S	0	S

Studio

Zuweisung von Signalen an den Zone-InOut

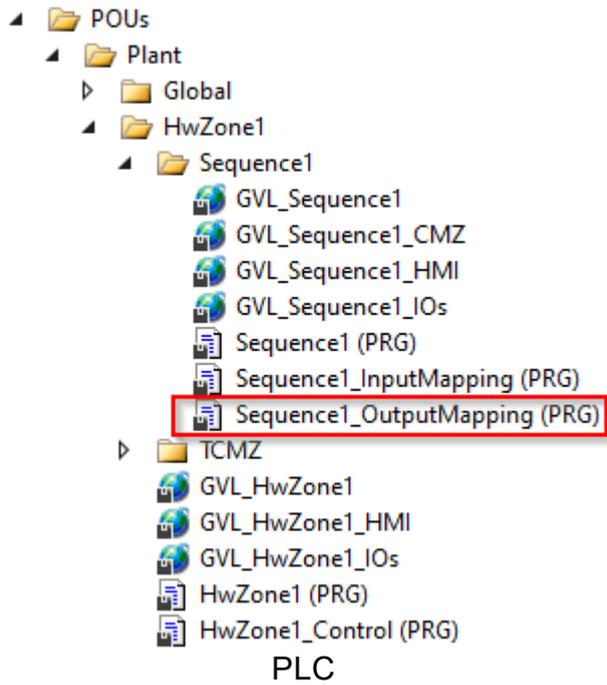
Weiterhin wird der Eingang des Zone-InOut im Code eingetragen und kann mit einem Signal verknüpft werden, siehe folgender Code:



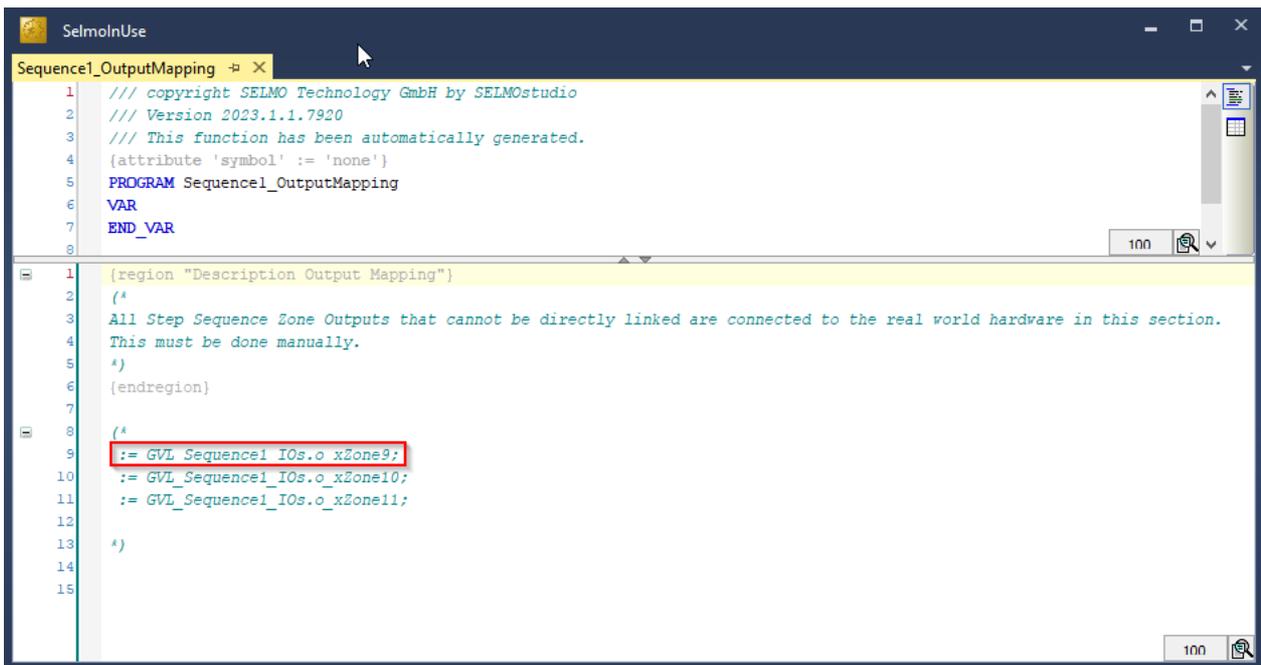
```
Sequence1_InputMapping
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  {attribute 'symbol' := 'none'}
5  PROGRAM Sequence1_InputMapping
6  VAR
7  END_VAR
8
9  {region "Description Input Mapping"}
10 (*
11 All Step Sequence Zone Inputs that cannot be directly linked are connected to the real world hardware in this section.
12 This must be done manually.
13 *)
14 {endregion}
15
16 (*
17 GVL_Sequence1_IOs.i_xZone7 := ;
18 GVL_Sequence1_IOs.i_xZone9 := ;
19 GVL_Sequence1_IOs.i_xZone10 := ;
20 *)
```

PLC

Außerdem wird der Ausgang des Zone-InOut im Code eingetragen und kann mit einem Signal verknüpft werden, siehe folgender Code:

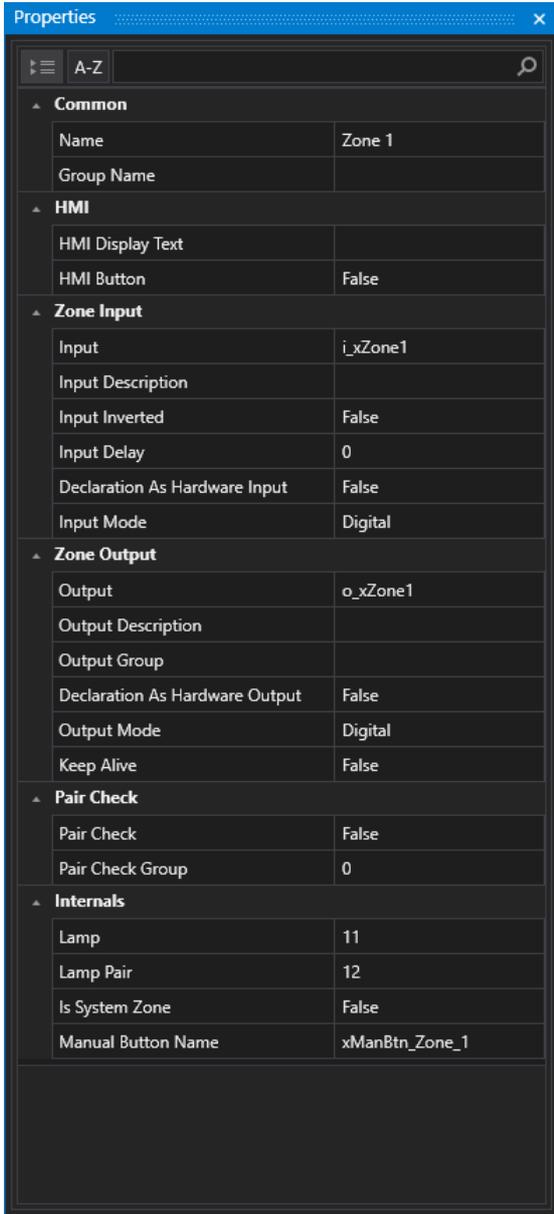


PLC



PLC

Properties



Studio

- Name, Gruppenname, HMI-Anzeigetext, HMI-Schaltfläche, HMI-Schaltflächentext, Eingang, Eingangsbeschreibung, Eingang invertiert, Eingangsverzögerung, Deklaration als Hardware-Eingang, Ghost Mode, Eingangsmodus, Keep Alive, Ausgang, Ausgangsbeschreibung, Ausgangsgruppe, Deklaration als Hardware-Ausgang, Ausgangsverteilung, Ausgangsverteilung gespeichert, PairCheck, PairCheckGroup, Lampe, Lampenpaar, Is System Zone, Manual Button Name

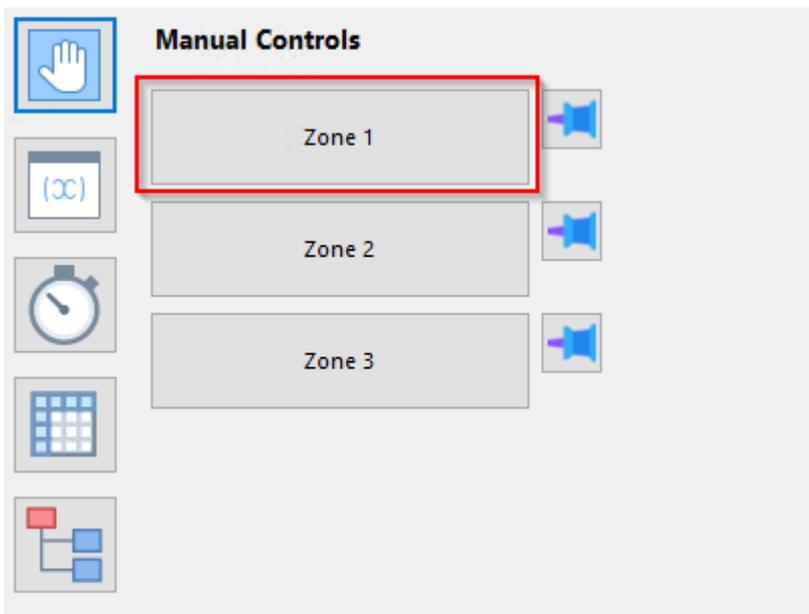
Siehe Zone-In

HMI Button

Die HMI-Taste ist eine Benutzerschnittstelle, die nur im manuellen Modus funktioniert und zur Steuerung der Ausgabe verwendet wird. Im Gegensatz zum Automatikmodus, in dem das System automatisch arbeitet, hat der Benutzer im manuellen Modus die Möglichkeit, manuelle Eingriffe vorzunehmen.

Die HMI-Taste wird gedrückt, um den Ausgang zu steuern, bis ein Rückmeldesignal (Eingang der Zone) aktiv wird. Das Rückmeldesignal gibt dem System Rückmeldung darüber, ob der gewünschte Zustand erreicht wurde oder ob Änderungen vorgenommen werden müssen. Wenn die MXIC-Querverriegelung aktiv ist, kann der Ausgang nicht gesteuert werden, und die HMI zeigt Informationen über die Zone an, mit der die Verriegelung stattfindet.

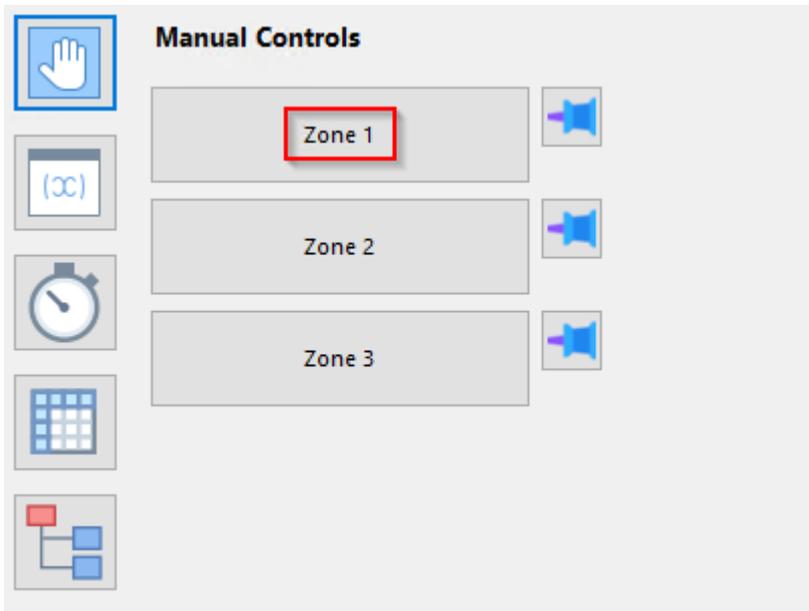
Die MXIC-Querverriegelung ist eine Sicherheitsfunktion, die verhindert, dass die Zone aktiviert wird, wenn bestimmte Bedingungen nicht erfüllt sind (Zonen x y befinden sich in einem bestimmten Zustand). Dies ist besonders wichtig bei kritischen Anwendungen, bei denen das gleichzeitige Auftreten mehrerer Ereignisse zu gefährlichen Situationen führen kann. Die Verwendung der HMI-Taste in Kombination mit der MXIC-Quersperre stellt sicher, dass nur der gewünschte Ausgang aktiviert wird und potenzielle Gefahren vermieden werden.



HMI

HMI Button Text

Der HMI-Tastentext ist der Bezeichner der Handbetriebstaste der Zone und bietet eine eindeutige Funktionsbezeichnung der Zone, um eine intuitive Bedienung für den Benutzer zu ermöglichen.



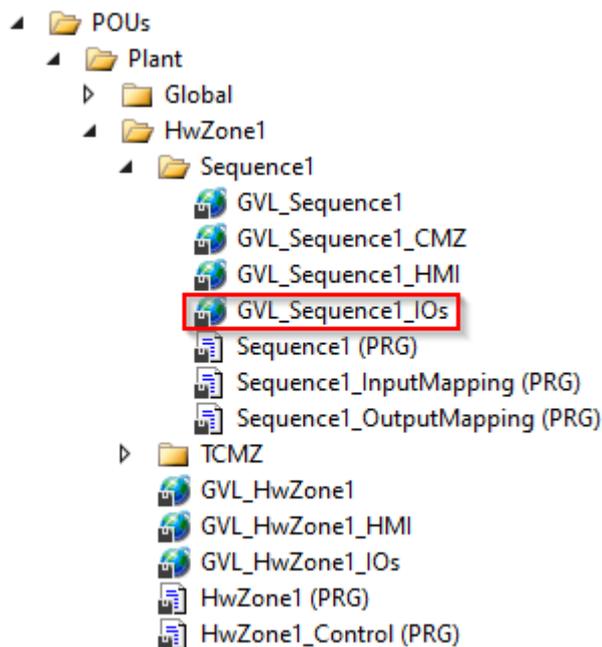
HMI

Output

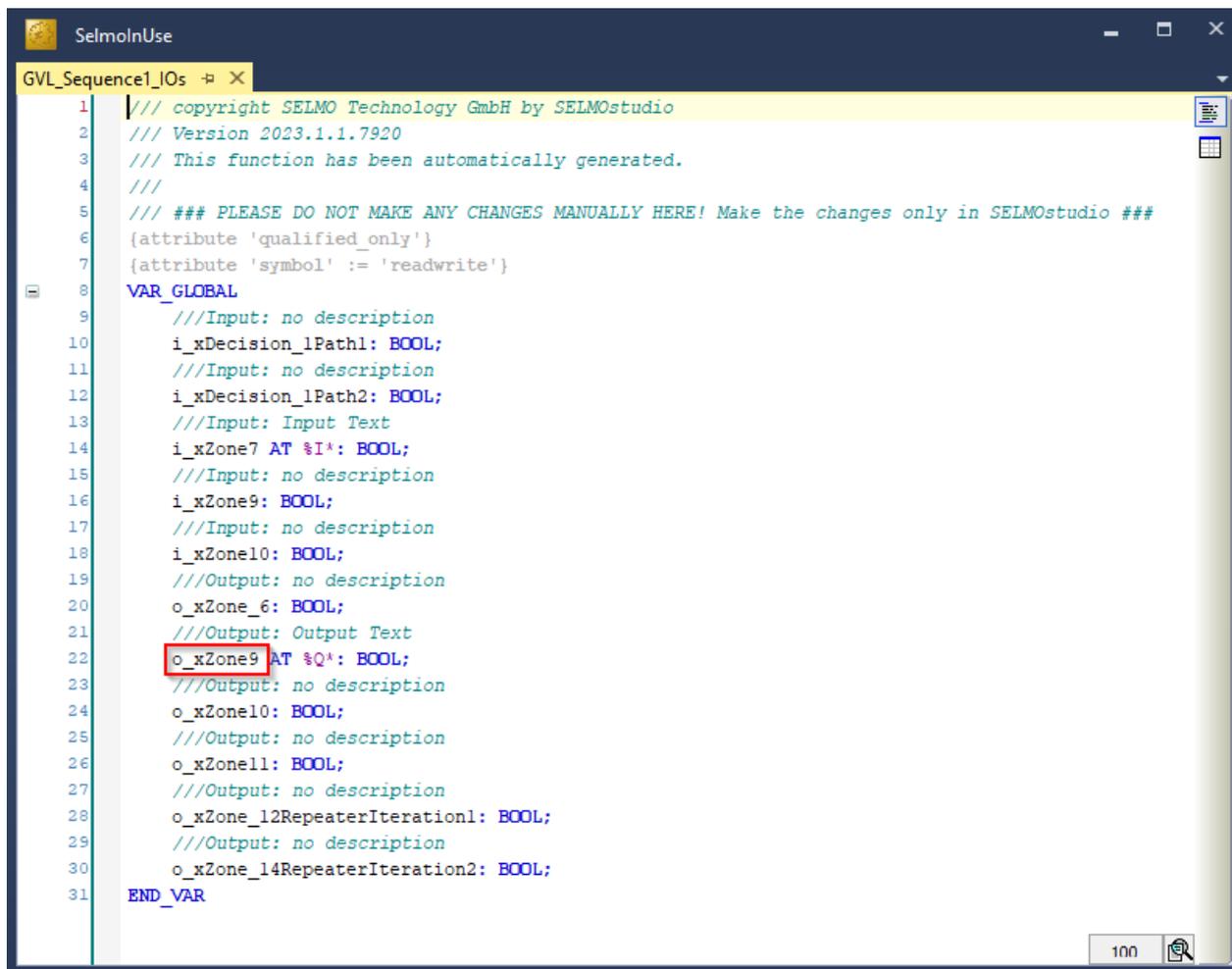
Im Zusammenhang mit der SPS-Programmierung bezieht sich der Begriff "Ausgangsvariablenname" auf die Bezeichnung oder den Namen einer Ausgangsvariablen, die im Programmcode der speicherprogrammierbaren Steuerung (SPS) definiert und verwendet wird. Eine Ausgangsvariable kann zum Beispiel ein Signal von einem Stellglied oder einer anderen Senke sein, das die SPS verarbeitet, um eine bestimmte Aktion auszuführen. Der Name der Ausgangsvariablen im SPS-Programm ist ein wichtiger Aspekt der Steuerungsprogrammierung, da er dazu beiträgt, dass der Code leichter zu lesen, zu verstehen und zu pflegen ist. Daher sollte der Name sorgfältig gewählt und aussagekräftig sein, um die Funktion der Ausgangsvariablen zu verdeutlichen und die Lesbarkeit des Codes zu erhöhen.

Zone Output	
Output	o_xZone9
Output Description	Output Text
Output Group	Conveyor
Declaration As Hardware Output	True
Output Mode	Digital
Keep Alive	False

Studio



PLC



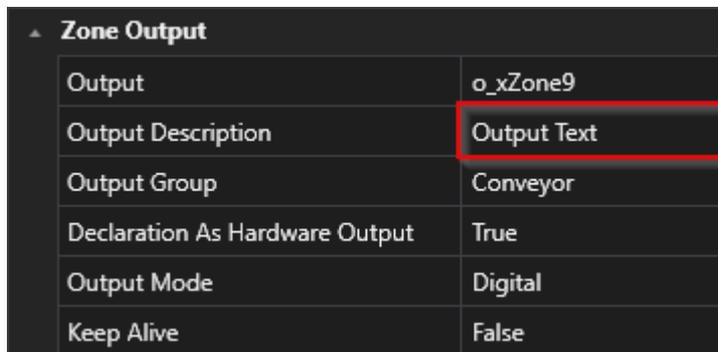
```
1  |/// copyright SELMO Technology GmbH by SELMOstudio
2  |/// Version 2023.1.1.7920
3  |/// This function has been automatically generated.
4  |///
5  |/// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  |{attribute 'qualified_only'}
7  |{attribute 'symbol' := 'readwrite'}
8  |VAR_GLOBAL
9  |    ///Input: no description
10 |    i_xDecision_1Path1: BOOL;
11 |    ///Input: no description
12 |    i_xDecision_1Path2: BOOL;
13 |    ///Input: Input Text
14 |    i_xZone7 AT %I^: BOOL;
15 |    ///Input: no description
16 |    i_xZone9: BOOL;
17 |    ///Input: no description
18 |    i_xZone10: BOOL;
19 |    ///Output: no description
20 |    o_xZone_6: BOOL;
21 |    ///Output: Output Text
22 |    o_xZone9 AT %Q^: BOOL;
23 |    ///Output: no description
24 |    o_xZone10: BOOL;
25 |    ///Output: no description
26 |    o_xZone11: BOOL;
27 |    ///Output: no description
28 |    o_xZone_12RepeaterIteration1: BOOL;
29 |    ///Output: no description
30 |    o_xZone_14RepeaterIteration2: BOOL;
31 |END_VAR
```

PLC

Output Description

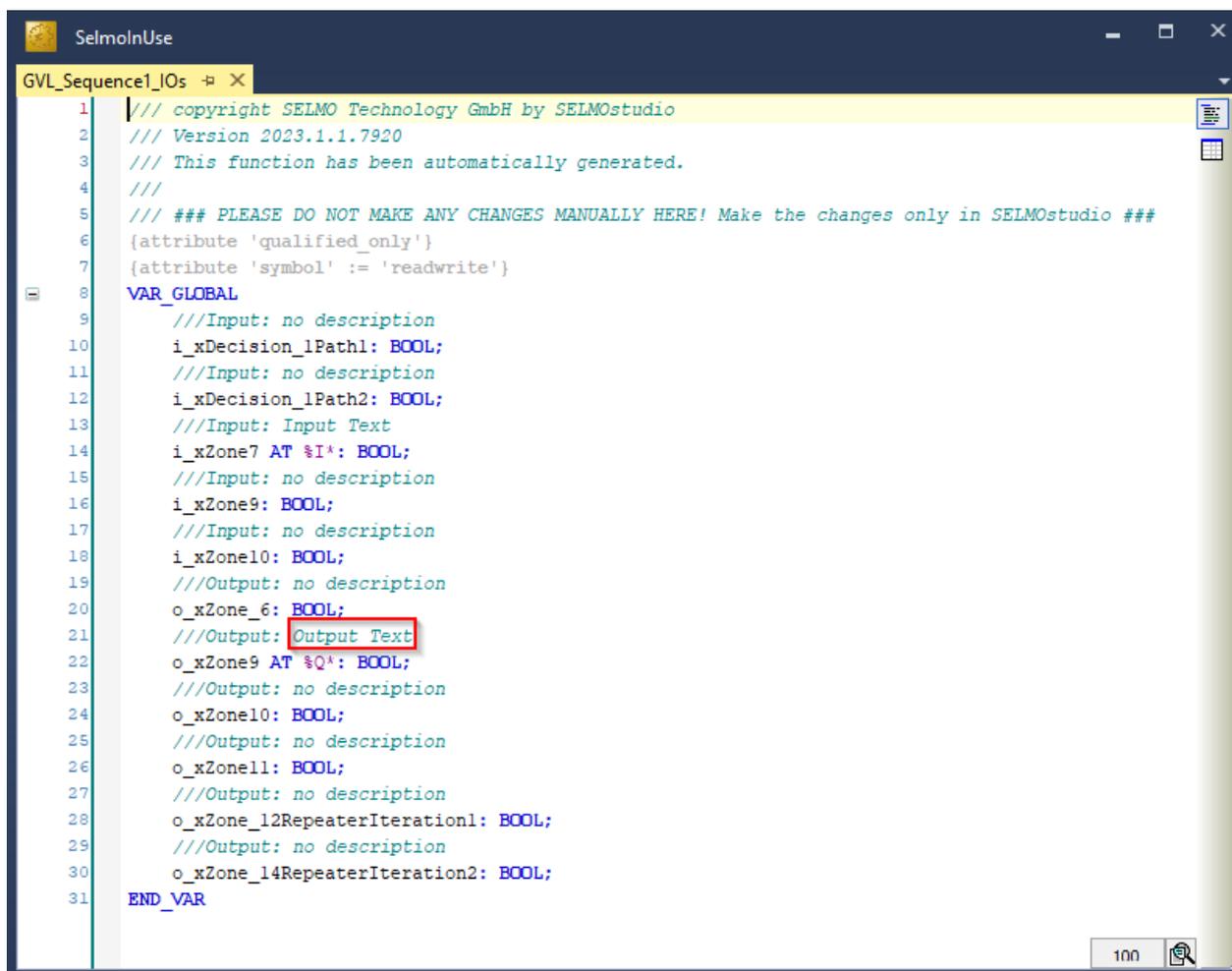
Bei der SPS-Programmierung ist es wichtig, dass jede Ausgangsvariable im Code eine aussagekräftige Beschreibung hat. Eine solche Beschreibung hilft anderen Programmierern oder dem Wartungspersonal, den Code leichter zu verstehen und zu bearbeiten. Die Beschreibung der Ausgangsvariablen sollte idealerweise den Zweck und die Funktionsweise der Variablen beschreiben.

Eine klare und prägnante Beschreibung der Ausgangsvariablen kann auch sicherstellen, dass sie richtig konfiguriert und kalibriert wird, bevor sie in das SPS-Programm integriert wird. Sie hilft auch bei der Fehlersuche und -behebung im Code, da der Programmierer so schnell feststellen kann, welche Ausgangsvariablen betroffen sind. Daher sollte die Beschreibung der Ausgangsvariablen als wichtiger Teil der SPS-Programmdokumentation angesehen werden, um die Effizienz des Codes, die Wartbarkeit und die Fehlerfreiheit zu erhöhen.



Zone Output	
Output	o_xZone9
Output Description	Output Text
Output Group	Conveyor
Declaration As Hardware Output	True
Output Mode	Digital
Keep Alive	False

Studio



```
1  |/// copyright SELMO Technology GmbH by SELMOstudio
2  |/// Version 2023.1.1.7920
3  |/// This function has been automatically generated.
4  |///
5  |/// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  |{attribute 'qualified_only'}
7  |{attribute 'symbol' := 'readwrite'}
8  |VAR_GLOBAL
9  |    ///Input: no description
10 |    i_xDecision_1Path1: BOOL;
11 |    ///Input: no description
12 |    i_xDecision_1Path2: BOOL;
13 |    ///Input: Input Text
14 |    i_xZone7 AT %I^: BOOL;
15 |    ///Input: no description
16 |    i_xZone9: BOOL;
17 |    ///Input: no description
18 |    i_xZone10: BOOL;
19 |    ///Output: no description
20 |    o_xZone_6: BOOL;
21 |    ///Output: Output Text
22 |    o_xZone9 AT %Q^: BOOL;
23 |    ///Output: no description
24 |    o_xZone10: BOOL;
25 |    ///Output: no description
26 |    o_xZone11: BOOL;
27 |    ///Output: no description
28 |    o_xZone_12RepeaterIteration1: BOOL;
29 |    ///Output: no description
30 |    o_xZone_14RepeaterIteration2: BOOL;
31 |END_VAR
```

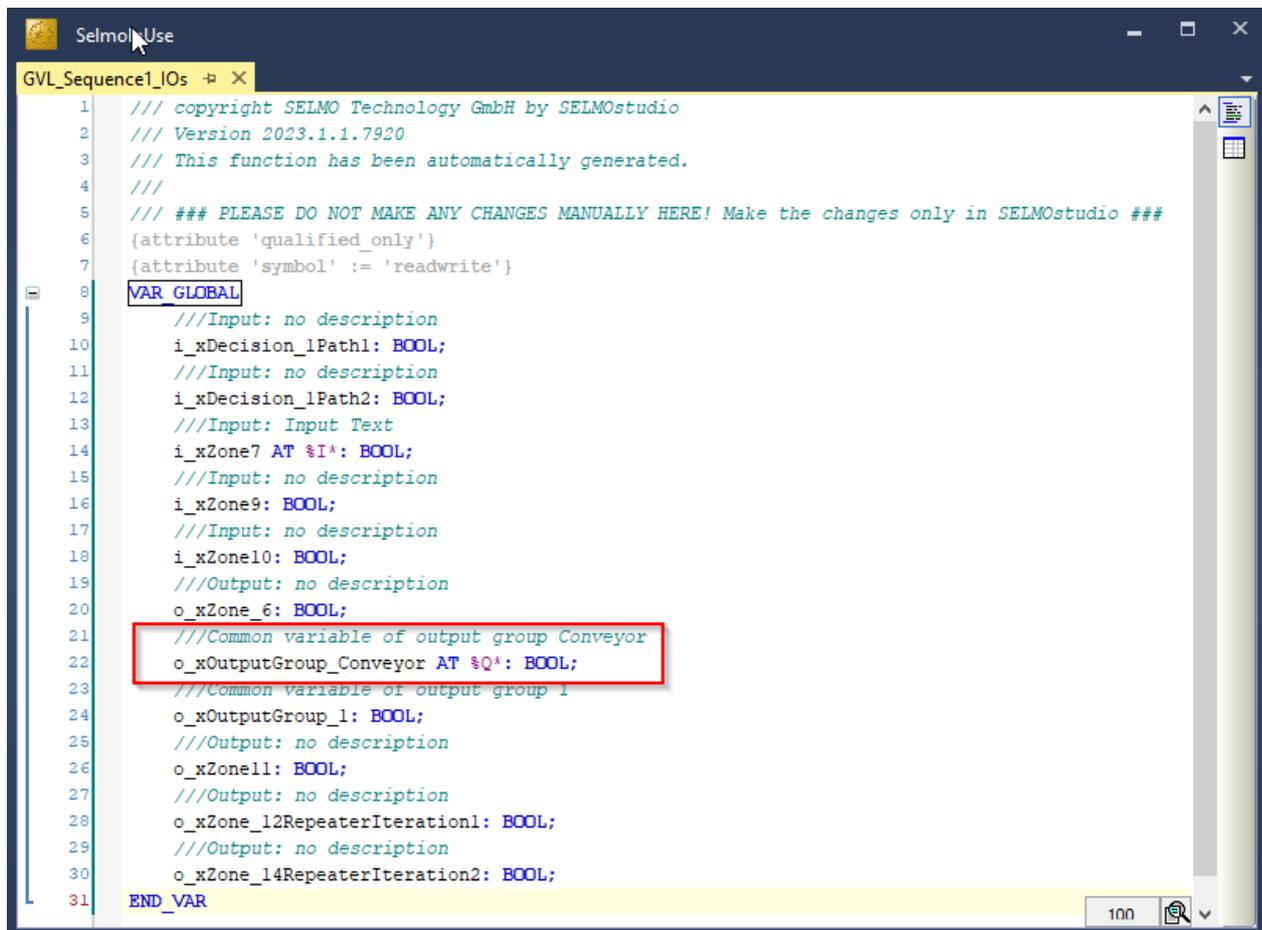
PLC

Output Group

Die Ausgangsgruppenfunktion ermöglicht den Anschluss mehrerer Zonen desselben Typs. Dabei wird nur ein gemeinsamer Ausgang für alle Zonen innerhalb der Gruppe verwendet. Das heißt, wenn Sie mehrere Zonen in einer Gruppe zusammenfassen, steuern alle diese Zonen einen Ausgang.

Zone Output	
Output	o_xZone9
Output Description	Output Text
Output Group	Conveyor
Declaration As Hardware Output	True
Output Mode	Digital
Keep Alive	False

Studio



```
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  ///
5  /// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  {attribute 'qualified only'}
7  {attribute 'symbol' := 'readwrite'}
8  VAR GLOBAL
9      ///Input: no description
10     i_xDecision_lPath1: BOOL;
11     ///Input: no description
12     i_xDecision_lPath2: BOOL;
13     ///Input: Input Text
14     i_xZone7 AT %I*: BOOL;
15     ///Input: no description
16     i_xZone9: BOOL;
17     ///Input: no description
18     i_xZone10: BOOL;
19     ///Output: no description
20     o_xZone_6: BOOL;
21     ///Common variable of output group Conveyor
22     o_xOutputGroup_Conveyor AT %Q*: BOOL;
23     ///Common variable of output group 1
24     o_xOutputGroup_1: BOOL;
25     ///Output: no description
26     o_xZonell: BOOL;
27     ///Output: no description
28     o_xZone_l2RepeaterIteration1: BOOL;
29     ///Output: no description
30     o_xZone_l4RepeaterIteration2: BOOL;
31 END_VAR
```

PLC

Declaration as Hardware Output

Wenn Sie den Ausgang als "wahr" deklarieren, wird er als Hardware-Ausgang deklariert und mit dem Attribut "AT %Q*" in die Programmierlogik aufgenommen. Das bedeutet, dass die Variable ein Signal oder einen Wert an einen physischen Ausgang des Systems sendet, z. B. von einem Ventil oder einem Wechselrichter.

Zone Output	
Output	o_xZone9
Output Description	Output Text
Output Group	Conveyor
Declaration As Hardware Output	True
Output Mode	Digital
Keep Alive	False

Studio

```
SELMOinUse
GVL_Sequence1_IOS
1  |/// copyright SELMO Technology GmbH by SELMOstudio
2  |/// Version 2023.1.1.7920
3  |/// This function has been automatically generated.
4  |///
5  |/// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  {attribute 'qualified_only'}
7  {attribute 'symbol' := 'readwrite'}
8  VAR_GLOBAL
9      |///Input: no description
10     i_xDecision_1Path1: BOOL;
11     |///Input: no description
12     i_xDecision_1Path2: BOOL;
13     |///Input: Input Text
14     i_xZone7 AT %I*: BOOL;
15     |///Input: no description
16     i_xZone9: BOOL;
17     |///Input: no description
18     i_xZone10: BOOL;
19     |///Output: no description
20     o_xZone_6: BOOL;
21     |///Output: Output Text
22     o_xZone9 AT %Q*: BOOL;
23     |///Output: no description
24     o_xZone10: BOOL;
25     |///Output: no description
26     o_xZonell: BOOL;
27     |///Output: no description
28     o_xZone_12RepeaterIteration1: BOOL;
29     |///Output: no description
30     o_xZone_14RepeaterIteration2: BOOL;
31 END_VAR
```

PLC

Output Mode

Zone Output	
Output	o_xZone9
Output Description	Output Text
Output Group	Conveyor
Declaration As Hardware Output	True
Output Mode	Digital
Keep Alive	False

Studio

Die Art des Signals wird durch den Modus des Ausgangs bestimmt. Dies kann entweder ein digitales oder analoges Signal oder ein Parameter sein. Der Modus des Ausgangs gibt also an, welche Art von Signal erwartet wird und wie dieses Signal interpretiert werden soll. Wenn der Ausgabemodus beispielsweise auf "digital" eingestellt ist, sendet das System ein Signal, das aus einem diskreten Wert besteht, während ein analoger Ausgabemodus ein kontinuierliches Signal sendet. Ein Parametermodus hingegen sendet einen Wert, der einen bestimmten Parameter darstellt. Insgesamt hängt die Art des Signals, das ein System empfängt und verarbeitet, weitgehend von dem eingestellten Ausgabemodus ab.

Output Distribution

Output Parameter List	
Output Distribution	HwZone1: .Parameter1
Output Distribution Stored	True

Diese Funktion kopiert einen Wert in einen anderen Parameter. Der Ausgangswert von Zone wird dem Parameter zugewiesen. Der Ausgabewert kann beliebig im Programm verwendet werden.

Output Distribution Stored

Output Parameter List	
Output Distribution	HwZone1: .Parameter1
Output Distribution Stored	True

Nur gültig, wenn der Ausgang gesetzt wurde. Nach der Sequenzprüfung der Zone bleibt der Wert der Ausgangsverteilung gleich.

Keep Alive

Zone Output	
Output	o_xZone9
Output Description	Output Text
Output Group	Conveyor
Declaration As Hardware Output	True
Output Mode	Digital
Keep Alive	False

Studio

Für die Steuerung der Ausgabe gibt es zwei verschiedene Ansätze, die durch die Eigenschaft "Keep-Alive" bestimmt werden:

Zone 9	Zone 10
0	0
0	0
D2	D1
I	0
I	0
I	0

- Wenn die Eigenschaft "Keep-Alive" auf den Wert "False" gesetzt ist, wird standardmäßig die Ausgabe gesteuert. In diesem Fall wird der Ausgang automatisch deaktiviert, sobald die Rückmeldung (Eingang) erreicht ist.
- Wird dagegen die Eigenschaft "Keep Alive" auf den Wert "True" gesetzt, wird der Ausgang unabhängig von der Rückmeldung gesteuert. Dies bedeutet, dass die Steuerung des Ausgangs nicht von der Rückmeldung abhängt und er unabhängig von der Rückmeldung aktiv bleibt.

1.3.1.3 Zone Out

Die **Zone-Out** wird verwendet für das Ansteuern von Aktoren ohne Feedback z.B. Lampen, usw. Um die Output-Zone zu setzen, ist es notwendig den Operanden 'S' einzutragen.

Zone-Out anlegen

Um eine Zone-Out anzulegen, muss auf das entsprechende Symbol geklickt werden.

The screenshot shows the Selmo Studio software interface. At the top, there is a toolbar with various tools. The 'Zone Out' tool, represented by a red circle with a white dot, is highlighted with a red box. Below the toolbar is a table representing the ladder logic logic.

#	Step	GroupName	Info	Decision_1	1	Group 1	Mem 1:
1	Step 1			Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv
2	Step 14			Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv
3	Decision 1		Path 1 jump to 4 Path 2 jump to 5	Zone 9	Zone 10	Zone 11	Zone Mem 12
4	Step Path 1	Path 1		Zone Mem 12	Zone Mem 12	Zone 14	
5	Step Path 2	Path 2					
6	Step 6	Path 2	Timer value: 5s				
7	Step 7	Path 2					
8	Step 8	Path 2					
9	Jump 9	Path 2	Conditional jump to 5				
10	Step 10	Path 2					
11	Step End						
12	Step 11						
13	Repeater 12		Repeater with 2 iterations to step 3				
14	Step 13						

Studio

Operanden der Zone-Out

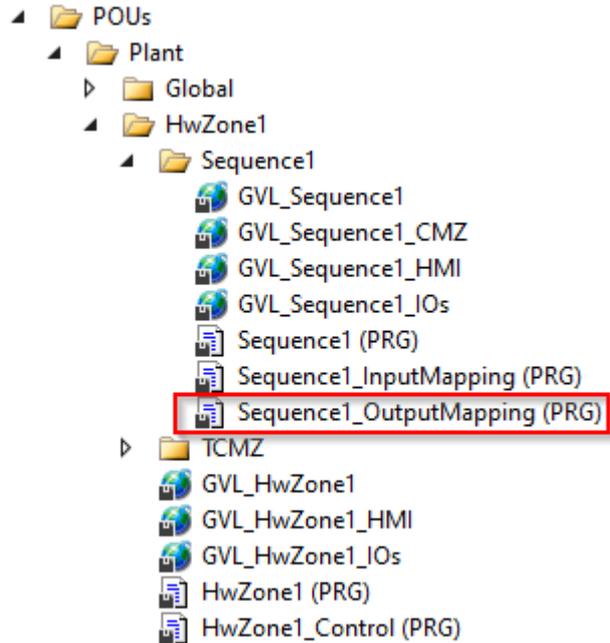
Die Zone-Out schaltet Aktoren und erwartet kein Feedback, sprich sie bedient reine Schaltlogik. Dieser Zonen-Typ wird hauptsächlich für Leuchten und Signale verwendet, jedoch auch für Aktoren, die kein Feedback besitzen. Es sollte jedoch technisch vermieden werden, falls möglich, da dies zu Unsicherheiten im Prozess führt. Der Operand "S" steuert den Ausgang.

		Decision_1					1	Group 1				Mem 1:					
#	Step	GroupName	Info	Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone 11	Zone Mem 12	Zone Mem 12	Zone 14
1	Step 1			0	0	0	0	0	0	S	0	0	0	0	0	0	0
2	Step 14			0	0	0	0	0	0	I	0	0	0	0	0	0	0
3	Decision 1		Path 1 jump to 4 Path 2 jump to 5	0	0	0	0	S	0	I	0	D2	D1	0	0	0	0
4	Step Path 1	Path 1		I	0	0	I	0	0	0	S	I	0	0	S	0	0
5	Step Path 2	Path 2		0	I	I	0	0	0	0	S	I	0	0	0	S	0
6	Step 6	Path 2	Timer value: 5s	0	I	I	0	0	S	0	I	I	0	0	0	I	0
7	Step 7	Path 2		0	I	I	0	0	0	0	I	S	0	S	0	I	0
8	Step 8	Path 2		0	I	I	0	0	0	0	I	0	S	0	0	I	0
9	Jump 9	Path 2	Conditional jump to 5	0	I	I	0	0	0	J	I	S	0	0	0	I	0
10	Step 10	Path 2		0	I	I	0	0	0	0	I	0	S	S	0	I	0
11	Step End			0	I	I	0	0	0	0	I	S	0	0	0	0	0
12	Step 11			0	0	0	0	0	0	0	I	I	0	S	0	0	0
13	Repeater 12		Repeater with 2 iterations to step 3	0	0	0	0	0	0	C	I	0	S	0	0	0	0
14	Step 13			0	0	0	0	0	0	0	I	S	0	S	0	0	0

Studio

Zuordnung der Signale zu der Zone-Out

Weiteres wird der Output der Zone-Out im Code eingetragen und kann mit einem Signal verlinkt werden, siehe nachfolgenden Code:



PLC

```
1 // copyright SELMO Technology GmbH by SELMOstudio
2 // Version 2023.1.1.7920
3 // This function has been automatically generated.
4 (attribute 'symbol' := 'none')
5 PROGRAM Sequence1_OutputMapping
6 VAR
7 END_VAR
8
9 {region "Description Output Mapping"}
10 (*
11 All Step Sequence Zone Outputs that cannot be directly linked are connected to the real world hardware in this section.
12 This must be done manually.
13 *)
14 {endregion}
15
16 (*
17 := GVL_Sequence1_IOs.o xZone9;
18 := GVL_Sequence1_IOs.o xZone10;
19 := GVL_Sequence1_IOs.o xZone11;
20 *)
```

PLC

Properties

A-Z	
Common	
Name	Zone 11
Group Name	Group 1
HMI	
HMI Display Text	
HMI Button	True
HMI Button Text	Button Zone 11
Zone Output	
Output	o_xZone11
Output Description	Output Text
Output Group	
Declaration As Hardware Output	False
Output Mode	Digital
Output Distribution	
Output Distribution Stored	False
Pair Check	
Pair Check	False
Pair Check Group	9
Internals	
Lamp	31
Lamp Pair	32
Is System Zone	False
Manual Button Name	xManBtn_Zone_11

Studio

Output Mode

Die Art des Signals wird durch den Modus des Ausgangs bestimmt. Dabei kann es sich entweder um ein digitales oder analoges Signal oder um einen Parameter handeln. Der Modus des Ausgangs gibt somit an, welche Art von Signal erwartet wird und wie dieses Signal interpretiert werden soll. Wenn der Ausgangsmodus beispielsweise auf "digital" eingestellt ist, sendet das System ein Signal, das aus einem diskreten Werte besteht, während bei einem analogen Ausgangsmodus ein kontinuierliches Signal gesendet wird. Bei einem Parametermodus hingegen wird ein Wert gesendet, der einen bestimmten Parameter repräsentiert. Insgesamt hängt die Art des Signals, welches ein System empfängt und verarbeitet, maßgeblich vom eingestellten Ausgangsmodus ab.

AnalogValue

Output Mode	AnalogValue
Output Analog Value	99
Output Distribution	Para OutDistribution Tes
Output Distribution Stored	True

Wenn der Sequence Check stattfindet wird dem analogen Ausgang der eingegebene Wert von Output Analog Value übergeben, ansonsten ist es der Wert 0. Diese Funktion ist ebenfalls im PLC-Code unter der entsprechenden Zone zu finden.

AnalogParameter

Output Mode	AnalogParameter
Output Analog Setpoint Paramete	
Output Distribution	Para OutDistribution Tes
Output Distribution Stored	True

Wenn der Sequence Check stattfindet wird dem analogen Ausgang der ausgewählte Parameter übergeben. Diese Funktion ist ebenfalls im PLC-Code unter der entsprechenden Zone zu finden.

ParameterList

Output Mode	ParameterList
Output Parameter List	
Output Distribution	Para OutDistribution
Output Distribution Stored	True

Die 'ParameterList' wird ein Wert dem Ausgangssignal zugewiesen. Pro Schritt können mehrere Werte dem Ausgang zugewiesen werden. Um diese Funktion nutzen zu können, muss bereits eine Parameterliste vorhanden sein oder erstellt werden um diese in der gewünschten Zone auswählen zu können. Die ausgewählte Parameter Liste steht dann in 'Output Parameter List'.

- **Name, Group Name, HMI Display Text, HMI Button, HMI Button Text, Input, Input Description, Input Inverted, Input Delay, Declaration as Hardware Input, Ghost Mode, Keep Alive, Output, Output Description, Output Group, Declaration as Hardware Output, Output Distribution, Output Distribution Stored, PairCheck, PairCheckGroup, Lamp, Lamp Pair, Is System Zone, Manual Button Name**

Siehe Zone-InOut

1.3.1.4 Zone Mem

Es werden automatisch 2 Zonen eingefügt, um einen Speicher der sich setzen und rücksetzen lässt, zu realisieren. Das Setzen und Rücksetzen findet jeweils mit den Operanden 'S' statt. Mit den Operanden 'I' kann der Speicher abgefragt werden. Dies ist sowohl bei der normalen Zone (Abfrage auf logisch 1), als auch bei der invertierten Zone (Abfrage auf logisch 0) möglich.

Zone-Mem anlegen

Um eine Zone-Mem anzulegen, muss auf das entsprechende Symbol geklickt werden.

The screenshot shows the Selmo Studio interface. In the top toolbar, the 'Zone Mem' icon is highlighted with a red box. Below the toolbar, a ladder logic diagram is displayed. The diagram includes a 'Decision_1' block with four paths, a 'Step 6 Timer', and several 'Zone' blocks (Zone 7, Zone 9, Zone 10, Zone 11, Zone 14). A 'Zone Mem 12' block is highlighted with a red box, showing two memory locations: 'Zone Mem 12' (green) and 'Zone Mem 12' (red).

#	Step	GroupName	Info	Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone 11	Zone Mem 12	Zone Mem 12	Zone 14
1	Step 1			0	0	0	0	0	0	S	0	0	0	0	0	0	0
2	Step 14			0	0	0	0	0	0	I	0	0	0	0	0	0	0
3	Decision 1		Path 1 jump to 4 Path 2 jump to 5	0	0	0	0	S	0	I	0	D2	D1	0	0	0	0
4	Step Path 1	Path 1		I	0	0	I	0	0	0	S	I	0	0	S	0	0
5	Step Path 2	Path 2		0	I	I	0	0	0	S	I	I	0	0	0	S	0
6	Step 6	Path 2	Timer value: 5s	0	I	I	0	0	S	0	I	I	0	0	0	I	0
7	Step 7	Path 2		0	I	I	0	0	0	0	I	S	0	S	0	I	0
8	Step 8	Path 2		0	I	I	0	0	0	0	I	0	S	0	0	I	0
9	Jump 9	Path 2	Conditional jump to 5	0	I	I	0	0	0	J	I	S	0	0	0	I	0
10	Step 10	Path 2		0	I	I	0	0	0	0	I	0	S	S	0	I	0
11	Step End			0	I	I	0	0	0	0	I	S	0	0	0	0	0
12	Step 11			0	0	0	0	0	0	0	I	I	0	S	0	0	0
13	Repeater 12		Repeater with 2 iterations to step 3	0	0	0	0	0	0	C	I	0	S	0	0	0	0
14	Step 13			0	0	0	0	0	0	0	I	S	0	S	0	0	0

Studio

Operanden der Zone-Mem

Soll eine Zone-Mem im Schritt überwacht werden, ist es notwendig den Operanden 'I' einzutragen. Mit den Operanden 'S' wird auf einen Zustandswechsel von logisch 0 auf logisch 1 gewartet bzw. der Speicher gesetzt oder zurückgesetzt. Die Zone-Mem wird verwendet, um sich Prozesszustände zu merken.

Home				Tools														
				Add zones				Edit				Grouping		Online				
				Zone In	Zone In-Out	Zone Out	Zone Mem	Copy	Paste	Clone to Inverted	Remove Zone	Edit Zone	Step grouping	Connect to PLC	Zones	Groups	Set operands filter	Filter
#	Step	GroupName	Info	Decision_1 1 Group 1 Mem 1:														
				Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone 11	Zone Mem 12	Zone Mem 12	Zone 14	
1	Step 1			0	0	0	0	0	0	S	0	0	0	0	0	0	0	0
2	Step 14			0	0	0	0	0	0	I	0	0	0	0	0	0	0	0
3	Decision 1		Path 1 jump to 4 Path 2 jump to 5	0	0	0	0	S	0	I	0	D2	D1	0	0	0	0	0
4	Step Path 1	Path 1		I	0	0	I	0	0	0	S	I	0	0	S	0	0	0
5	Step Path 2	Path 2		0	I	I	0	0	0	0	S	I	0	0	0	S	0	0
6	Step 6	Path 2	Timer value: 5s	0	I	I	0	0	S	0	I	I	0	0	0	I	0	0
7	Step 7	Path 2		0	I	I	0	0	0	0	I	S	0	S	0	I	0	0
8	Step 8	Path 2		0	I	I	0	0	0	0	I	0	S	0	0	I	0	0
9	Jump 9	Path 2	Conditional jump to 5	0	I	I	0	0	0	J	I	S	0	0	0	I	0	0
10	Step 10	Path 2		0	I	I	0	0	0	0	I	0	S	S	0	I	0	0
11	Step End			0	I	I	0	0	0	0	I	S	0	0	0	0	0	0
12	Step 11			0	0	0	0	0	0	0	I	I	0	S	0	0	0	0
13	Repeater 12		Repeater with 2 iterations to step 3	0	0	0	0	0	0	C	I	0	S	0	0	0	0	0
14	Step 13			0	0	0	0	0	0	0	I	S	0	S	0	0	0	0

Studio

Properties

A-Z	
▲ Common	
Name	Zone Mem 12
Group Name	Mem 12
▲ HMI	
HMI Display Text	
HMI Button	False
▲ Zone Output	
Output	o_xZoneMem12
Output Description	
Zone Mem Parent	
Declaration As Hardware Output	False
Output HMI Reset	True
Output Distribution	
Output Distribution Stored	False
▲ Pair Check	
Pair Check	True
Pair Check Group	10
▲ Internals	
Lamp	33
Lamp Pair	34
Is System Zone	False
Manual Button Name	xManBtn_Zone_Mem_12_Me

Studio

- Name, Group Name, HMI Display Text, HMI Button, HMI Button, TextOutput, PairCheck, PairCheckGroup, Lamp, Lamp Pair, Is System Zone, Manual Button Name

Siehe Zone-InOut

Output HMI Reset

Diese Funktion betrifft den Reset Button auf der HMI. Wenn diese Funktion auf True gesetzt ist, dann wird die Memory Zone zurückgesetzt.

1.3.2 Bit Controlled

Im System Layer wird die logische Schrittfolge aus dem Logic Layer mit Zonen verbunden. Die Operanden der Zonen steuern und überwachen den Prozessablauf. In jeder Sequence lässt sich ein eigener System Layer definieren. Dadurch können Signale mit Zonen aus verschiedenen Sequences verknüpft werden.

Operanden

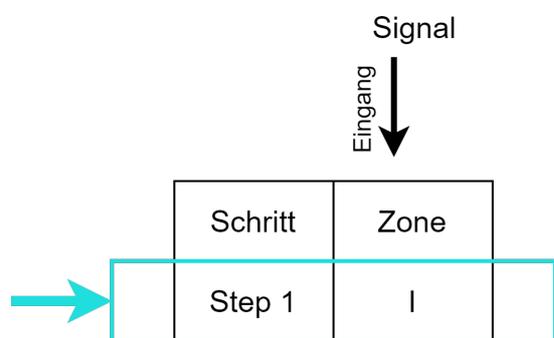
'0' don't care

Das bedeutet es findet keine Überwachung der Zone statt.

Handelt es sich um eine Output-Zone wird damit der Ausgang ausgeschaltet.

'I' Interlock-Check

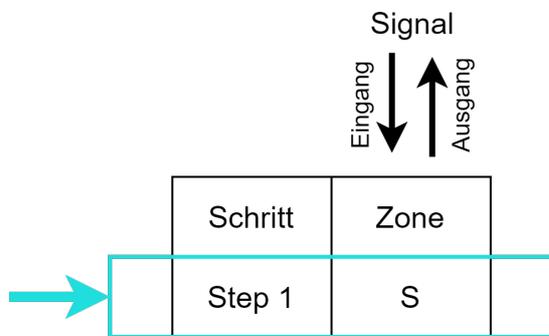
Der Interlock Check ist eine Funktion, welche den Zustand des mit der Zone verknüpften Eingangssignals überwacht. Ist das Signal der Zone nicht im geforderten Zustand, wird der Automatik-Betrieb der Sequence unterbrochen.



Beim Interlock Check (Operand I) wird der Wert des mit der Zone verknüpften Signals überwacht. Wert des Signals ist true.

'S' Sequence-Check

Der Sequence Check ist eine Funktion, die in einem Schritt einer Zone über den Operand S gesetzt werden kann. Der boolesche Wert des mit der Zone verknüpften Signals (Input der Zone) wird auf eine Zustandsänderung geprüft. Das Signal kann die Werte 0 (false) oder 1 (true) annehmen. Eine Zustandsänderung des Signals wird als Transition bezeichnet und ist der Übergang von 0 auf 1, oder false auf true. Solange die Transition nicht durchgeführt ist, bleibt die Bedingung für eine Schrittweitschaltung false und der Schritt aktiv. Ist die Transition beendet wird die Bedingung für eine Schrittweitschaltung true und der Schrittzähler wird erhöht. Für Zone-InOut gilt, dass mit dem Sequence Check der Ausgang so lange geschaltet wird, bis die Rückmeldung erfüllt ist. Bei Zone-Out wird nur der Ausgang gesetzt und bleibt aktiv, bis durch die Transition einer anderen Zone die Schrittweitschaltung erfüllt ist. Die Zone-Out hält in keinem Fall die Schrittweitschaltung an.



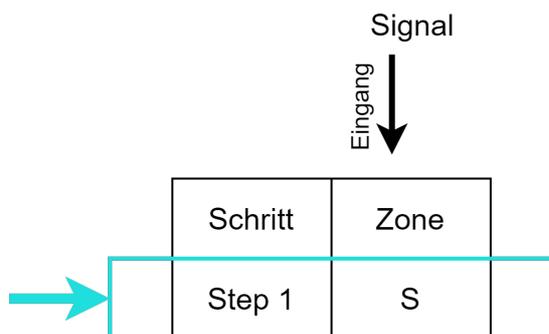
Beim Sequence Check (Operand S) wird die Transition des mit der Zone verknüpften Signals gesteuert und überwacht. Wert des Signals geht von 0 auf 1, oder false auf true.

Invertiertes Signal

Transitionen von 1 auf 0 eines Signals kann man durch Zonen, welche ein invertiertes Signal als Input verwenden überprüfen.

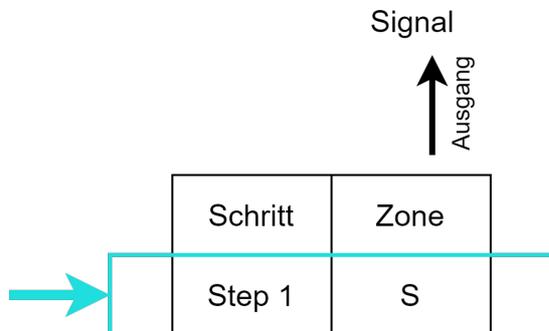
Input-Zone

Es wird auf einen Zustandswechsel von logisch 0 auf 1 gewartet.



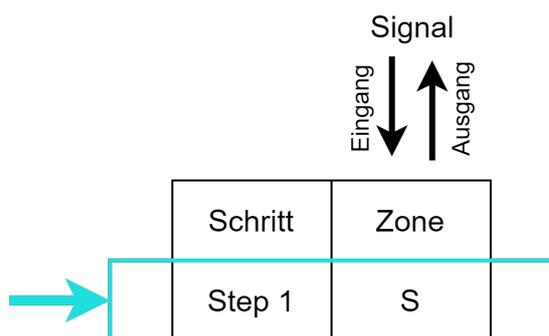
Output-Zone

Wird verwendet, um den Ausgang logisch 1 zu setzen.



In-Out-Zone

Es findet zusätzlich eine Auswertung der Input-Zone statt, der Output bleibt so lange aktiv bis die Input-Zone aktiv wird.



'M' Monitoring-Check

Gleiche Funktion wie 'Interlock-Check', es wird jedoch kein Interlock-Fehler ausgelöst und die Automatik bleibt aktiv.

Der Interlock wird in der Alarm-History mitgeloggt.

'D1' Decision Path 1

Bietet die Möglichkeit, eine oder mehrere Input-Zonen, als Bedingung für Path 1, in der Decision zu definieren.

'D2' Decision Path 2

Bietet die Möglichkeit, eine oder mehrere Input-Zonen, als Bedingung für Path 2, in der Decision zu definieren.

'J' Jump

Bietet die Möglichkeit, eine oder mehrere Input-Zonen, im Jump-Step als Conditional-Jump Variable zu definieren.

'C' Cancel

Wird verwendet, um eine Input-Zone als Abbruchvariable für den Repeater zu definieren.

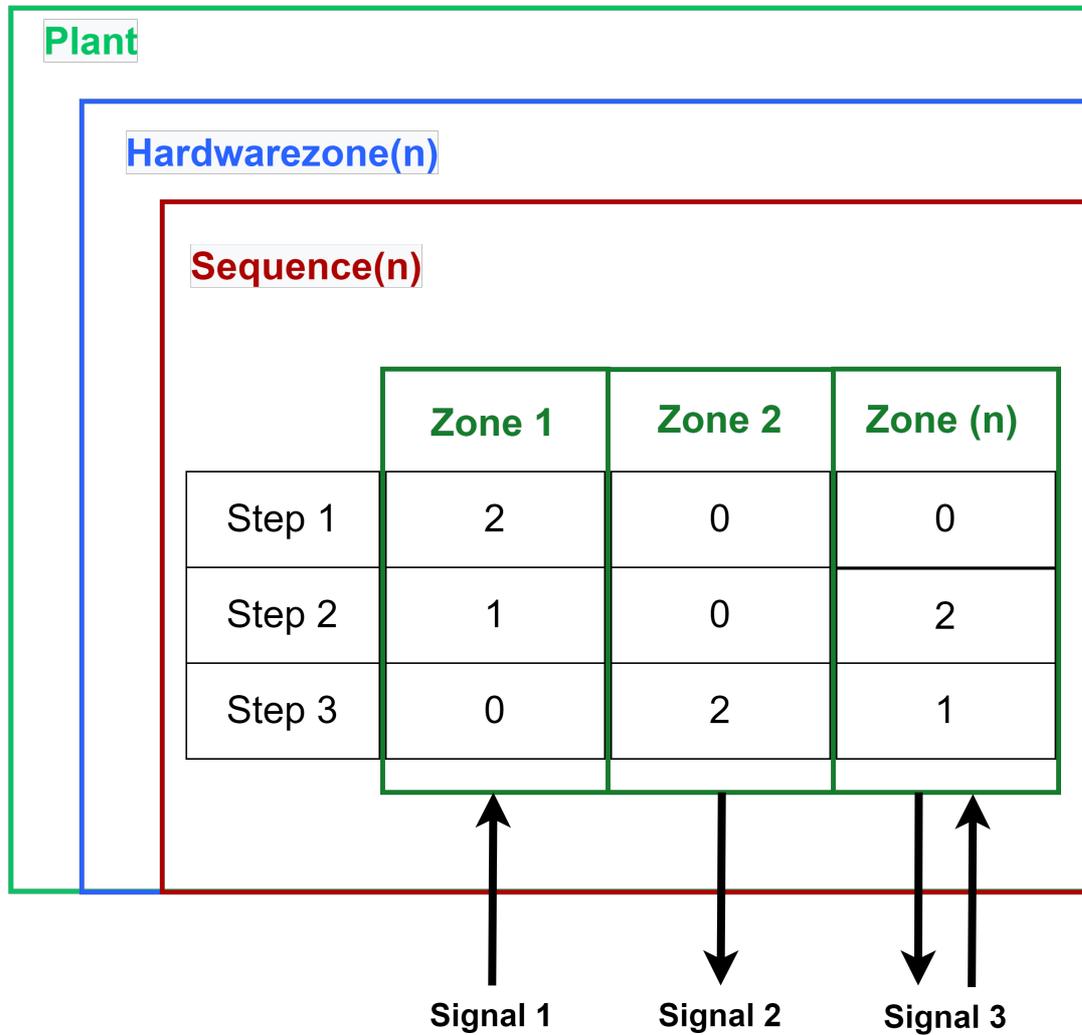
Mit Hilfe der Operanden wird nun definiert in welchem Schritt das System welchen Zustand haben muss bzw. welchen Zustand das System einnehmen muss. Mittels der Zone-InOut und Zone-Out wird definiert welche Outputs in welchem Schritt geschaltet werden. Das wiederum bedeutet, dass die definierten Zustände auch die einzig gültigen sein können. Das System zeigt alle anderen Zustände der Anlage als Fehler an und diagnostiziert die Abweichung zum Soll-Zustand.

Step	Group/Name	Info	Decision_1 Path 1 In	Decision_1 Path 1 Out	Decision_2 Path 2 In	Decision_2 Path 2 Out	Decision_3 XOR	Step 6 timer	Zone 7 In	Zone 7 Out	Zone 9	Zone 10	Zone 11
1	Step 1		0	0	0	0	0	S	0	I	0	0	0
2	Decision 1	Path 1 jump to 3 Path 2 jump to 4	0	0	0	0	S	0	I	0	D2	D1	0
3	Step Path 1		1	0	0	0	0	0	S	I	0	0	0
4	Step Path 2		0	1	0	0	0	0	S	I	0	0	0
5	Step 6	Timer value 5s	0	1	0	0	0	S	0	I	0	0	0
6	Step 7		0	1	0	0	0	0	I	S	0	S	0
7	Step 8		0	1	0	0	0	0	I	0	S	0	0
8	Jump 9	Conditional jump to 4	0	1	0	0	0	J	I	S	0	0	0
9	Step 10		0	1	0	0	0	0	I	0	S	S	0
10	Step End		0	1	0	0	0	0	I	S	0	0	0
11	Step 11		0	0	0	0	0	0	1	1	0	S	0
12	Repeater 12	Repeater with 2 iterations to step 2	0	0	0	0	0	0	C	I	0	S	0
13	Step 13		0	0	0	0	0	0	I	S	0	S	0

Studio

Soll-Ist-Vergleich

Die Bit-genaue Definition der Ein- und Ausgangssignale ermöglicht einen ständigen Soll-Ist-Vergleich der Anlage mit dem definierten Bit-Muster je Prozessschritt, jede Abweichung wird erkannt und die Abweichung zum Soll-Zustand angezeigt.



1.3.3 CMZ

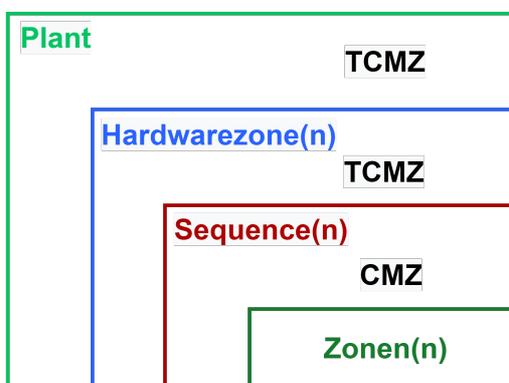
Eine "Constantly Monitored Zone" CMZ bei Maschinen bezieht sich auf einen Bereich oder eine Zone in einer Maschine, die ständig von Sensoren oder anderen Überwachungsmechanismen überwacht wird. In diesen Zonen können beispielsweise kritische Komponenten oder Prozesse einer Maschine liegen, bei denen das Auftreten eines Fehlers oder einer Störung schwerwiegende Konsequenzen haben kann. Die Überwachung dieser Zonen kann durch verschiedene Arten von Sensoren erfolgen, wie z.B. Temperatur- oder Drucksensoren, die kontinuierlich die Werte in der Zone messen und an das Steuerungssystem der Maschine übermitteln. Das Steuerungssystem kann dann bei Abweichungen von den normalen Werten entsprechende Maßnahmen einleiten, um potenzielle Probleme zu vermeiden oder zu minimieren.

Beispiele für Constantly Monitored Zones in Maschinen sind beispielsweise:

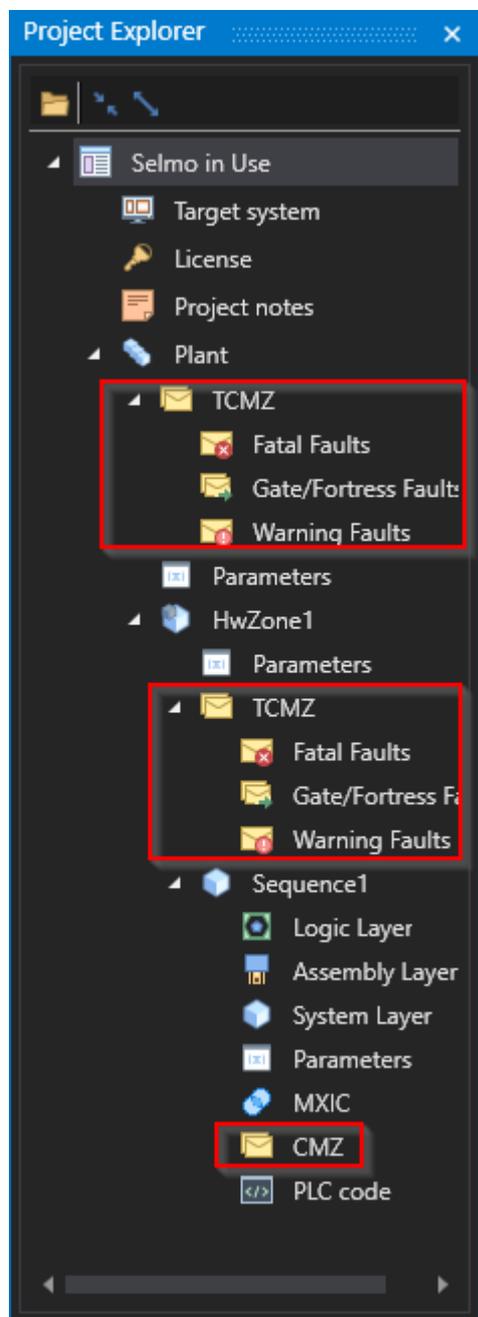
- Die Zone um den Bearbeitungsbereich in einer CNC-Maschine, wo die Temperatur des Schneidwerkzeugs und des Materials, das geschnitten wird, ständig überwacht wird, um Überhitzung und Beschädigung zu vermeiden.
- Die Zone um den Brenner in einem Industrieofen, wo die Temperatur und die Gaszufuhr ständig überwacht werden, um die Brennqualität und Sicherheit zu gewährleisten.
- Die Zone um die Schweißstelle in einer Roboterschweißmaschine, wo die Temperatur und der Druck der Schweißausrüstung überwacht werden muss, um eine gleichmäßige Schweißqualität sicherzustellen.

Insgesamt ist die Überwachung von Constantly Monitored Zones bei Maschinen ein wichtiger Bestandteil der Sicherheits- und Qualitätskontrolle, um einen reibungslosen Betrieb und den Schutz von Mensch und Maschine zu gewährleisten.

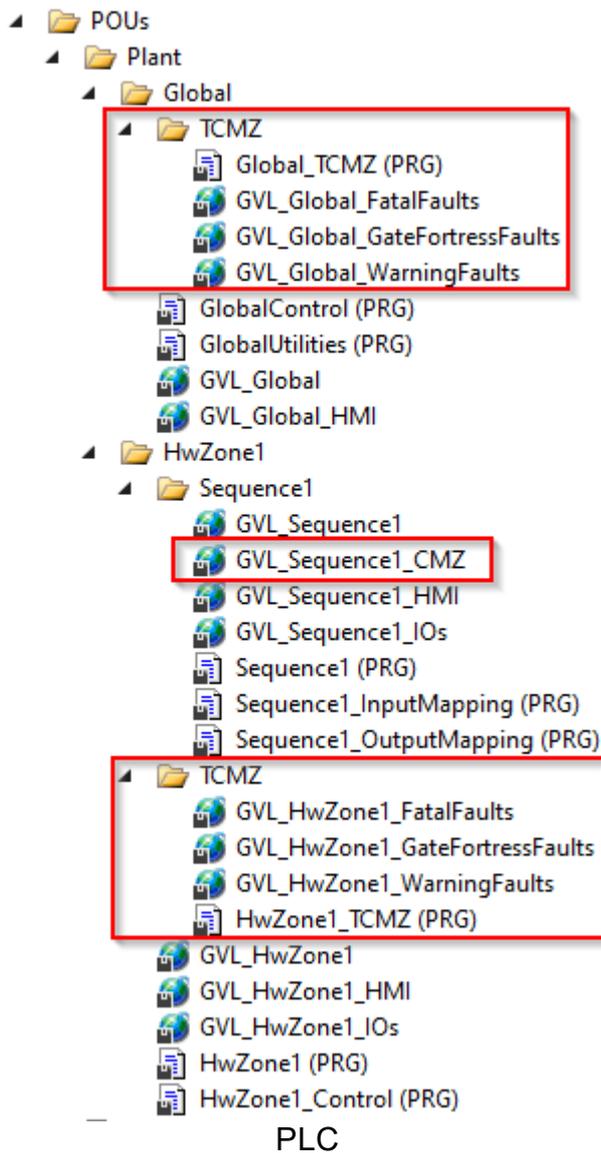
Die CMZ sind auf der Plant-, Hardwarezonen- und Sequenceebene zu finden. Die Einteilung in diese Ebenen legt fest, welche Ebene bei einem Fehlerfall abgeschaltet wird. Wenn ein Fehler auf der Plant-Ebene auftritt, wie zum Beispiel Druckluftverlust, wird die Automatik der gesamten Anlage gestoppt. Fehler auf der Hardwarezonen-Ebene führen zum Stopp der jeweiligen Hardwarezone und Fehler in Sequences stoppen die betreffende Sequence. Die Fehler werden in der Alarmhistorie protokolliert.



Je nachdem, in welcher Ebene die CMZ deklariert wurde, entsteht im PLC-Code ein entsprechender Eintrag. Dies kann entweder auf der Plant-, Hardwarezonen- oder Sequenceebene erfolgen.



Studio



Die CMZ werden mithilfe eines Editors bearbeitet und erstellt. Dabei werden verschiedene Optionen zur Verfügung gestellt, die im Folgenden erläutert werden:

VariableName	Variable Type	Hmi Text	Section	IV	HI	Parameter	Parameter Mode	Window Positive Parameter	Window Negative Parameter	Auto Reset	Error Delay [ms]
xCMZ_1	BOOL	CMZ 1		<input type="checkbox"/>	<input type="checkbox"/>	Equals				<input type="checkbox"/>	0

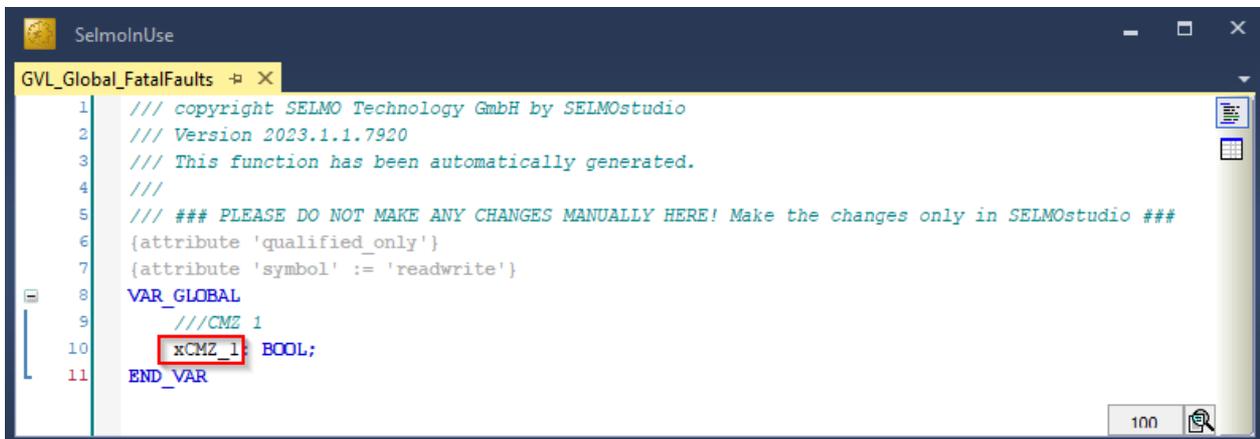
Studio

Variable Name

Der Name einer Variablen ist wichtig für die Lesbarkeit und Verständlichkeit des Codes. Eine gute Variable sollte eindeutig, aussagekräftig und funktionell sein. Das heißt, sie sollte nur eine Bedeutung haben, den Inhalt oder Zweck der Variable beschreiben und mit dem Datentyp und der Logik des Programms übereinstimmen.

Hinweis:

Selmo verwendet die PLCopen Coding Guidelines. Dies ist ein international anerkannter Standard für die Programmierung von Automatisierungssystemen, insbesondere für die Programmierung von speicherprogrammierbaren Steuerungen (SPS). Die PLCopen Coding Guidelines legen Regeln und Best Practices fest, die den Entwicklern helfen, sicherzustellen, dass ihre SPS-Programme lesbar, wiederverwendbar und robust sind. Die Guidelines umfassen eine Vielzahl von Themen, einschließlich Namensgebung von Variablen, Kommentierung von Code, Programmstruktur und Fehlerbehandlung. Die Namensgebung von Variablen in der PLCopen Coding Guidelines folgt bestimmten Regeln. So sollen beispielsweise Variablennamen aussagekräftig sein und den Zweck und den Typ der Variable widerspiegeln. Der Name sollte in englischer Sprache verfasst sein und sich an bestimmte Konventionen halten, wie zum Beispiel die Verwendung von CamelCase. Weiterhin werden Regeln für die Benennung von Eingangs- und Ausgangsvariablen, temporären Variablen und Konstanten festgelegt. Insgesamt zielt die PLCopen Coding Guideline darauf ab, die Lesbarkeit, Wartbarkeit und Robustheit von SPS-Programmen zu verbessern und somit zu einer höheren Effizienz und Produktivität bei der Entwicklung von Automatisierungssystemen beizutragen.

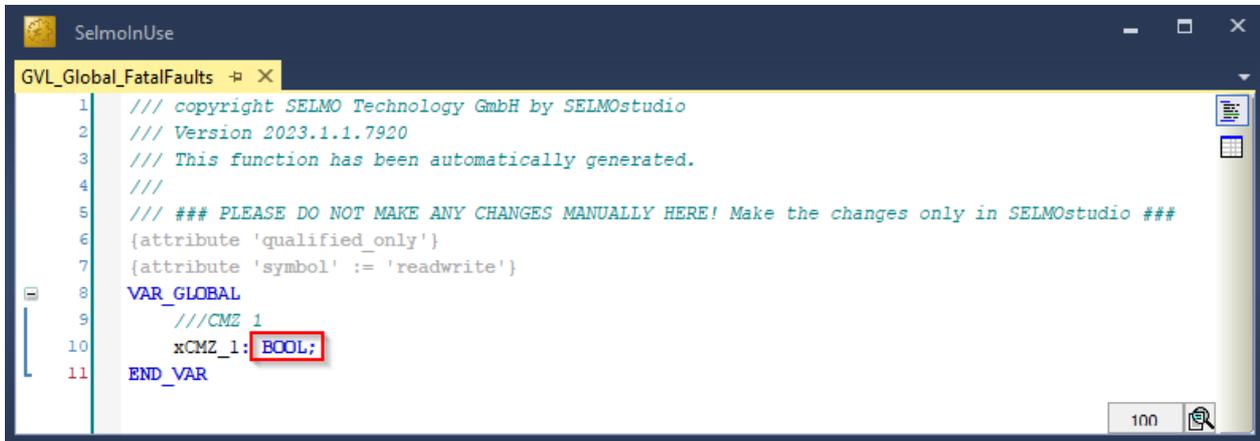


```
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  ///
5  /// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  {attribute 'qualified_only'}
7  {attribute 'symbol' := 'readwrite'}
8  VAR_GLOBAL
9      ///CME 1
10     xCMZ_1  BOOL;
11 END_VAR
```

PLC

Variable Type

Die Deklaration des Variablentyps erfolgt standardmäßig als Boolean-Datentyp. Über das Dropdown-Menü kann jedoch der entsprechende Datentyp ausgewählt werden. Die verfügbaren Datentypen entsprechen der Liste der Standarddatentypen.

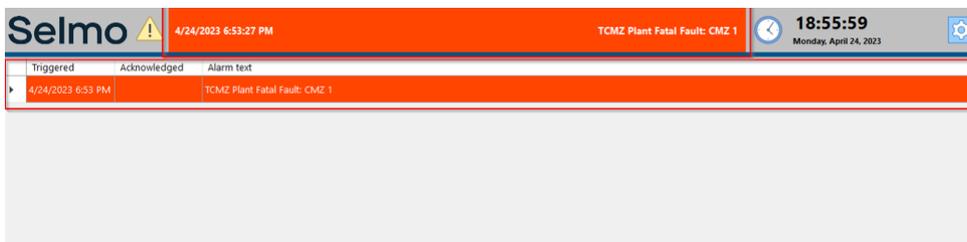


```
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  ///
5  /// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  {attribute 'qualified_only'}
7  {attribute 'symbol' := 'readwrite'}
8  VAR_GLOBAL
9  ///CME 1
10  xCMZ_1: BOOL;
11  END_VAR
```

PLC

HMI Text

Dieser Text erscheint auf der HMI, wenn ein Fehler oder eine Warnung auftritt. Er informiert den Benutzer über die Art und den Ort des Problems. Der Text wird sowohl in der Alarm Bar als auch auf der Alarm Page der HMI angezeigt und dient außerdem als Kommentar der Variable im PLC Code.



HMI

Section

Dies dient der Gruppierung und verbesserten Übersichtlichkeit der CMZ.

VariableName	Variable Type	Hmi Text	Section	IV	HI	Parameter	Parameter Mode	Window Positive Parameter	Window Negative Parameter	Auto Reset	Error Delay [ms]
xCMZ_1	BOOL	CMZ 1		<input type="checkbox"/>	<input type="checkbox"/>	...	Equals	<input type="checkbox"/>	0
xCMZ_2	BOOL	CMZ 2		<input type="checkbox"/>	<input type="checkbox"/>	...	Equals	<input type="checkbox"/>	0
xCMZ_3	BOOL	CMZ 3		<input type="checkbox"/>	<input type="checkbox"/>	...	Equals	<input type="checkbox"/>	0
xCMZ_4	BOOL	CMZ 4		<input type="checkbox"/>	<input type="checkbox"/>	...	Equals	<input type="checkbox"/>	0

Studio

Inverted

Wenn das Feld "Inverted" aktiviert wird, wird der Wert der verknüpften Variable im PLC-Code der entsprechenden Ebene invertiert.

Dies ist nur mit booleschen Variablen Typen möglich

```

SelmoInUse
HwZone1_TCMZ
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  {attribute 'symbol' := 'readwrite'}
5  PROGRAM HwZone1_TCMZ
6  VAR
7  {attribute 'symbol' := 'none'}
8  fbFatalFault: FB_CMZFault;
9  aFatalFault: ARRAY[0..1] OF BOOL;
10 {attribute 'symbol' := 'none'}
11 fbGateFortressFault: FB_CMZFault;
12 aGateFortressFault: ARRAY[0..1] OF BOOL;
13 {attribute 'symbol' := 'none'}
14 fbWarningFault: FB_CMZFault;
15 aWarningFault: ARRAY[0..1] OF BOOL;
16 END VAR

1  // ##### fatal faults #####
2  // TCMZ assigns
3  aFatalFault[0] := GVL_Global.stGlobalIf.xFatalFault;
4  aFatalFault[1] := NOT GVL_HwZone1_FatalFaults.xCMZ_1;
5
6  // TCMZ call
7  fbFatalFault(aFaultMatrix := aFatalFault, xFaultActive => GVL_HwZone1.stHwzIf.xFatalFault);
8
9  // ##### gate / fortress faults #####
10 // TCMZ assigns
11 aGateFortressFault[0] := GVL_Global.stGlobalIf.xGateFortressFault; aGateFortressFault[1] := GVL_HwZone1_GateFortressFaults.xCMZ_1;
12
13 // TCMZ call
14 fbGateFortressFault(aFaultMatrix := aGateFortressFault, xFaultActive => GVL_HwZone1.stHwzIf.xGateFortressFault);
15
16 // ##### warning faults #####
17 // TCMZ assigns
18 aWarningFault[0] := GVL_Global.stGlobalIf.xWarningFault; aWarningFault[1] := GVL_HwZone1_WarningFaults.xCMZ_1;
19
20 // TCMZ call
21 fbWarningFault(aFaultMatrix := aWarningFault, xFaultActive => GVL_HwZone1.stHwzIf.xWarningFault);
22

```

PLC

Declaration as Input

Wenn Sie die CMZ als "True" deklarieren, wird sie als Hardware-Eingang deklariert und mit dem "AT %I*" Attribut in der Programmierungslogik eingebunden. Dies bedeutet, dass die Variable ein Signal oder einen Wert von einem physikalischen Eingang des Systems empfängt, wie beispielsweise von einem Sensor oder einem Schalter.

```

1  /// Copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  ///
5  /// ### PLEASE DO NOT MAKE ANY CHANGES MANUALLY HERE! Make the changes only in SELMOstudio ###
6  {attribute 'qualified_only'}
7  {attribute 'symbol' := 'readwrite'}
8  VAR_GLOBAL
9  ///CMZ 1
10 xCMZ_1 AT %I*: BOOL;
11 END_VAR

```

PLC

HardwareInput

Präfix wird im generierten SPS-Code bei der Variablendeklaration hinzugefügt.

Parameter

Variablen mit einer Funktion (z.B Temperatur Maximum).

Parameter Mode

Auswahl zwischen None, Greater Than, Less Than, Equals, GreaterEquals, LessEquals, NotEquals und Window.

Window Positive Parameter und Window Negative Parameter

Eine Zahl wird verwendet, um diesen Parameter zu initialisieren.

Um den Window Mode auswählen zu können muss der Variable Type etwas anderes als BOOL sein. Window ist eine Vergleichsfunktion und fungiert wie ein Toleranzfenster. Die Window Funktion kann in der HMI immer angepasst werden. Der Window Positive Parameter ist der Maximale Wert und der Window Negative Parameter ist der Minimale Wert.

Auto Reset (Option nur in der Seqence CMZ vorhanden)

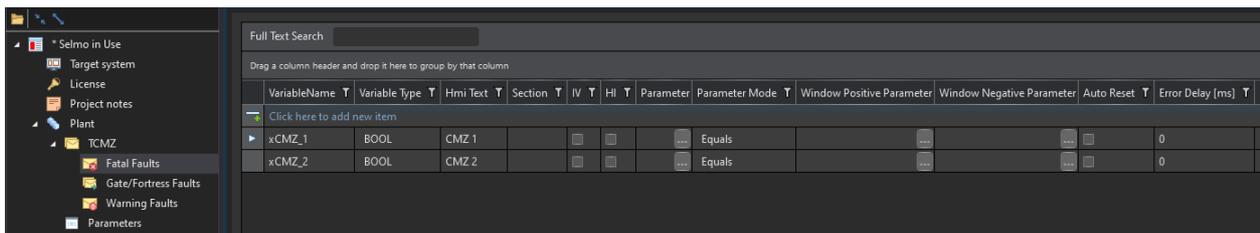
Ein Auto Reset im Fehlerfall bezeichnet eine Funktion, bei der ein Fehler automatisch zurückgesetzt wird, ohne dass manuell eingegriffen werden muss. Wenn ein Fehler auftritt, wird dieser automatisch erkannt und der Systemzustand wird auf den normalen Betriebszustand zurückgesetzt. Diese Funktion wird häufig in automatisierten Systemen verwendet, um sicherzustellen, dass der Betrieb fortgesetzt werden kann, ohne dass ein Bediener manuell eingreifen muss, um den Fehler zu beheben. Das Auto Reset im Fehlerfall ist besonders nützlich in kritischen Anwendungen, bei denen ein sofortiges Eingreifen notwendig ist, um Ausfallzeiten oder Schäden an der Ausrüstung zu minimieren.

Error Delay [ms]

Wird eingesetzt, wenn der Fehler zeitverzögert ausgelöst werden soll. Zusammenfassend lässt sich sagen, dass "Error Delay" als Begriff eingesetzt wird, wenn ein Fehler in der Signalverarbeitung auftritt und eine zeitverzögerte Auslösung gewünscht ist. Das Phänomen, bei dem ein Signal kurzzeitig mehrere schnelle Impulse erzeugt, anstatt eines einzigen Impulses, wird als "Signalprellen" oder auch als "Kontaktprellen" bezeichnet.

1.3.3.1 Fatal Faults

Fatal Faults sind schwerwiegende Fehler der Anlage, die den Automatik Modus deaktivieren wenn die Bedingung einer verknüpften Variable eintritt. Beispielsweise wird das Eingangssignal eines Schutzschalters auf die Variable xCMZ_1 verknüpft. Diese Variable des Datentyps bool ist so lange im Betrieb false, bis das Eingangssignal des Schutzschalters den Wert true annimmt. Infolgedessen wird der Automatik Modus sofort deaktiviert. Ein solcher schwerwiegender Fehler muss behoben werden, bevor eine manuelle Freigabe der Anlage wieder möglich ist. Schwerwiegende Fehler können nicht überbrückt werden.



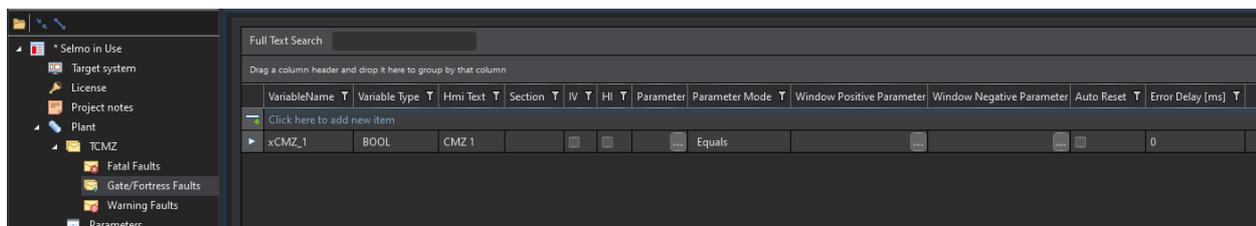
The screenshot shows the Selmo Studio interface. On the left is a navigation tree with folders for 'Selmo in Use', 'Target system', 'License', 'Project notes', 'Plant', 'TCMZ', 'Fatal Faults', 'Gate/Fortress Faults', 'Warning Faults', and 'Parameters'. The main area displays a table with the following columns: VariableName, Variable Type, Hmi Text, Section, IV, HI, Parameter, Parameter Mode, Window Positive Parameter, Window Negative Parameter, Auto Reset, and Error Delay [ms]. Two rows are visible in the table:

VariableName	Variable Type	Hmi Text	Section	IV	HI	Parameter	Parameter Mode	Window Positive Parameter	Window Negative Parameter	Auto Reset	Error Delay [ms]
xCMZ_1	BOOL	CMZ 1		<input type="checkbox"/>	<input type="checkbox"/>		Equals	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
xCMZ_2	BOOL	CMZ 2		<input type="checkbox"/>	<input type="checkbox"/>		Equals	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Studio

1.3.3.2 Gate/Fortress Faults

Eine Zutrittsverletzung wird als Gate/Fortress Fehler bezeichnet. Es können Variablen definiert werden um Signale von Schutztüren, Lichtgitter oder Sensoren zu überwachen. Damit können Schutzbereiche definiert werden. Gibt es im Betrieb eine Übertretung eines solchen Bereiches, wird der Automatik Modus der Anlage deaktiviert und die entsprechende Zutrittsverletzung auf der HMI angezeigt.

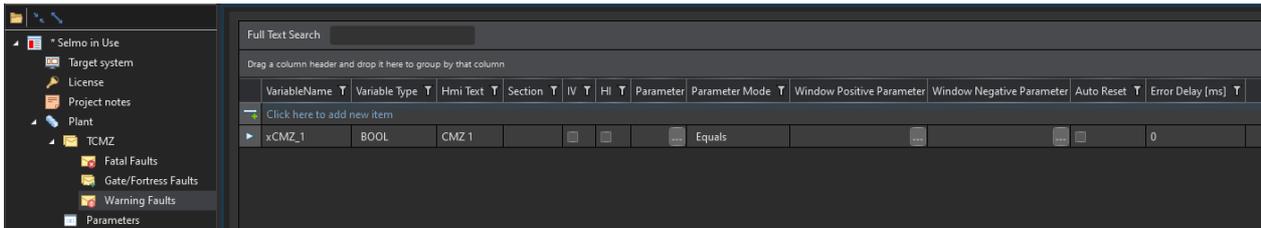


VariableName	Variable Type	Hmi Text	Section	IV	HI	Parameter	Parameter Mode	Window Positive Parameter	Window Negative Parameter	Auto Reset	Error Delay (ms)
xCMZ_1	BOOL	CMZ 1				Equals					0

Studio

1.3.3.3 Warning Faults

Als Warning Fault werden Warnungen einer Anlage bezeichnet, die als Information für den Benutzer gelten aber den Automatik Modus der Anlage nicht gefährden. Treten diese Fehler im Betrieb auf, werden diese auf der HMI angezeigt, der Automatik Modus der Anlage bleibt jedoch erhalten. Warning Faults können dazu verwendet werden, eine sich nähernde Gefährdung wie z.B. Material Fehlbestände, sinkende Füllstände etc. aufzuzeigen um den Benutzer zu warnen.

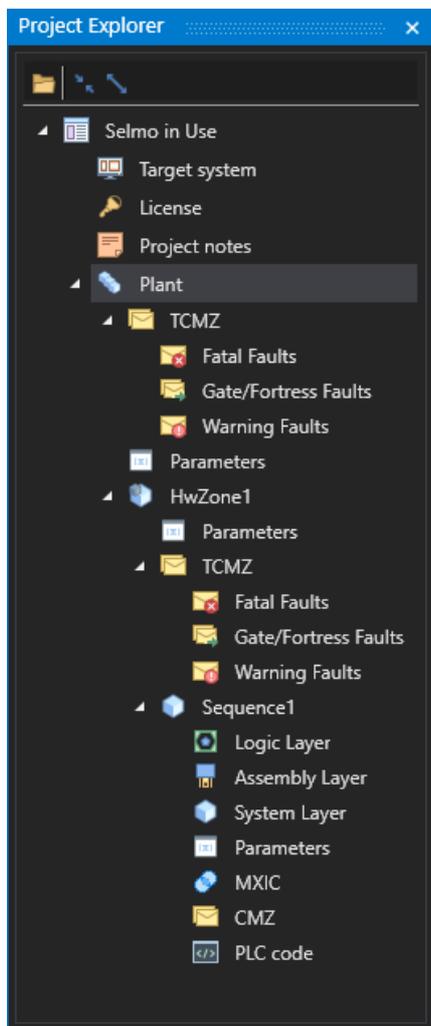


Studio

1.4 Ergebnis nach Schritt 1 - 3

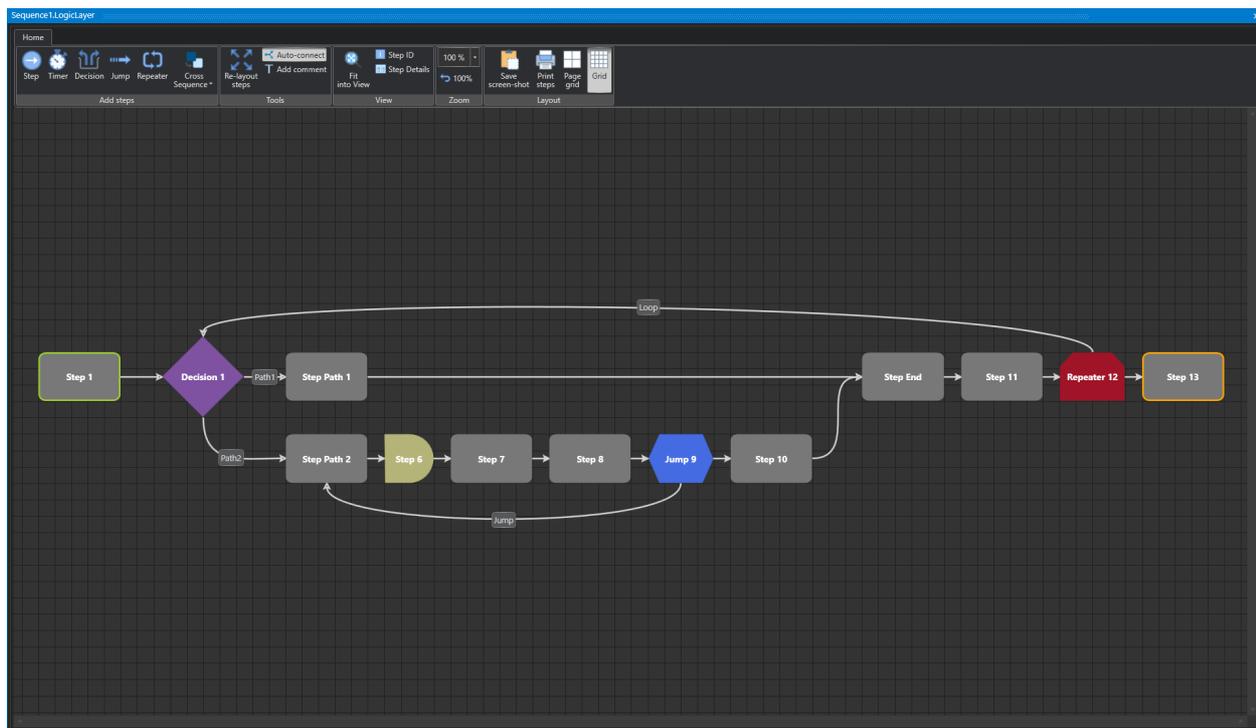
Was ist schon alles vorhanden nach Schritt 1 bis 3

Nach den ersten drei Schritten ist bereits eine grundlegende Struktur der Plant mit einer Hardwarezone und einer Sequence vorhanden.



Studio

Außerdem wurde der logische Ablauf des Prozesses grafisch dargestellt.



Studio

Die Signale wurden mit den entsprechenden Zonen, Zuständen, Operanden und Überwachungen definiert.

Step	Group/Name	Info	Decision_1 Path 1	Decision_1 Path 1 Inv	Decision_1 Path 2	Decision_1 Path 2 Inv	Decision_1 XOR	Step 6 Timer	Zone 7 Inv	Zone 7	Zone 9	Zone 10	Zone 11	Repeater 12 Iteration 1	Repeater 12 Iteration 1 Inv	Repeater 12 Iteration 2	Repeater 12 Iteration 2 Inv
1	Step 1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Decision 1	Path 1 jump to 3 Path 2 jump to 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Step Path 1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Step Path 2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	Step 6	Timer value: 5s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	Step 7		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Step 8		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	Jump 9	Conditional jump to 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	Step 10		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Step End		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	Step 11		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	Repeater 12 Cancel Jump	Conditional jump to 23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	Decision 1 Iteration 2	Path 1 jump to 14 Path 2 jump to 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	Step Path 1 Iteration 2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	Step Path 2 Iteration 2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	Step 6 Iteration 2	Timer value: 5s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	Step 7 Iteration 2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	Step 8 Iteration 2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	Jump 9 Iteration 2	Conditional jump to 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Step 10 Iteration 2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	Step End Iteration 2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	Step 11 Iteration 2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	Repeater 12	Repeater with 2 iterations to step 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	Step 13		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Studio

Damit ist die Basis für die weitere Programmierung und Testung geschaffen.

Was fehlt noch

Mit diesem Schritt haben wir die Grundlage für die weitere Entwicklung und Prüfung unseres Systems gelegt. Nun müssen wir die Details festlegen, welche die Qualität und Sicherheit unserer Lösung gewährleisten. Dazu gehören die Funktionen der Handsteuerung (Manual Cross Interlock Check), welche verhindern, dass sich die Anlage bei falscher Bedienung beschädigt, sowie die Flexibilisierung durch den Parameter Layer, welche es uns ermöglichen, die Einstellungen an verschiedene Anforderungen anzupassen.

1.5 Hand Funktionen

Der Handbetrieb bei Maschinen bezieht sich auf eine Funktion, bei der die Maschine manuell und ohne automatische Steuerung betrieben wird. Im Handbetrieb kann der Bediener einer Maschine bestimmte Funktionen und Bewegungen manuell ausführen, indem er beispielsweise Hebel, Schalter oder Tasten an der Maschine betätigt. Dadurch kann der Bediener die Maschine präzise steuern und bestimmte Funktionen ausführen, die möglicherweise nicht automatisch durch die Maschine ausgeführt werden können.

Der Handbetrieb kann in verschiedenen Situationen nützlich sein, wie beispielsweise bei der Wartung der Maschine, beim Einrichten der Maschine für eine bestimmte Aufgabe oder bei der Fehlersuche, wenn ein Problem auftritt. Wenn ein Bediener im Handbetrieb arbeitet, muss er jedoch besonders vorsichtig sein, da er vollständig für die Kontrolle der Maschine verantwortlich ist und die Sicherheitsrisiken minimieren muss. In einigen Fällen kann es auch erforderlich sein, spezielle Schulungen oder Zertifizierungen für den Handbetrieb von Maschinen zu absolvieren.

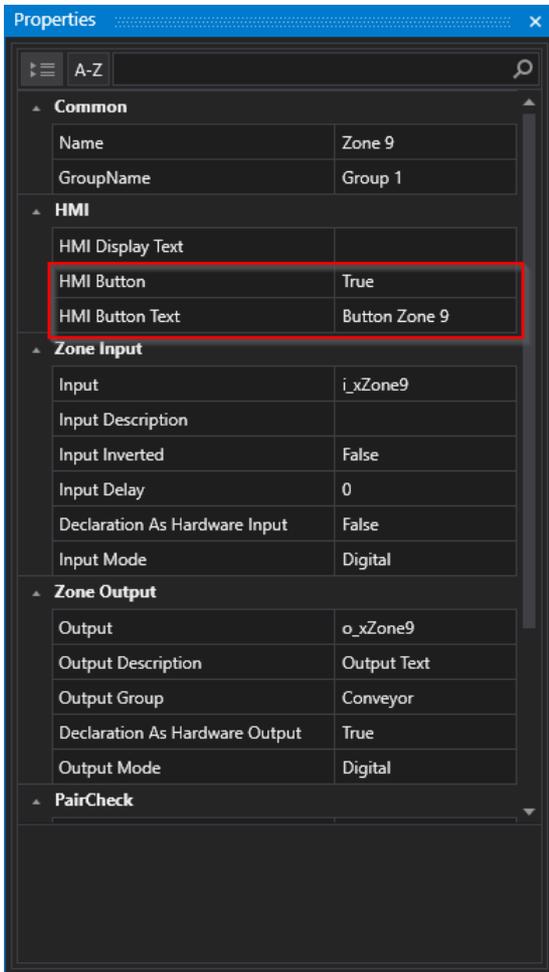
Die Handbedienung ist eine Benutzerschnittstelle, die nur in der Betriebsart Handbetrieb funktioniert und verwendet wird, um den Ausgang zu steuern. Im Gegensatz zur automatischen Betriebsart, bei der das System automatisch arbeitet, gibt der Handbetrieb dem Benutzer die Möglichkeit, manuelle Eingriffe vorzunehmen.

Über einen HMI-Button wird der Ausgang gesteuert, bis ein Feedback-Signal (Input der Zone) aktiv wird. Das Feedback-Signal gibt dem System eine Rückmeldung darüber, ob der gewünschte Zustand erreicht wurde oder ob Änderungen vorgenommen werden müssen. Wenn die MXIC-Kreuzverriegelung aktiv ist, kann der Ausgang nicht angesteuert werden und die HMI zeigt Informationen zur Zone an, mit der die Verriegelung stattfindet.

Die MXIC-Kreuzverriegelung ist eine Sicherheitsfunktion, die verhindert, dass die Zone nur unter bestimmten Voraussetzungen (Zonen x y befinden sich in einem definierten Zustand) aktiviert werden darf. Dies ist besonders bei kritischen Anwendungen wichtig, bei denen das gleichzeitige Auftreten von mehreren Ereignissen zu gefährlichen Situationen führen kann. Durch die Verwendung des HMI-Buttons in Kombination mit der MXIC-Kreuzverriegelung wird sichergestellt, dass nur der gewünschte Ausgang aktiviert wird und dass potenzielle Gefahren vermieden werden.

HMI Tasten

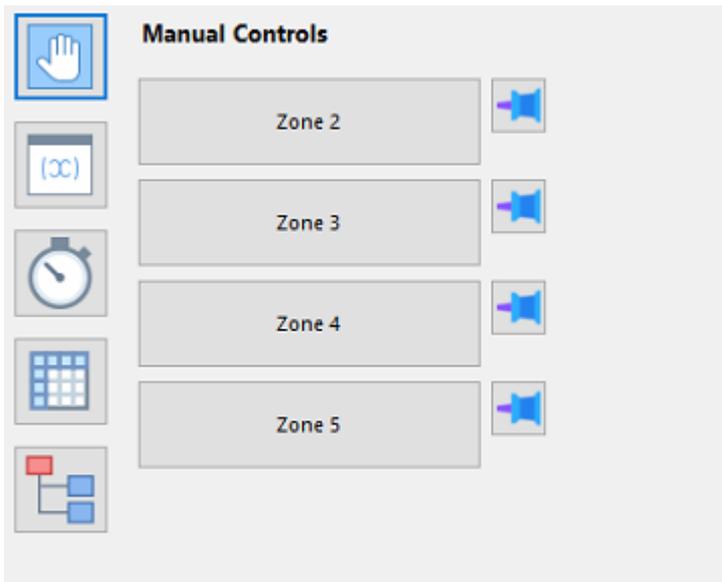
Um HMI-Tasten für eine spezifische Zone (gültig nur bei InOut- und Out-Zonen) zu aktivieren, muss in den Eigenschaften der Zone die Option "HMI Button" aktiviert werden. Der Beschriftungstext der HMI-Taste kann durch die Eigenschaft "HMI Button Text" angepasst werden, z.B. "Zylinder vor", "Achse 1 Position 1" oder "Sende Daten an ERP".



Studio

Die Buttons sind in der HMI in der Sequence unter Manual Controls zu finden.

Mit dem Stecknadel-Symbol ist es möglich, dass man den Wert fixiert.



HMI

MXIC

Die MXIC-Funktion (Manual Cross Interlock Check) verhindert Fehlbedienungen im manuellen Betrieb. In der linken Spalte sind alle manuellen Funktionen aufgeführt und den zugehörigen Zonen gegenübergestellt. Für jede manuelle Funktion wird festgelegt, welche Zonen (mit Feedback) aktiv sein müssen, damit die Handbewegung ausgeführt werden kann. Diese Informationen werden auch zur Diagnose und Bedienerführung verwendet. Wenn der Bediener eine gesperrte Handbewegung ausführen möchte, wird diese blockiert und gleichzeitig wird im HMI angezeigt, warum die Bewegung nicht ausgeführt werden kann.

	MXIC	Manual Button	Manual Button Text	Step 6 Timer Inv	Zone 7	Zone 7 Inv	Zone 9	Zone 10	Zone Mem 12	Zone Mem 12
▶	Zone 9	<input checked="" type="checkbox"/>	Button Zone 9	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Zone 10	<input checked="" type="checkbox"/>	Button Zone 10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	Zone 11	<input checked="" type="checkbox"/>	Button Zone 11	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	Zone Mem 12	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Zone Mem 12	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Studio

Sequence Automatic Release

Manual Controls

Previous step

Actual step
1: Step 1

Next step
2: Decision 1

Waiting for
DEMO MODE --> Reset Automatic mode in 30 minutes!

- Zone 7 On
- Zone Mem 12

Button Zone 9

Button Zone 10

Button Zone 11

HMI

1.6 Flexibilisierung über Parameter

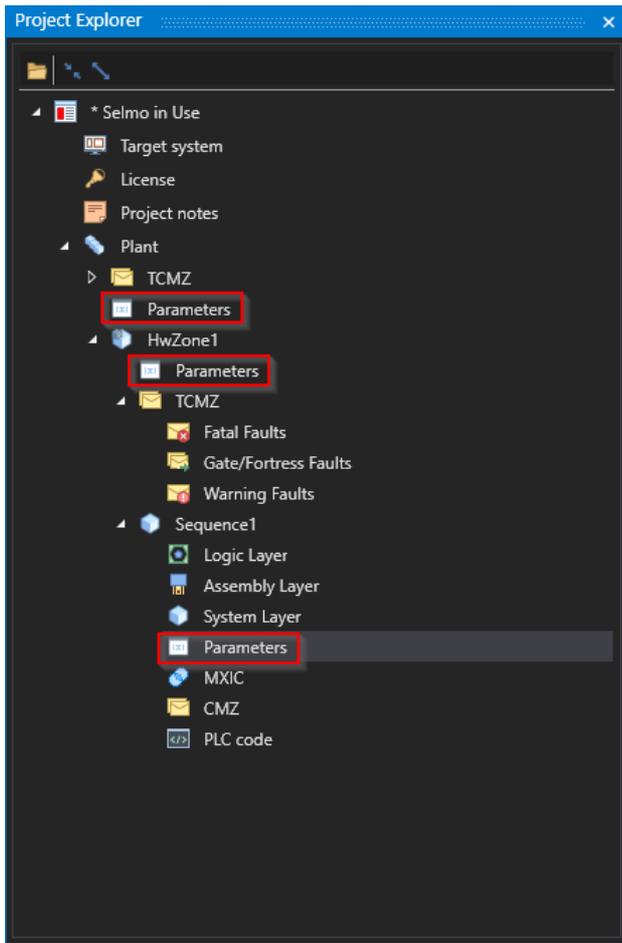
Die Flexibilisierung über Parameter bezieht sich auf die Möglichkeit, ein System oder eine Methode so anzupassen, dass sie durch das Ändern von Parametern an unterschiedliche Bedingungen oder Anforderungen angepasst werden kann. Durch die Flexibilisierung über Parameter können Prozesse oder Systeme schnell und effizient an veränderte Anforderungen angepasst werden, ohne dass umfangreiche Überarbeitungen oder Änderungen notwendig sind. Ein Beispiel hierfür wäre die Anpassung von Produktionsprozessen in der Industrie, um auf veränderte Kundenbedürfnisse oder Marktbedingungen zu reagieren. Indem man bestimmte Parameter wie Produktionsgeschwindigkeit, Produktqualität oder Materialverwendung ändert, kann man die Produktion schnell anpassen, ohne das gesamte Produktionssystem neu zu gestalten.

1.6.1 Parameter Ebenen

Die Parameterebenen der Plant, Hardwarezone und Sequence werden in der Automatisierungstechnik häufig verwendet, um eine hierarchische Strukturierung von Parametern zu ermöglichen. Die Plant-Ebene umfasst alle Parameter, die für eine gesamte Anlage oder ein Gesamtsystem relevant sind. Hier werden beispielsweise Parameter wie der Betriebsmodus oder die Gesamtleistung der Anlage definiert. Die Hardwarezone-Ebene bezieht sich auf spezifische Bereiche oder Komponenten der Anlage, wie z.B. einen bestimmten Förderbandabschnitt oder Maschinenbereich. Hier werden Parameter definiert, die nur auf diesen Bereich oder diese Komponente anwendbar sind. Die Sequence-Ebene bezieht sich auf spezifische Abläufe oder Prozesse innerhalb der Anlage, wie z.B. das Öffnen oder Schließen eines Ventils. Hier werden Parameter definiert, die nur auf diese spezifischen Abläufe oder Prozesse anwendbar sind. Durch diese hierarchische Strukturierung von Parametern wird eine klare und geordnete Verwaltung von Parametern ermöglicht, was wiederum die Wartung und Fehlersuche erleichtert.

Parameter Ebenen sind eine Möglichkeit, die Komplexität von automatisierten Anlagen zu reduzieren. Sie erlauben es, die Anlage in verschiedene Bereiche zu unterteilen, die jeweils eigene Parameter haben. Zum Beispiel kann eine Anlage aus mehreren Hardwarezonen bestehen, die unterschiedliche Sensoren und Aktoren haben. Jede Hardwarezone kann wiederum aus mehreren Sequenzen bestehen, die die Abläufe der Anlage steuern. Eine Sequence kann zum Beispiel einen Teilprozess ein- oder ausschalten, eine Temperatur regeln oder einen Teil einer Station steuern. Durch die Verwendung von Parameter Ebenen kann man die Anlage übersichtlicher gestalten und leichter anpassen.

Parameter Ebenen sind wie folgt aufgeteilt.



Studio

Plant Ebene: Auf der Plant-Ebene werden grundlegende Parameter definiert, die für die gesamte Anlage relevant sind. Dabei handelt es sich um Einstellungen und Vorgaben, die unabhängig von spezifischen Prozessschritten oder Maschinen gelten. Zu den typischen Parametern, die auf dieser Ebene festgelegt werden, gehören beispielsweise Temperaturwarnungen, die sicherstellen, dass die Anlage innerhalb eines bestimmten Temperaturbereichs arbeitet. Darüber hinaus werden hier auch Stückzähler definiert, um beispielsweise die Anzahl der produzierten Produkte zu überwachen. Ein weiterer wichtiger Parameter auf der Plant-Ebene sind Rezepturen, die die genauen Bestandteile und Mengenverhältnisse für die Herstellung eines Produkts festlegen. Durch die Festlegung dieser grundlegenden Parameter auf der Plant-Ebene kann eine effiziente und zuverlässige Produktion sichergestellt werden.

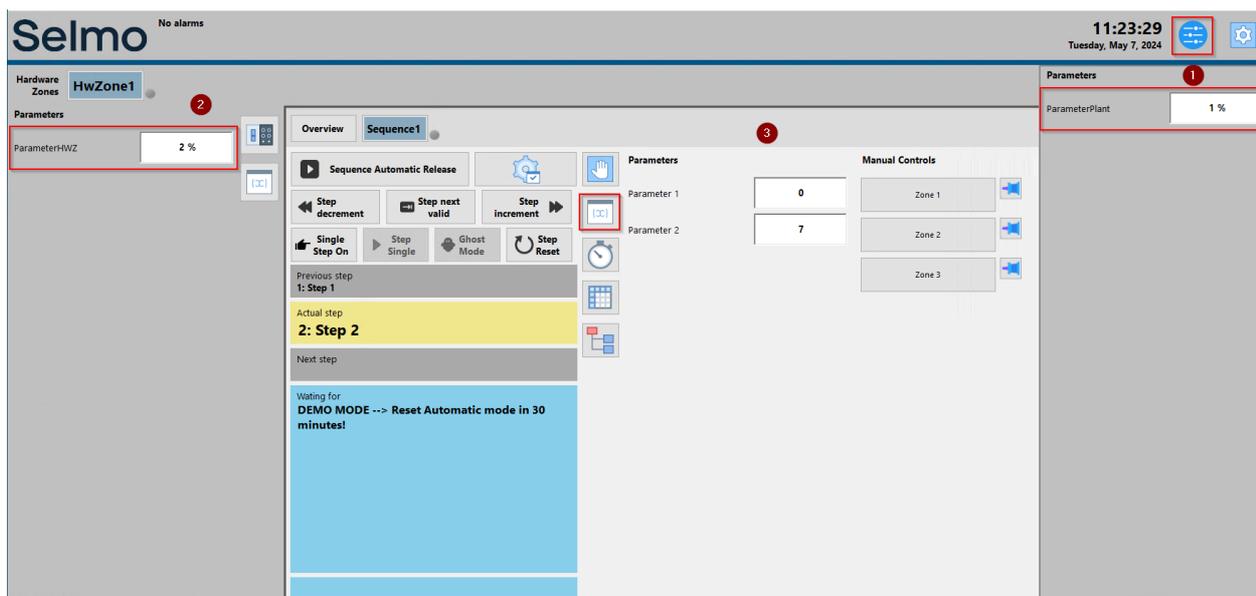
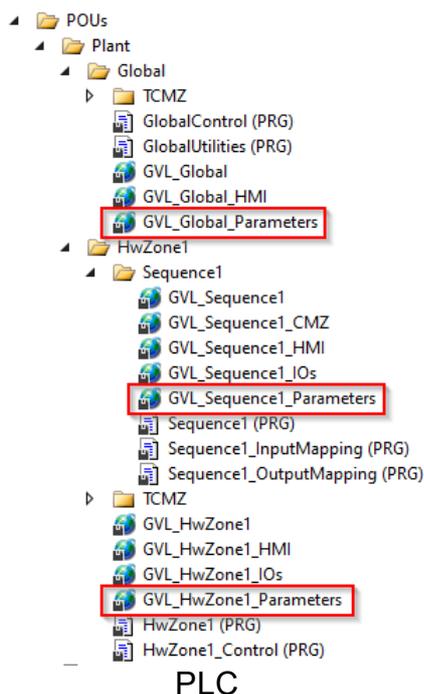
Hardware Zone Ebene: Auf der Hardware-Zone-Ebene werden Parameter definiert, die speziell für einen bestimmten Maschinenbereich gelten. Hier werden die Einstellungen und Vorgaben festgelegt, die nur auf diese spezifische Hardware-Zone zutreffen und nicht auf andere Teile der Anlage. Typische Parameter, die auf dieser Ebene angelegt werden, sind beispielsweise Anzeigeparameter, die den Betrieb und Status des betreffenden Maschinenbereichs anzeigen, wie beispielsweise Temperaturen, Drücke oder Durchflussraten. Ein weiteres Beispiel für einen Parameter, der die Hardwarezone betrifft, sind die Rezepturen des Maschinenbereichs. Hierbei handelt es sich um spezifische Anweisungen, die der Maschine sagen, welche Materialien und in welchen Mengen sie verwenden soll, um ein bestimmtes Produkt herzustellen. Diese Rezepturen können in der Hardwarezone angelegt und gespeichert werden, um sicherzustellen, dass die Maschine bei der Produktion von Produkten konsistent und effizient arbeitet. Weitere Beispiele für Parameter, die die Hardwarezone betreffen könnten, sind die Konfiguration von Sensoren, die Kalibrierung von Messgeräten oder die Festlegung von Sicherheitsparametern wie Not-Aus-Schaltern oder Grenzwerten für Temperaturen und Drücke. Insgesamt bezieht sich das Anlegen von Parametern in der Hardwarezone auf die Feinabstimmung und Konfiguration von Maschinenbereichen, um eine optimale Leistung und Effizienz zu gewährleisten und gleichzeitig die Sicherheit von Mitarbeitern und Betriebsmitteln zu gewährleisten.

Sequence Ebene: Die Sequence Ebene bezieht sich auf die Ebene der Steuerungselemente, die die Reihenfolge der Abläufe in einem Maschinenprozess steuern. In diesem Kontext bezieht sich das Anlegen von Parametern auf die Einstellungen, die nur die Abläufe innerhalb der Sequence betreffen. Ein Beispiel für einen Parameter auf der Sequence Ebene ist die Achsposition. Das bedeutet, dass die Position jeder Achse des Maschinensystems präzise definiert werden muss, damit die Maschine in der Lage ist, die erforderlichen Bewegungen und Prozesse auszuführen. Diese Achspositionen können als Parameter in der Steuerungsebene hinterlegt werden. Ein weiteres Beispiel für einen Parameter auf der Sequence Ebene sind Korrekturwerte. Wenn während des Betriebs der Maschine Unregelmäßigkeiten auftreten, müssen Korrekturwerte definiert werden, um diese zu kompensieren und die Leistung der Maschine zu optimieren. Beispielsweise können Korrekturwerte für Temperatur, Druck oder Geschwindigkeit eingestellt werden, um

sicherzustellen, dass die Maschine stets innerhalb bestimmter Toleranzen arbeitet. Die Geschwindigkeit der Maschine ist ein weiterer wichtiger Parameter auf der Sequence Ebene. Die Geschwindigkeit kann je nach den Anforderungen des Prozesses angepasst werden, um sicherzustellen, dass die Maschine die erforderliche Ausgabe produziert, ohne dabei überlastet zu werden oder unerwünschte Auswirkungen auf das Endprodukt zu haben. Zusammenfassend werden auf der Sequence Ebene Parameter festgelegt, die spezifische Funktionen des Maschinensystems steuern und optimieren. Dazu gehören Achspositionen, Korrekturwerte und Geschwindigkeiten, die dazu beitragen, dass der Maschinenprozess reibungslos und präzise abläuft und die Ausgabe des Systems optimiert wird.

Die Parameter, die in einem System oder Programm festgelegt wurden, können auf allen Ebenen darunter verwendet werden. Wenn Sie sich beispielsweise eine Baumstruktur vorstellen, können die Parameter, die auf der obersten Ebene festgelegt wurden, auf allen Ebenen darunter verwendet werden. Wenn Sie jedoch versuchen, diese Parameter auf einer höheren Ebene als der ursprünglichen Ebene zu verwenden, wird dies nicht funktionieren. Ein Parameter ist eine bestimmte Einstellung, die in einem System oder Programm vorgenommen wurde. Diese Einstellungen können verschiedene Eigenschaften haben, die bearbeitet werden können. Diese Eigenschaften können beispielsweise Datentyp, Wertebereich oder Standardwert sein. Zusammenfassend kann man sagen, dass Parameter in einem System oder Programm festgelegt werden, um Einstellungen zu definieren, die von verschiedenen Ebenen im System oder Programm verwendet werden können. Jeder Parameter hat verschiedene Eigenschaften, die bearbeitet werden können, um seine Funktionalität zu ändern.

Je nachdem, in welcher Ebene die Parameter deklariert wurde, entsteht im PLC-Code und in der HMI ein entsprechender Eintrag. Dies kann entweder auf der Plant-, Hardwarezonen- oder Sequence-Ebene erfolgen.



HMI

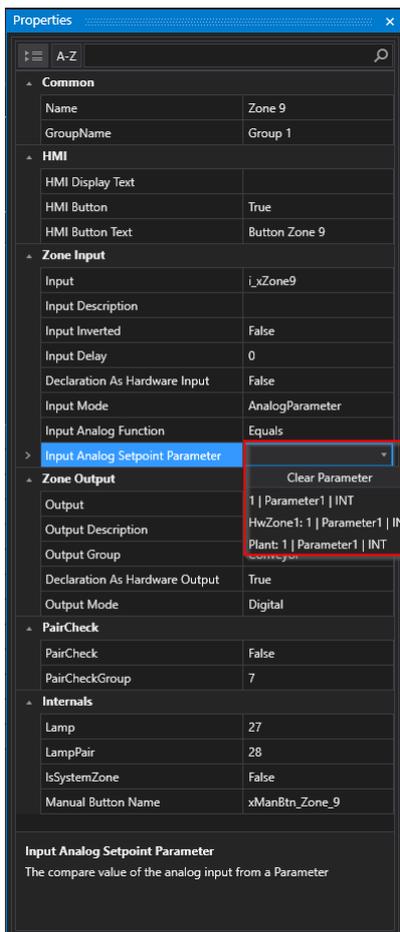
1. Plant Ebene
2. Hardwarezone Ebene
3. Sequence Ebene

Die Parameter werden mithilfe eines Editors bearbeitet und erstellt. Dabei werden verschiedene Optionen zur Verfügung gestellt, die im Folgenden erläutert werden:

Name

Der Name des Parameters dient als Grundlage für den Variablennamen. Die Notation für den Namen des Parameters ist frei wählbar, es empfiehlt sich jedoch, einen eindeutigen und beschreibenden Namen zu verwenden.

Wenn der Parameter als Eingabeparameter für eine Zone verwendet wird, wird er anhand der Ebene und des Parameternamens ausgewählt.

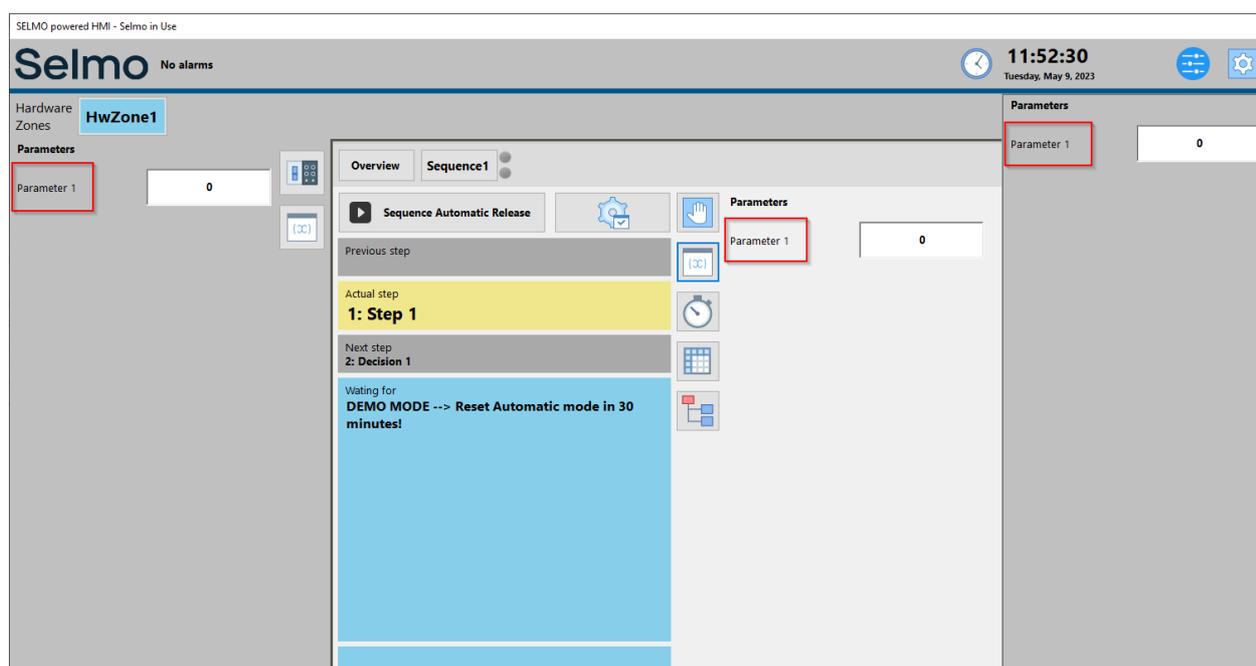


Studio

HMI Display Text

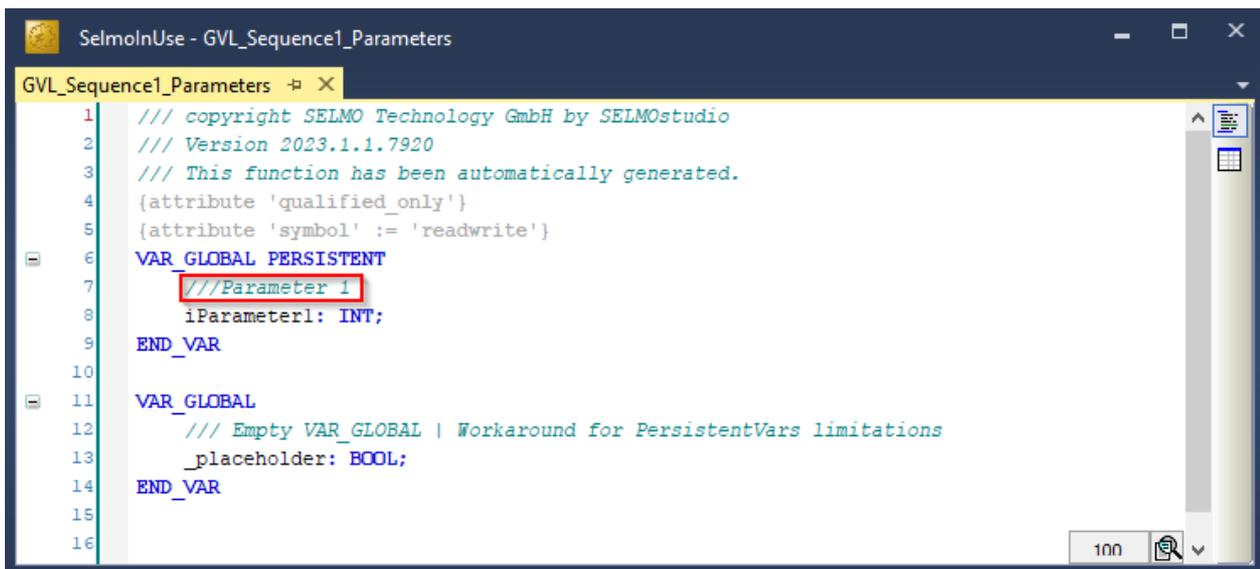
Der Anzeigename des Parameters in der HMI bezieht sich auf den Namen des Parameters, der auf dem HMI-Display des Systems oder der Maschine angezeigt wird. Dabei gibt es keine spezielle Notation für den Anzeigename, allerdings ist es wichtig, dass der Name des Parameters für den Benutzer verständlich und leicht lesbar ist.

Aus Gründen der Lesbarkeit sollte der Umfang des HMI-Displaytexts jedoch minimal gehalten werden. Das bedeutet, dass der Anzeigename des Parameters so kurz wie möglich sein sollte, aber dennoch ausreichend aussagekräftig, um vom Benutzer schnell erkannt und verstanden werden zu können.



HMI

Zusätzlich dient der Anzeigename des Parameters in der HMI auch als Kommentar für die zugehörige Variable des Parameters im PLC-Code.



```
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  {attribute 'qualified_only'}
5  {attribute 'symbol' := 'readwrite'}
6  VAR GLOBAL PERSISTENT
7  ///Parameter 1
8  iParameter1: INT;
9  END_VAR
10
11 VAR_GLOBAL
12   /// Empty VAR_GLOBAL | Workaround for PersistentVars limitations
13   _placeholder: BOOL;
14 END_VAR
15
16
```

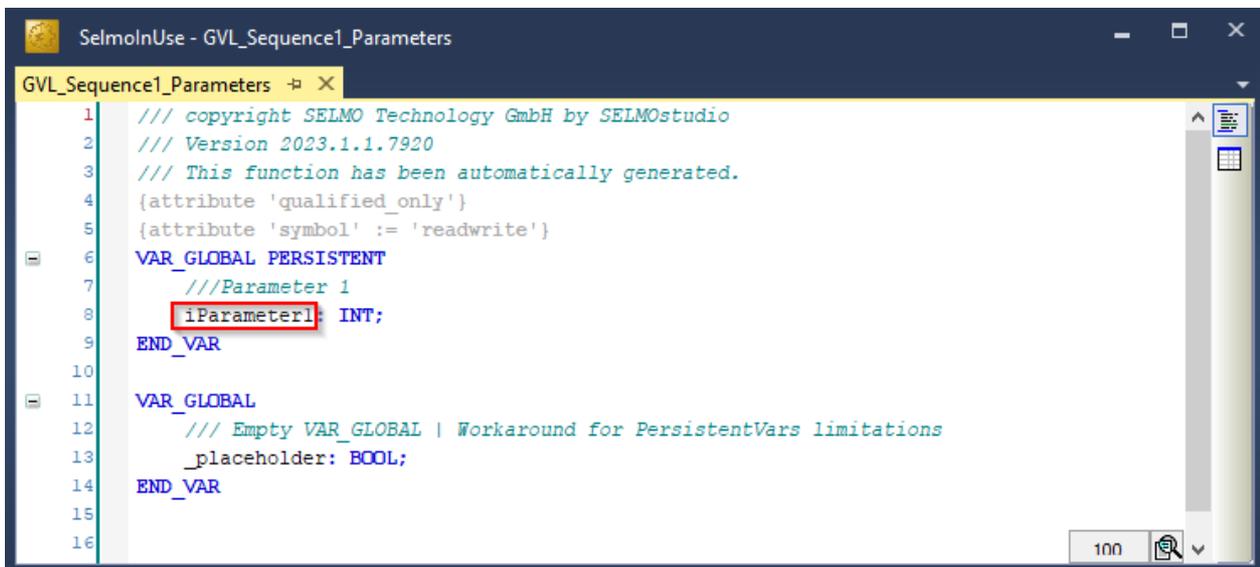
PLC

Variable Name

Der Variable Name leitet sich vom Namen des Parameters automatisch ab und ist nicht editierbar. Der Name einer Variablen ist wichtig für die Lesbarkeit und Verständlichkeit des Codes. Eine gute Variable sollte eindeutig, aussagekräftig und funktionell sein. Das heißt, sie sollte nur eine Bedeutung haben, den Inhalt oder Zweck der Variable beschreiben und mit dem Datentyp und der Logik des Programms übereinstimmen.

Hinweis:

Selmo verwendet die PLCopen Coding Guidelines. Dies ist ein international anerkannter Standard für die Programmierung von Automatisierungssystemen, insbesondere für die Programmierung von speicherprogrammierbaren Steuerungen (SPS). Die PLCopen Coding Guidelines legen Regeln und Best Practices fest, die den Entwicklern helfen, sicherzustellen, dass ihre SPS-Programme lesbar, wiederverwendbar und robust sind. Die Guidelines umfassen eine Vielzahl von Themen, einschließlich Namensgebung von Variablen, Kommentierung von Code, Programmstruktur und Fehlerbehandlung. Die Namensgebung von Variablen in der PLCopen Coding Guidelines folgt bestimmten Regeln. So sollen beispielsweise Variablennamen aussagekräftig sein und den Zweck und den Typ der Variable widerspiegeln. Der Name sollte in englischer Sprache verfasst sein und sich an bestimmte Konventionen halten, wie zum Beispiel die Verwendung von CamelCase. Weiterhin werden Regeln für die Benennung von Eingangs- und Ausgangsvariablen, temporären Variablen und Konstanten festgelegt. Insgesamt zielt die PLCopen Coding Guideline darauf ab, die Lesbarkeit, Wartbarkeit und Robustheit von SPS-Programmen zu verbessern und somit zu einer höheren Effizienz und Produktivität bei der Entwicklung von Automatisierungssystemen beizutragen.



```
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  {attribute 'qualified_only'}
5  {attribute 'symbol' := 'readwrite'}
6  VAR_GLOBAL PERSISTENT
7  ///Parameter 1
8  iParameter1: INT;
9  END_VAR
10
11 VAR_GLOBAL
12  /// Empty VAR_GLOBAL | Workaround for PersistentVars limitations
13  _placeholder: BOOL;
14  END_VAR
15
16
```

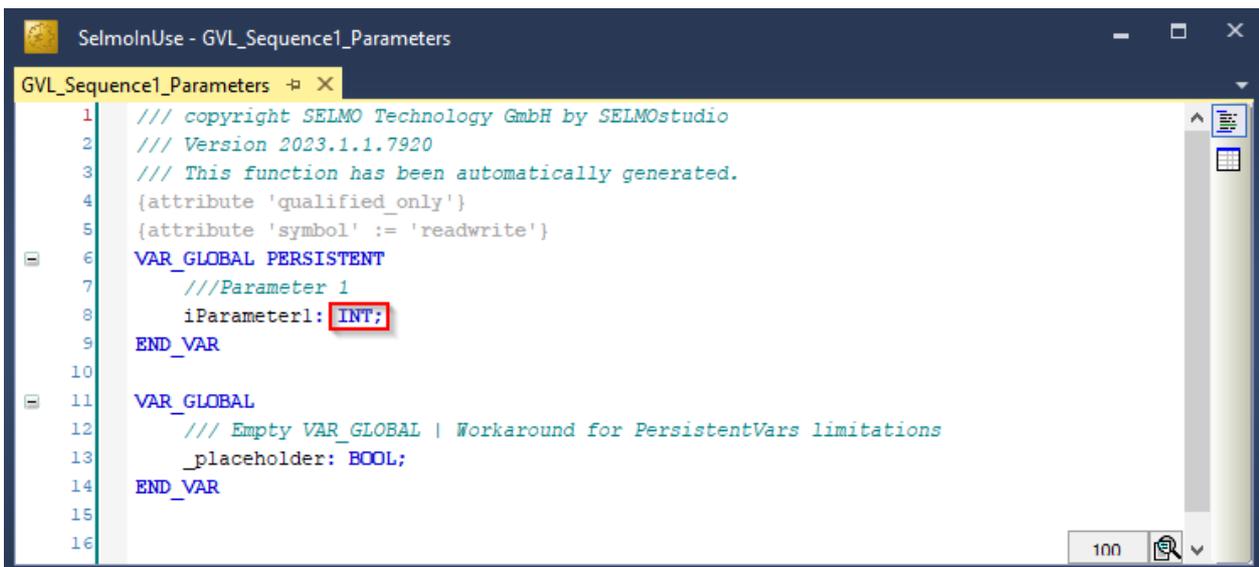
PLC

Type

Der Parameter-Typ gibt an, ob es sich um einen Eingangsparameter (Input) oder Ausgangsparameter (Output) handelt. Ein Eingangsparameter wird verwendet, um Werte in das System einzugeben oder bestimmte Einstellungen vorzunehmen, während ein Ausgangsparameter verwendet wird, um Werte aus dem System auszugeben oder bestimmte Informationen anzuzeigen. Ein Beispiel für einen Eingangsparameter könnte ein Parameter sein, der die gewünschte Position eines Motors oder eine bestimmte Betriebsart definiert. Der Benutzer gibt den Wert für diesen Parameter ein, der dann vom System verwendet wird. Ein Beispiel für einen Ausgangsparameter könnte ein Parameter sein, der die Ist-Position eines Wegmesssystems anzeigt oder eine Warnmeldung ausgibt, wenn ein bestimmter Zustand erreicht ist. Zusammenfassend gibt der Parameter-Typ an, ob der Parameter zur Eingabe oder zur Ausgabe von Werten dient, um das System oder die Maschine zu steuern und zu überwachen.

PLC Data Type

Die Deklaration des Variablentyps erfolgt standardmäßig als Integer-Datentyp. Über das Dropdown-Menü kann jedoch der entsprechende Datentyp ausgewählt werden. Die verfügbaren Datentypen entsprechen der Liste der Standarddatentypen.



```
1  /// copyright SELMO Technology GmbH by SELMOstudio
2  /// Version 2023.1.1.7920
3  /// This function has been automatically generated.
4  {attribute 'qualified_only'}
5  {attribute 'symbol' := 'readwrite'}
6  VAR_GLOBAL PERSISTENT
7  ///Parameter 1
8  iParameter1: INT;
9  END_VAR
10
11 VAR_GLOBAL
12  /// Empty VAR_GLOBAL | Workaround for PersistentVars limitations
13  _placeholder: BOOL;
14  END_VAR
15
16
```

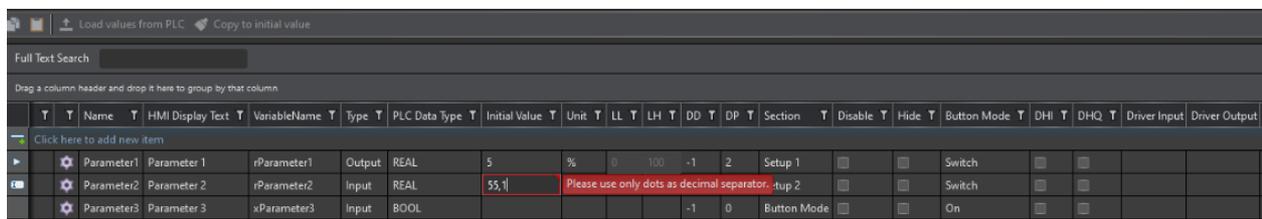
PLC

Initial Value

Der Initialwert für eine Variable gibt den anfänglichen Wert an, der der Variable zugewiesen wird, wenn das Programm gestartet wird oder der Parameter neu initialisiert wird. Je nach PLC-Datentyp sind nur bestimmte Eingabeformate für den Initialwert zulässig. Das bedeutet, dass der Initialwert für eine Variable in einem bestimmten Format eingegeben werden muss, das mit dem Datentyp der Variable übereinstimmt.

Hinweis:

Wenn der Initialwert in einem falschen Format eingegeben wird, kann dies zu Fehlfunktionen oder Fehlern im Programm führen. Es ist daher wichtig, sicherzustellen, dass der Initialwert für eine Variable im richtigen Format eingegeben wird, um ein reibungsloses Funktionieren des Programms zu gewährleisten. Dies wird bereits im Vorfeld geprüft und entsprechend angezeigt.

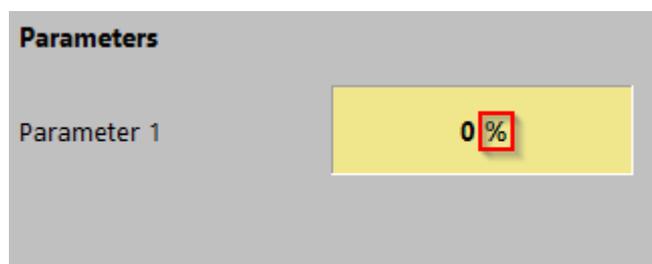


T	Name	HMI Display Text	VariableName	Type	PLC Data Type	Initial Value	Unit	LL	LH	DD	DP	Section	Disable	Hide	Button Mode	DHI	DHQ	Driver Input	Driver Output
	Parameter1	Parameter 1	rParameter1	Output	REAL	5	%	0	100	-1	2	Setup 1	<input type="checkbox"/>	<input type="checkbox"/>	Switch	<input type="checkbox"/>	<input type="checkbox"/>		
	Parameter2	Parameter 2	rParameter2	Input	REAL	55,1						Setup 2	<input type="checkbox"/>	<input type="checkbox"/>	Switch	<input type="checkbox"/>	<input type="checkbox"/>		
	Parameter3	Parameter 3	xParameter3	Input	BOOL					-1	0	Button Mode	<input type="checkbox"/>	<input type="checkbox"/>	On	<input type="checkbox"/>	<input type="checkbox"/>		

Studio

Unit

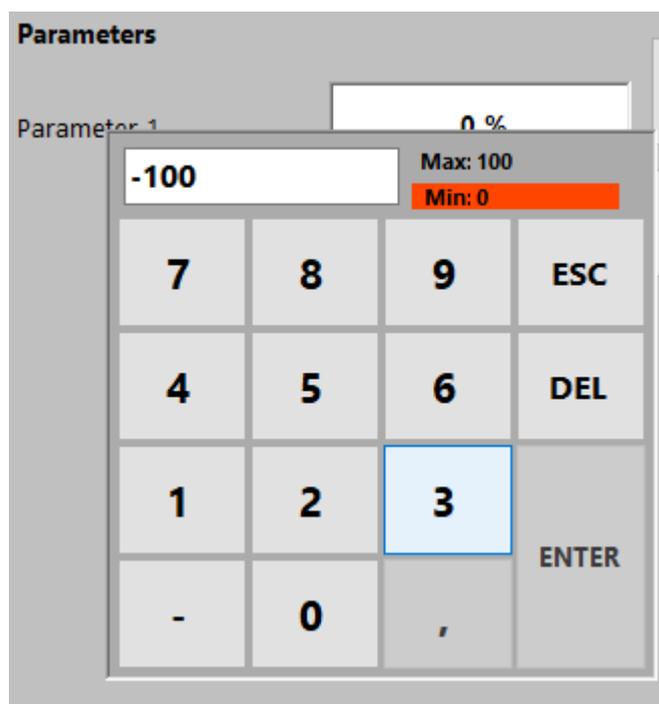
Die Anzeige-Einheit des Parameters in der HMI bezieht sich auf die Einheit, in der der Parameter auf dem HMI-Display des Systems oder der Maschine angezeigt wird. Die Einheit kann je nach PLC-Datentyp variieren, jedoch sind alle Einträge gültig und werden nicht auf ihre Richtigkeit oder Konsistenz mit anderen Parametern oder dem System überprüft. Es ist jedoch wichtig, dass die Einheit für den Benutzer verständlich ist und ihm dabei hilft, den Parameter richtig zu interpretieren. Eine korrekte und eindeutige Anzeige-Einheit kann dazu beitragen, Missverständnisse zu vermeiden und die Bedienung des Systems zu erleichtern. Daher sollte die Anzeige-Einheit des Parameters sorgfältig gewählt und richtig dokumentiert werden, um ein korrektes Verständnis und eine fehlerfreie Verwendung des Systems oder der Maschine zu gewährleisten.



HMI

LL (Limit Low)

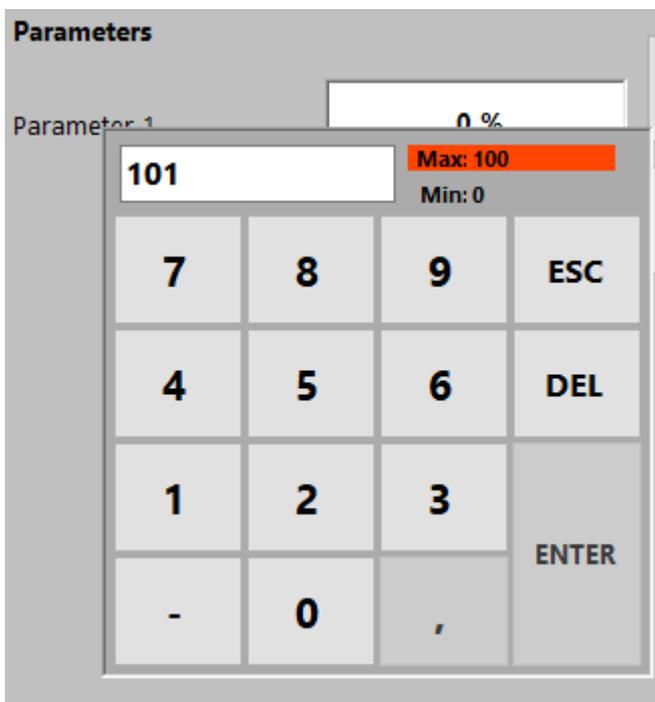
Der minimale Eingabewert für einen Input-Parameter ist der kleinste Wert, der für diesen Parameter akzeptabel ist. Dieser Wert kann je nach PLC-Datentyp variieren, und es sind nur bestimmte Eingabeformate zulässig, die mit dem Datentyp des Parameters übereinstimmen. Es ist wichtig sicherzustellen, dass der minimale Eingabewert im richtigen Format eingegeben wird, da ansonsten Fehlfunktionen oder Fehler im Programm auftreten können. Eine fehlerhafte Eingabe kann beispielsweise zu einer Überlastung des Systems oder einer unerwarteten Reaktion führen. Daher sollte der minimale Eingabewert sorgfältig dokumentiert und eingegeben werden, um die korrekte Funktionsweise des Systems oder der Maschine zu gewährleisten. Wenn ein Wert für einen Parameter eingegeben wird, der unter dem definierten unteren Grenzwert (LL) liegt, wird dieser Grenzwert in der Regel rot markiert, um anzuzeigen, dass die Eingabe nicht akzeptiert wird. Das System oder die Maschine verweigert die Eingabe des Wertes, um zu verhindern, dass das System fehlerhaft oder instabil wird. Die rote Markierung soll dem Benutzer auf den Fehler hinweisen und ihm helfen, den Eingabewert entsprechend zu korrigieren.



HMI

LH (Limit High)

Der maximale Eingabewert für einen Parameter ist der höchste Wert, der für diesen Parameter akzeptabel ist. Wenn ein Wert für einen Parameter eingegeben wird, der über dem definierten oberen Grenzwert (LH) liegt, wird dieser Grenzwert in der Regel rot markiert, um anzuzeigen, dass die Eingabe nicht akzeptiert wird. Ähnlich wie bei einem zu niedrigen Eingabewert, verweigert das System die Eingabe des Wertes, um zu verhindern, dass das System fehlerhaft oder instabil wird. Der obere Grenzwert ist wichtig, um sicherzustellen, dass die Maschine oder das System innerhalb sicherer und effizienter Betriebsparameter arbeitet. Eine korrekte Eingabe des Parameters ist wichtig, um die Leistung und Sicherheit des Systems zu gewährleisten. Daher sollte der obere Grenzwert sorgfältig dokumentiert und eingehalten werden, um zu verhindern, dass das System fehlerhaft oder instabil wird.

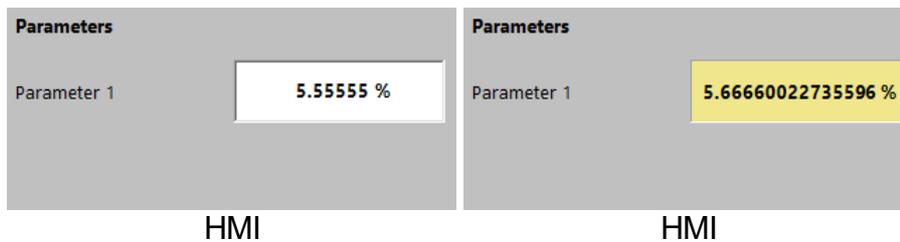


HMI

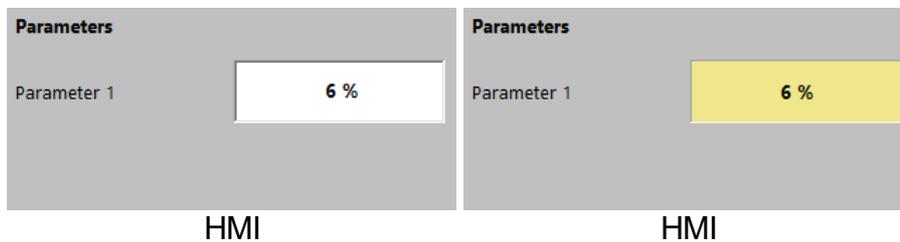
DD (Decimal Digits)

DD (Decimal Digits) steht für die Anzahl der Dezimalstellen, die bei der Anzeige des Parameterwerts berücksichtigt werden sollen. Diese Anzahl wird üblicherweise im Parameter-Setup definiert und kann je nach Anwendung variieren. Wenn der Wert von DD auf 0 gesetzt wird, bedeutet dies, dass keine Nachkommastellen angezeigt werden sollen und der Wert als Ganzzahl dargestellt wird. Wenn DD auf -1 gesetzt wird, werden keine Nachkommastellen berücksichtigt, wenn DD auf 1 gesetzt wird, wird eine Dezimalstelle berücksichtigt und so weiter. Die Anzahl der Dezimalstellen ist wichtig, um eine korrekte und genaue Anzeige des Parameterwerts sicherzustellen. Wenn die Anzahl der Dezimalstellen nicht ausreichend ist, können wichtige Informationen verloren gehen oder ungenau dargestellt werden. Wenn die Anzahl der Dezimalstellen zu hoch ist, kann dies die Lesbarkeit des Wertes beeinträchtigen. Daher ist es wichtig, die Anzahl der Dezimalstellen sorgfältig zu definieren und zu überwachen, um eine genaue und lesbare Darstellung des Parameterwerts zu gewährleisten.

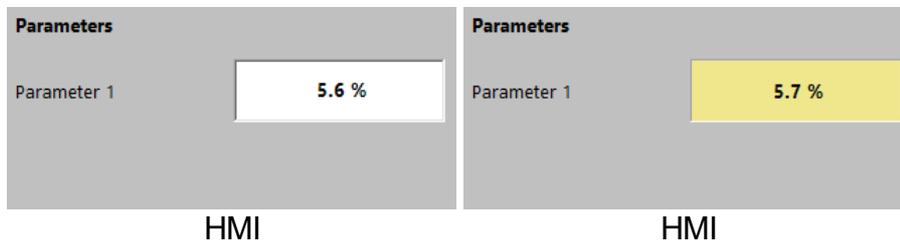
DD -1 (Input/Output)



DD 0 (Input/Output)



DD 1 (Input/Output)

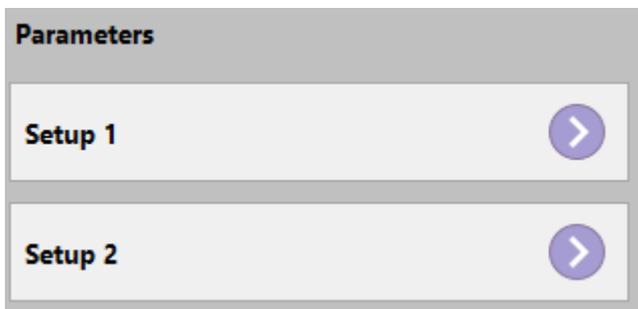


DP (Decimal Place)

DP (Decimal Place) bezieht sich auf die Position der Kommastelle bei der Anzeige des Parameterwerts. Dieser Wert wird üblicherweise im Parameter-Setup definiert und gibt an, an welcher Stelle der Parameterwert aufgeteilt werden soll, um die Dezimalstelle zu kennzeichnen. Wenn beispielsweise der Wert von DP auf 2 gesetzt wird, bedeutet dies, dass der Parameterwert an der zweiten Stelle von rechts geteilt wird, um die Dezimalstelle zu kennzeichnen. Der Wert 12345 würde beispielsweise als "123.45" dargestellt werden. Die Position der Kommastelle ist wichtig, um sicherzustellen, dass der Parameterwert korrekt und verständlich dargestellt wird. Wenn die Kommastelle an der falschen Stelle platziert wird, kann dies zu Verwirrung oder ungenauen Informationen führen. Daher ist es wichtig, die Position der Kommastelle sorgfältig zu definieren und zu überwachen, um eine klare und genaue Darstellung des Parameterwerts zu gewährleisten.

Section

Die Eigenschaft "Section" kann verwendet werden, um Parameter zu gruppieren und sie in einem gemeinsamen Menüpunkt in der HMI darzustellen. Wenn Parameter mit der gleichen Section-Eigenschaft markiert sind, werden sie automatisch in der gleichen Gruppe zusammengefasst. Durch das Gruppieren von Parametern in Sektionen wird das Menü übersichtlicher und benutzerfreundlicher gestaltet, da verwandte Parameter zusammengefasst werden und der Benutzer sie leichter finden und ändern kann. Die Section-Eigenschaft kann beliebig benannt werden, um die Gruppe von Parametern entsprechend zu beschreiben. Ein Beispiel für eine sinnvolle Benennung einer Section-Eigenschaft ist "Kommunikationseinstellungen" für Parameter, die sich auf die Kommunikation des Systems beziehen. Durch die Verwendung von Sections kann die Handhabung von Parametern einfacher und effizienter gestaltet werden, was insgesamt zu einer verbesserten Benutzererfahrung führt.



HMI

Disable Input in Automatic

Die Funktion "Disable Input in Automatic" ermöglicht es, das Ändern von Eingabevariablen zu sperren, solange der Automatik-Modus aktiv ist. Dies bedeutet, dass Benutzer den Wert einer Eingabevariablen nicht manuell ändern können, solange das System im Automatik-Modus arbeitet. Diese Funktion ist besonders nützlich, um die Sicherheit und Integrität des Systems zu gewährleisten, da sie verhindert, dass Benutzer versehentlich den Betrieb des Systems beeinträchtigen, während es in einem automatisierten Betriebsmodus arbeitet. Wenn der Automatik-Modus deaktiviert ist, können Benutzer den Wert der Eingabevariablen wieder manuell ändern. Diese Funktion ist besonders hilfreich in industriellen Anwendungen, in denen es wichtig ist, dass das System sicher und zuverlässig funktioniert, auch wenn es von unterschiedlichen Benutzern betrieben wird.

Hide

Wenn das Feld "Hide" für einen Parameter gesetzt ist, wird dieser Parameter in der HMI nicht angezeigt. Das bedeutet, dass Benutzer den Parameter nicht sehen oder darauf zugreifen können, selbst wenn er im System vorhanden ist. Diese Funktion kann nützlich sein, wenn ein Parameter nur für interne Zwecke oder zur Konfiguration des Systems verwendet wird und Benutzer nicht wissen müssen, dass er existiert. Sie kann auch dazu beitragen, die Benutzeroberfläche übersichtlicher zu gestalten, indem sie unwichtige oder selten verwendete Parameter ausblendet. Es ist jedoch wichtig zu beachten, dass das Ausblenden eines Parameters nicht bedeutet, dass er nicht mehr im System vorhanden ist. Der Parameter kann immer noch von anderen Systemkomponenten verwendet werden, und Änderungen an seinem Wert können immer noch Auswirkungen auf das System haben.

Button mode

Boolesche Parameter können als Buttons angelegt werden. Es stehen verschiedene Modi zur Verfügung:

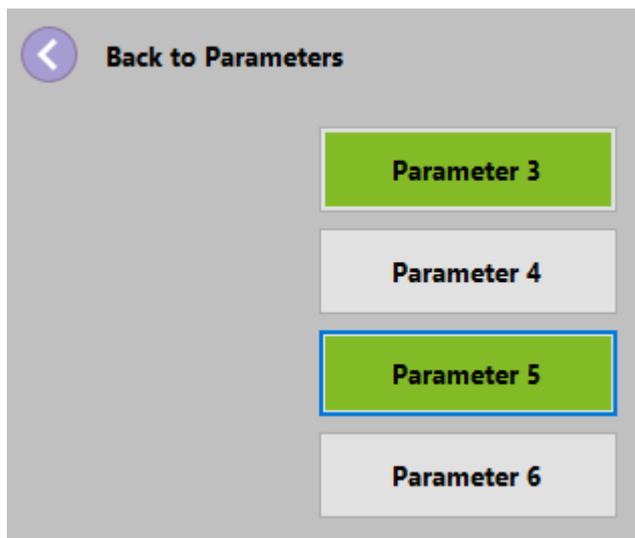
"On": Schaltet die Parametervariable auf "true" beim Betätigen des HMI-Buttons.

"Off": Schaltet die Parametervariable auf "false" beim Betätigen des HMI-Buttons.

"Switch": Schaltet die Parametervariable auf "true" beim Drücken und auf "false" beim Loslassen des HMI-Buttons.

"Toggle": Wechselt bei jedem Tastendruck den Zustand der Parametervariable zwischen "true" und "false".

Wenn der HMI-Button im grünen Zustand ist, entspricht dies dem Wert 'true' der Parametervariable. Wenn der Button im grauen Zustand ist, entspricht dies dem Wert 'false'.



HMI

DHI (Declaration as Hardware Input)

Boolescher Wert welcher nur bei Bekhoff Zielsystem/Twincat Zielsystem damit man diese direkt mit den IO's verknüpfen kann. Bei Adressendeklaration steht, wenn man es anhackt %I*.

DHQ (Declaration as Hardware Output)

Boolescher Wert welcher nur bei Bekhoff Zielsystem/Twincat Zielsystem damit man diese direkt mit den IO's verknüpfen kann. Bei Adressendeklaration steht, wenn man es anhackt %O*.

Driver Input

Informationen darüber, welche Parameter mit dem Treiber verknüpft sind. Wichtig ist der Zusammenhang mit analogen Ein- oder Ausgängen. In einigen Treibern, auf der Assembly-

Ebene, wenn ein analoger Eingang vorhanden ist, besteht die Möglichkeit, diesen Eingang direkt mit einem Parameter zu verknüpfen. Unter dem Feld 'Driver Input' kann man dann den Paramet, welcher ausgewählt wurde lesen.

Driver Output

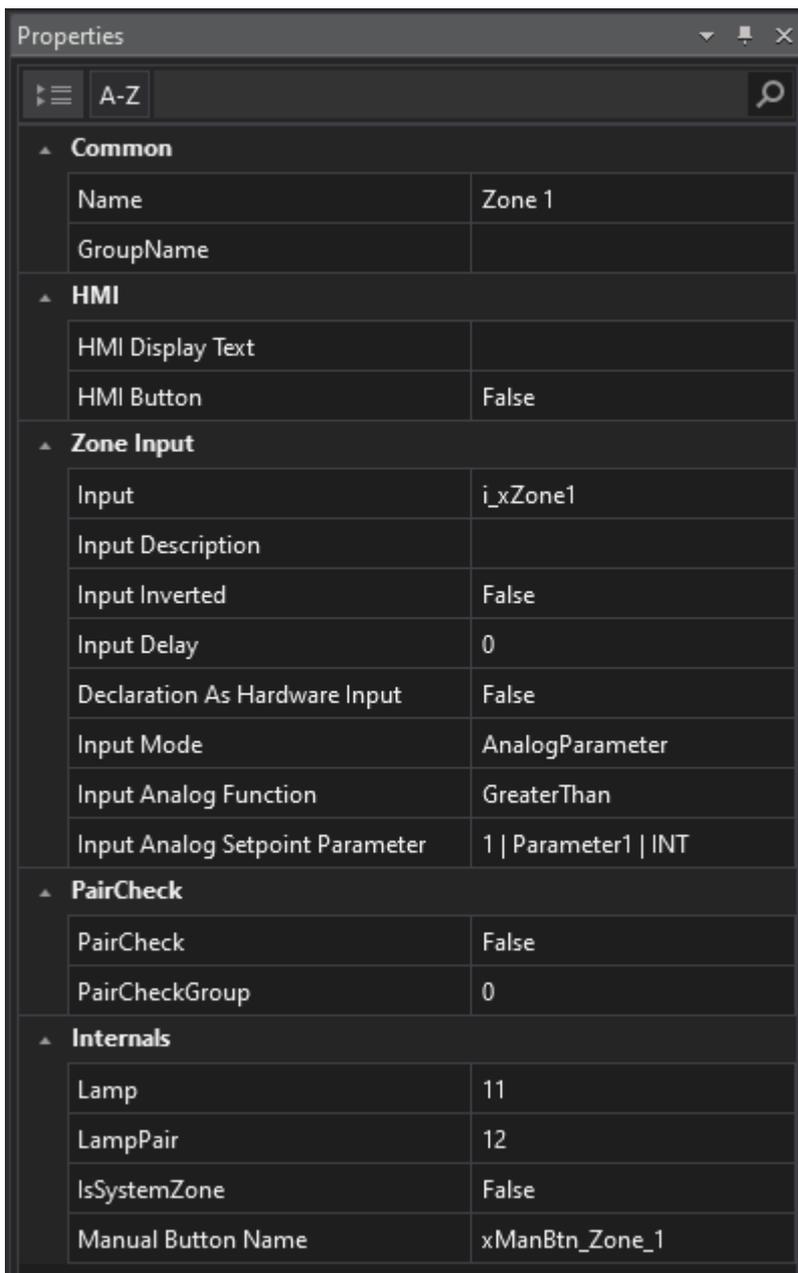
Informationen darüber, welche Parameter mit dem Treiber verknüpft sind. Wichtig ist der Zusammenhang mit analogen Ein- oder Ausgängen. In einigen Treibern, auf der Assembly-Ebene, wenn ein analoger Ausgang vorhanden ist, besteht die Möglichkeit, diesen Ausgang direkt mit einem Parameter zu verknüpfen. Unter dem Feld 'Driver Output' kann man dann den Paramet, welcher ausgewählt wurde lesen.

1.6.2 Szenarien wo können Parameter eingebunden werden

Die angelegten Parameter können zur Flexibilisierung des modellierten Prozesses verwendet werden. Nachfolgend sind die Verwendungsstellen erklärt und beschrieben.

Zone-In und Zone-InOut Inputs

Für Inputs von Zone-In und Zone-InOut kann der Input Mode Analoge Parameter aktiviert werden, danach wird der angelegte Parameter auf welchen verglichen werden soll ausgewählt. Der analoge Eingang wird nun mit dem Parameter über die ausgewählte Input Analog Funktion verglichen.



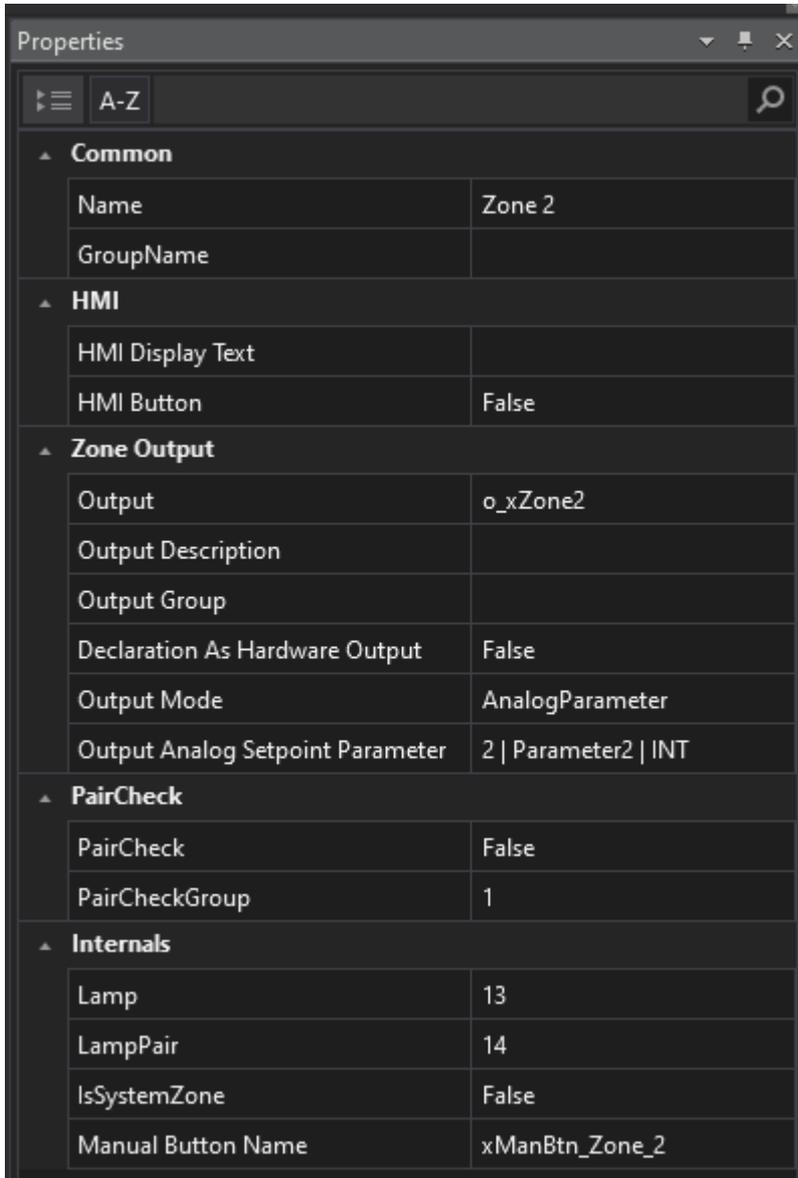
The screenshot shows the Properties window in Studio, displaying the configuration for a Zone Input. The window is titled 'Properties' and has a search bar and a sort button (A-Z). The properties are organized into several sections:

- Common**
 - Name: Zone 1
 - GroupName:
- HMI**
 - HMI Display Text:
 - HMI Button: False
- Zone Input**
 - Input: i_xZone1
 - Input Description:
 - Input Inverted: False
 - Input Delay: 0
 - Declaration As Hardware Input: False
 - Input Mode: AnalogParameter
 - Input Analog Function: GreaterThan
 - Input Analog Setpoint Parameter: 1 | Parameter1 | INT
- PairCheck**
 - PairCheck: False
 - PairCheckGroup: 0
- Internals**
 - Lamp: 11
 - LampPair: 12
 - IsSystemZone: False
 - Manual Button Name: xManBtn_Zone_1

Studio

Zone-InOut und Zone-Out Outputs

Für Outputs von Zone-InOut und Zone-Out kann der Output Mode: AnalogeParameter aktiviert werden, danach wird der angelegte Parameter, mit welchem der Analoge Ausgang beschalten werden soll ausgewählt. Wird der Output der Zone aus der Sequence geschalten nimmt er den Wert des Parameters an.



The screenshot shows the Properties window in Studio, displaying the configuration for a Zone Output. The window is titled 'Properties' and has a search bar at the top. The configuration is organized into several sections:

- Common**
 - Name: Zone 2
 - GroupName:
- HMI**
 - HMI Display Text:
 - HMI Button: False
- Zone Output**
 - Output: o_xZone2
 - Output Description:
 - Output Group:
 - Declaration As Hardware Output: False
 - Output Mode: AnalogParameter
 - Output Analog Setpoint Parameter: 2 | Parameter2 | INT
- PairCheck**
 - PairCheck: False
 - PairCheckGroup: 1
- Internals**
 - Lamp: 13
 - LampPair: 14
 - IsSystemZone: False
 - Manual Button Name: xManBtn_Zone_2

Studio

Time Step

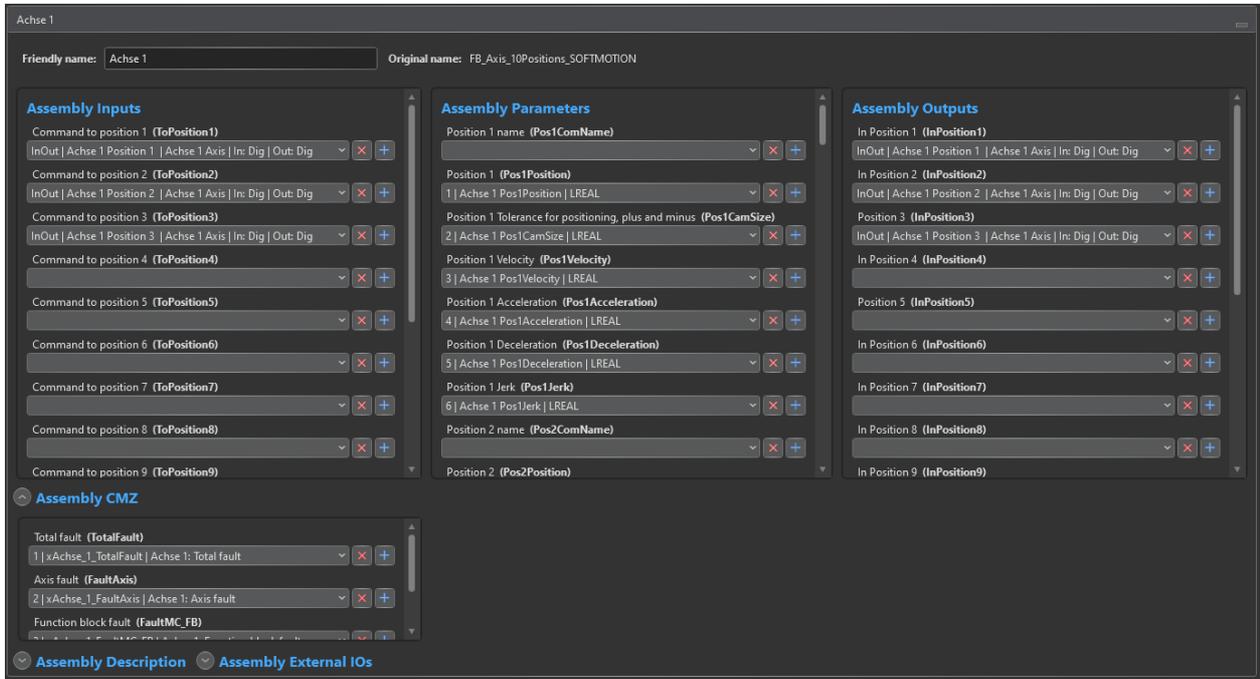
In Time Steps muss für die Festlegung der Schrittzeit ein Parameter mit Datentyp Time angelegt und schließlich im Time Step ausgewählt werden. Es gibt die Möglichkeit den Parameter vom Bediener verändern zu lassen oder auch die Möglichkeit den Parameter als fixe Schrittzeit zu definieren, hierfür wird der Parameter in der HMI ausgeblendet.

▲ Monitoring	
Disable Timeout	False
Timeout	300
Timeout Additional	5
▲ Step Control	
End Of Cycle	False
▲ Common	
Group Name	Path 2
ID	6
Name	Step 6
New ID	7
▲ HMI	
HMI Display Text	
Show Indication	False
▲ System Zones	
Show System Zones	True
▲ Logic Layer	
Start Shape	False
▲ Timer	
Timer Parameter	
Timer Value	5

Studio

Treiber

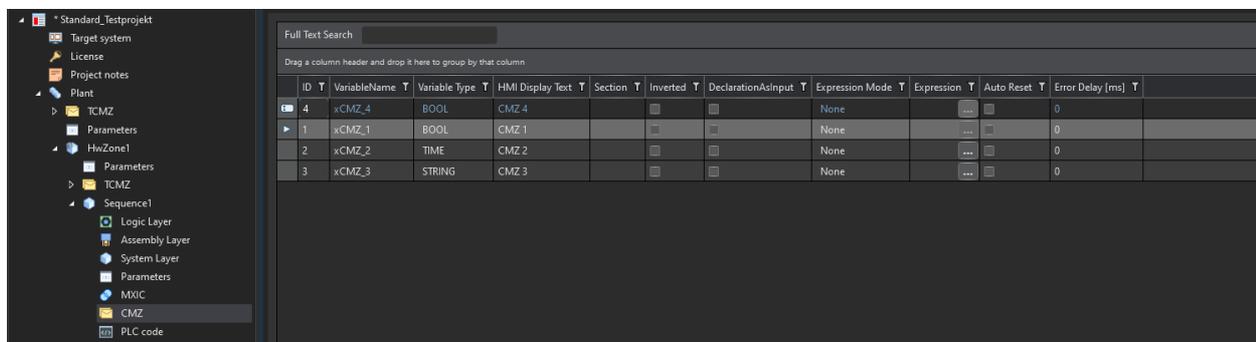
Parameter können auch mit Treiber-Bausteinen verknüpft werden oder werden von Treibern direkt angelegt. Nachfolgend ist das Beispiel eines Achstreibers zusehen, es können unterschiedlichste Treiber mit verschiedenen Parametern bedient werden, z.B.: Regeltreiber, Druckertreiber, FU-Treiber



Studio

CMZ

Alarmlevel können über den Expression Mode der CMZ mittels Parameter variabel eingegeben werden.



ID	VariableName	Variable Type	HMI Display Text	Section	Inverted	DeclarationAsInput	Expression Mode	Expression	Auto Reset	Error Delay (ms)
4	xCMZ_4	BOOL	CMZ 4		<input type="checkbox"/>	<input type="checkbox"/>	None	...	<input type="checkbox"/>	0
1	xCMZ_1	BOOL	CMZ 1		<input type="checkbox"/>	<input type="checkbox"/>	None	...	<input type="checkbox"/>	0
2	xCMZ_2	TIME	CMZ 2		<input type="checkbox"/>	<input type="checkbox"/>	None	...	<input type="checkbox"/>	0
3	xCMZ_3	STRING	CMZ 3		<input type="checkbox"/>	<input type="checkbox"/>	None	...	<input type="checkbox"/>	0

Studio

1.7 Finalisierung

Die Finalisierung der Selmo Solution hat folgende Schritte umfasst:

- Zonen Handbetrieb wurde definiert: Dies ermöglicht eine einfache und sichere Bedienung der Maschine in jedem Zustand.
- Mit MXIC (Manual Cross Interlock Check) wurden die Zonen untereinander verriegelt: Dies verhindert unerwünschte Zustandsübergänge und erhöht die Zuverlässigkeit der Software.
- Flexibilisierung durch Parameter Layer geschaffen: Dies erlaubt eine schnelle und einfache Anpassung der Maschine an unterschiedliche Anforderungen und Bedingungen.

Endergebnis:

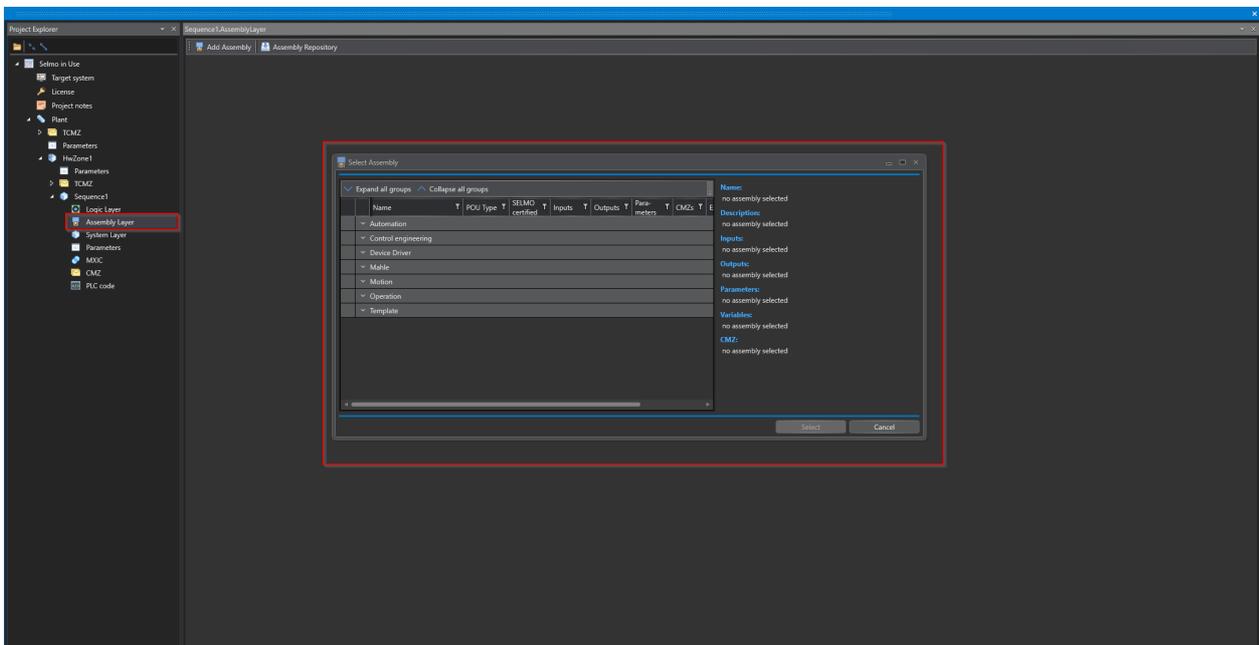
- Hohe Qualität und Stabilität der Software durch Standardisierung und durch Low Code Engineering: Die Software wurde automatisch aus dem digitalen Modell des Prozesses generiert, ohne manuellen Programmieraufwand oder Fehlerquellen.

For software that never lets you down!

Die Software ist jederzeit digital überprüfbar und kann bei Bedarf wiederverwendet oder geändert werden. Die Kommunikation zwischen Maschinenanwender, Konstrukteur und SPS-Programmierer ist transparent und verständlich.

1.8 Assembly/driver layer

Ein Assembly Layer ist eine Schicht in der Softwareentwicklung, die vorgefertigte Funktionsbausteine oder -komponenten enthält, die von Entwicklern genutzt werden können, um Softwareanwendungen zu erstellen. Es handelt sich dabei um eine Art Bibliothek, die den Entwicklungsprozess beschleunigt, indem sie bereits vorhandene Lösungen für häufig auftretende Probleme bereitstellt. Der Assembly Layer ist in der Regel eine abstrakte Schicht, die zwischen der eigentlichen Anwendungslogik und der zugrundeliegenden Hardware liegt. Er stellt eine Schnittstelle zwischen der Anwendung und der Hardware bereit und ermöglicht es Entwicklern, die zugrundeliegende Hardware durch die Verwendung von vorgefertigten Bausteinen zu abstrahieren. Der Einsatz eines Assembly Layers kann dazu beitragen, dass Entwickler weniger Zeit und Energie für die Entwicklung von Grundfunktionalitäten aufwenden müssen, und sich stattdessen auf die spezifischen Anforderungen der Anwendung konzentrieren können.

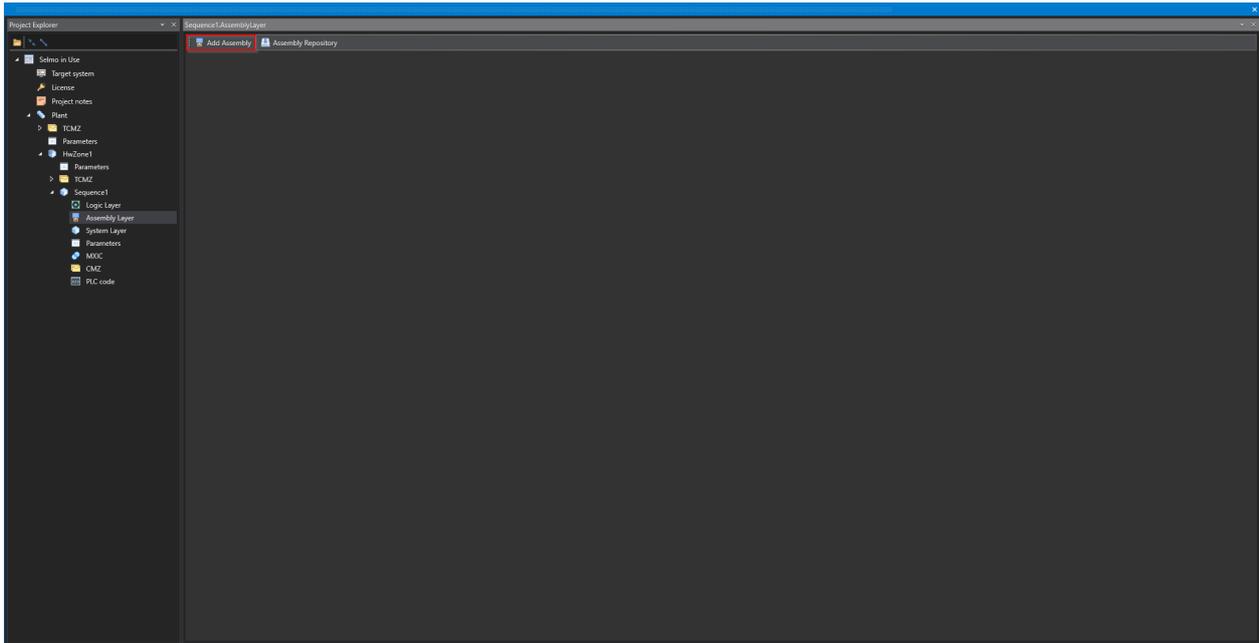


Studio

Treiber hinzufügen

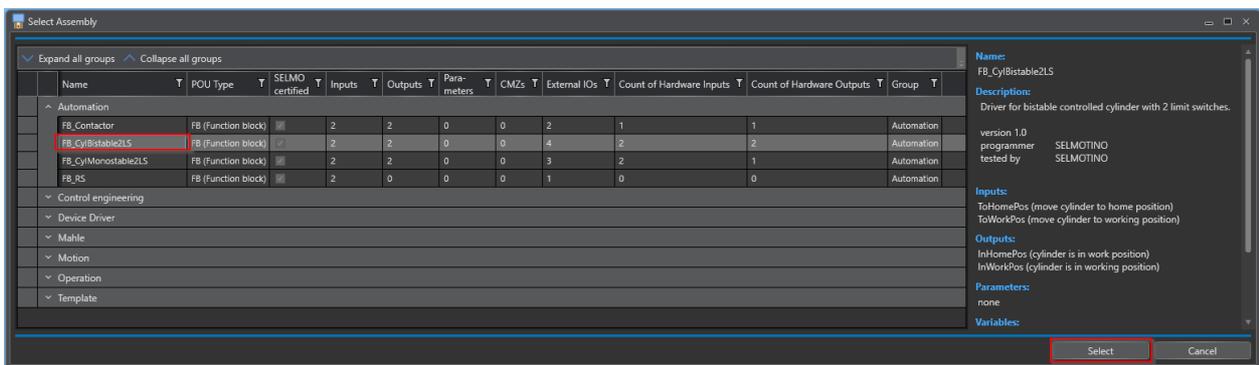
Ein Treiber ist eine Softwarekomponente, die es ermöglicht, bestimmte Geräte oder Funktionen anzusteuern oder zu verwalten. Das Hinzufügen von weiteren Treibern kann notwendig sein, um weitere Geräte oder Funktionen in die Anwendung zu integrieren oder um die bestehende Funktionalität zu erweitern.

Durch das Klicken auf die Schaltfläche 'Add Assembly' können in dieser Sequenz weitere Treiber hinzugefügt werden.



Studio

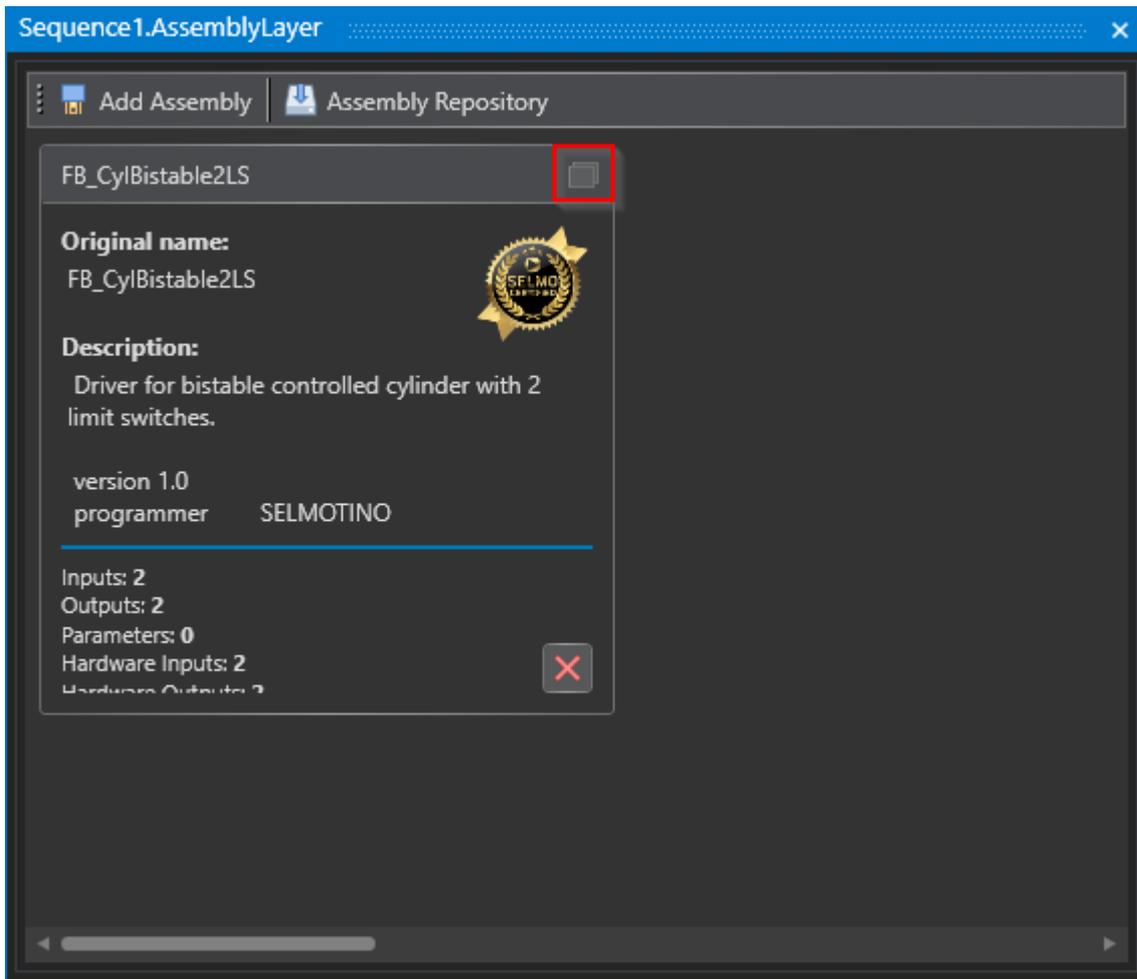
Es wird ein Dialogfeld geöffnet, in dem der Benutzer aus einer Liste von verfügbaren Treibern wählen kann. Sobald ein Treiber ausgewählt und hinzugefügt wurde, kann der Benutzer die Einstellungen für diesen Treiber konfigurieren, um ihn an die Anforderungen der Anwendung anzupassen.



Studio

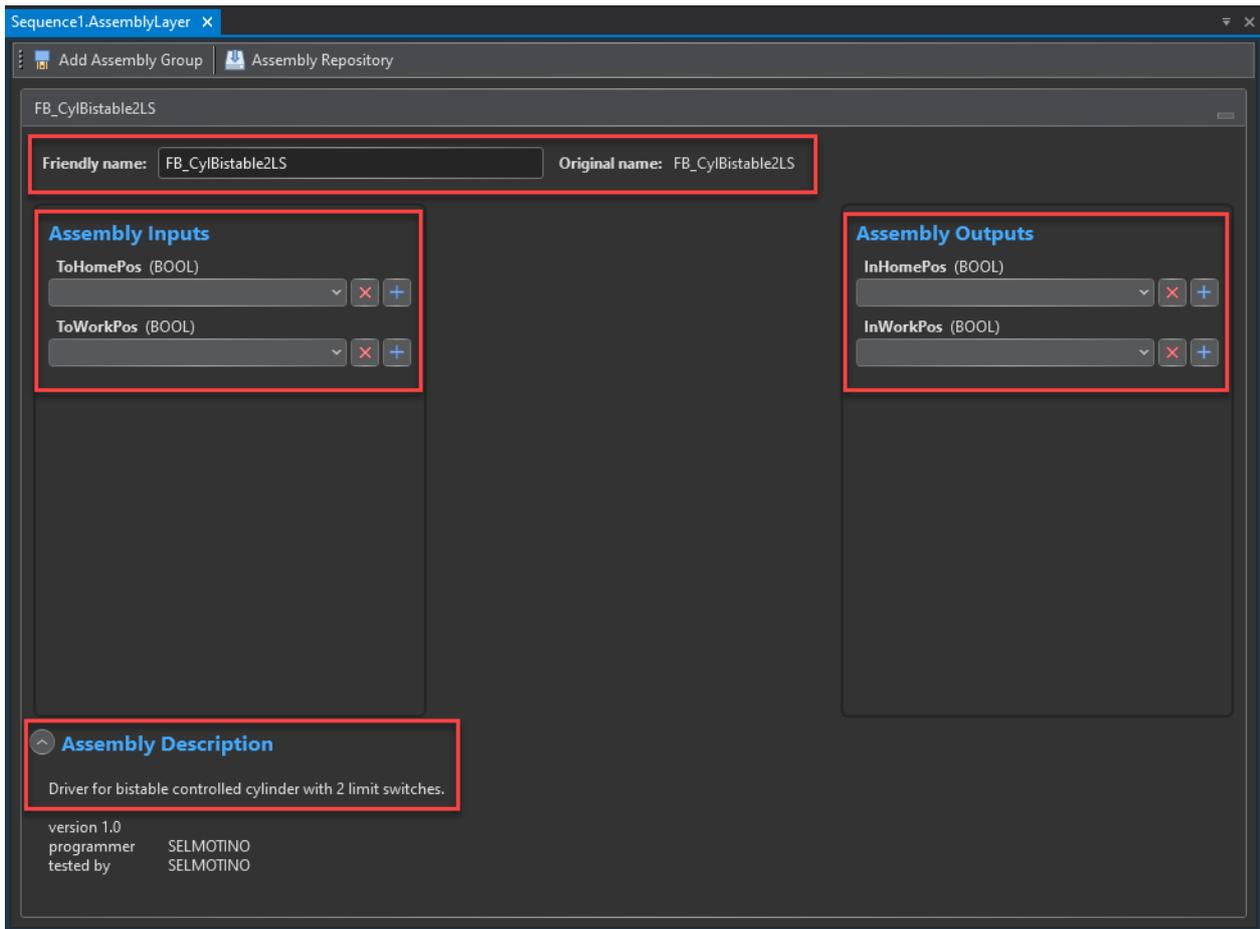
Die Assemblies, bzw. Treiber, finden sich in Gruppen unterteilt, durch auswählen eines Treiber und klicken auf Select, kann dieser der Sequence hinzugefügt werden.

Zonen verknüpfen



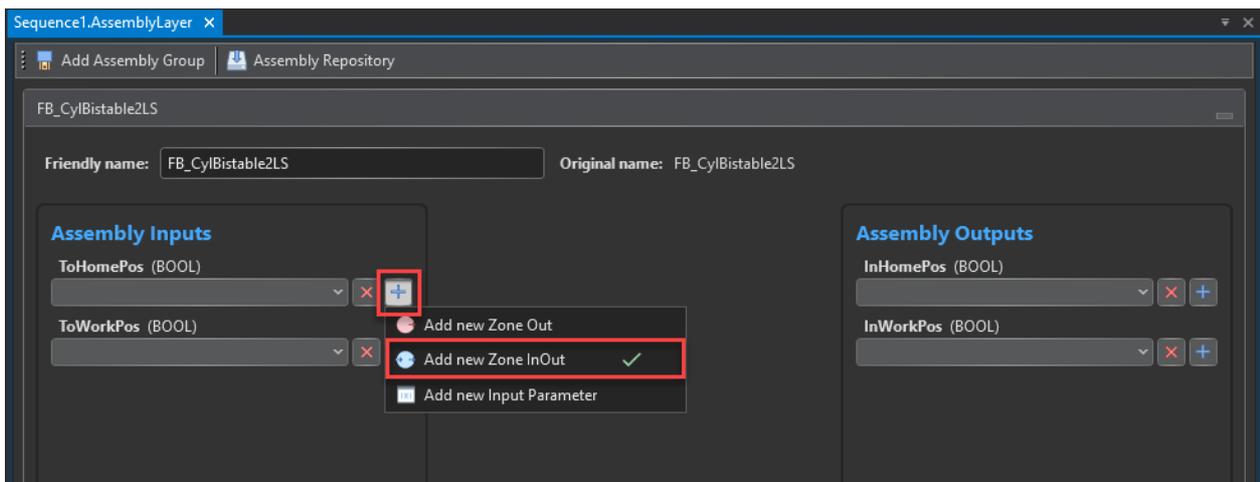
Studio

Wenn Sie auf die Schaltfläche "Maximize" in der rechten oberen Ecke des Treibers klicken, öffnen sich die Optionen für diesen Treiber. Dort können Sie einen Namen für die Baugruppe wählen und die Ein- und Ausgänge erstellen oder verknüpfen. Am unteren Rand des Fensters finden Sie außerdem eine kurze Beschreibung des Treibers, die Ihnen dabei helfen kann, die Funktionen und Eigenschaften des Treibers besser zu verstehen. Durch das Öffnen der Optionen können Sie die Konfiguration des Treibers anpassen, um sicherzustellen, dass er richtig in Ihre Anwendung integriert wird und die gewünschte Funktionalität bereitstellt. Die Möglichkeit, Ein- und Ausgänge zu erstellen oder zu verknüpfen, ist besonders wichtig, da dies es Ihnen ermöglicht, Daten von anderen Komponenten Ihrer Anwendung zu erhalten oder an diese zu senden. Insgesamt bietet die Schaltfläche "Maximize" und die darunter liegenden Optionen eine benutzerfreundliche Möglichkeit, um Treiber in Ihre Anwendung zu integrieren und anzupassen.



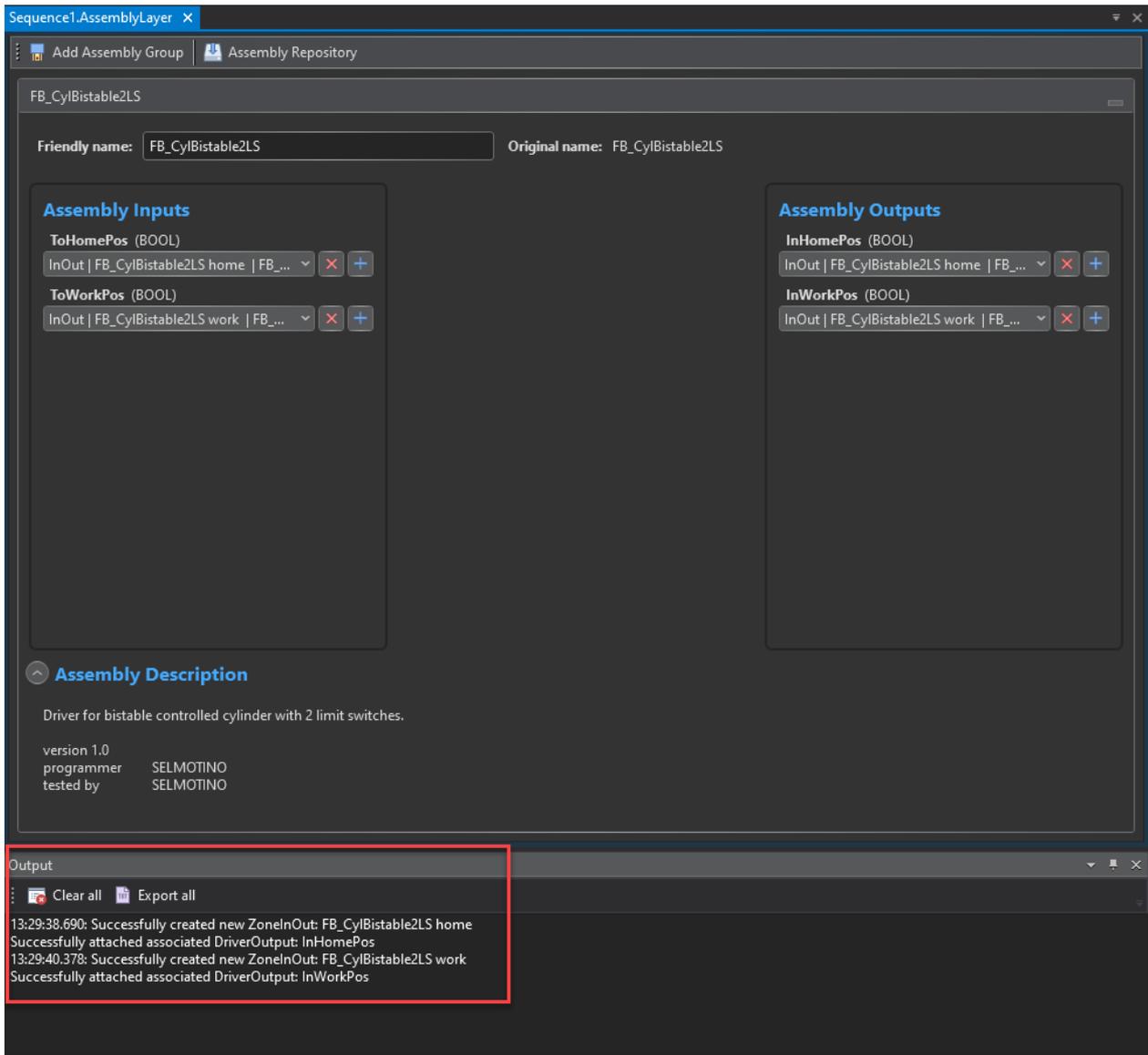
Studio

Es wird vorgeschlagen, welche Zonen für die Eingänge und Ausgänge eingefügt werden sollen.



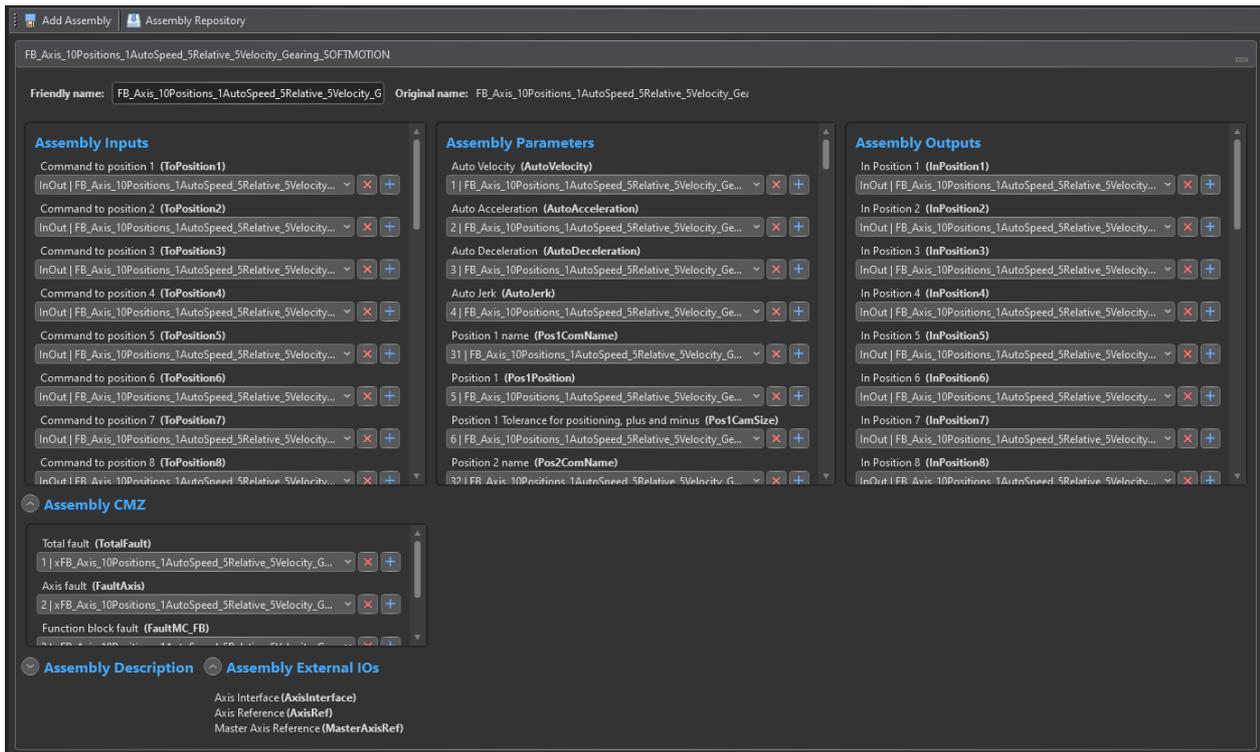
Studio

Im Ausgabefenster wird angezeigt, dass die Zonen erfolgreich erstellt worden sind.



Studio

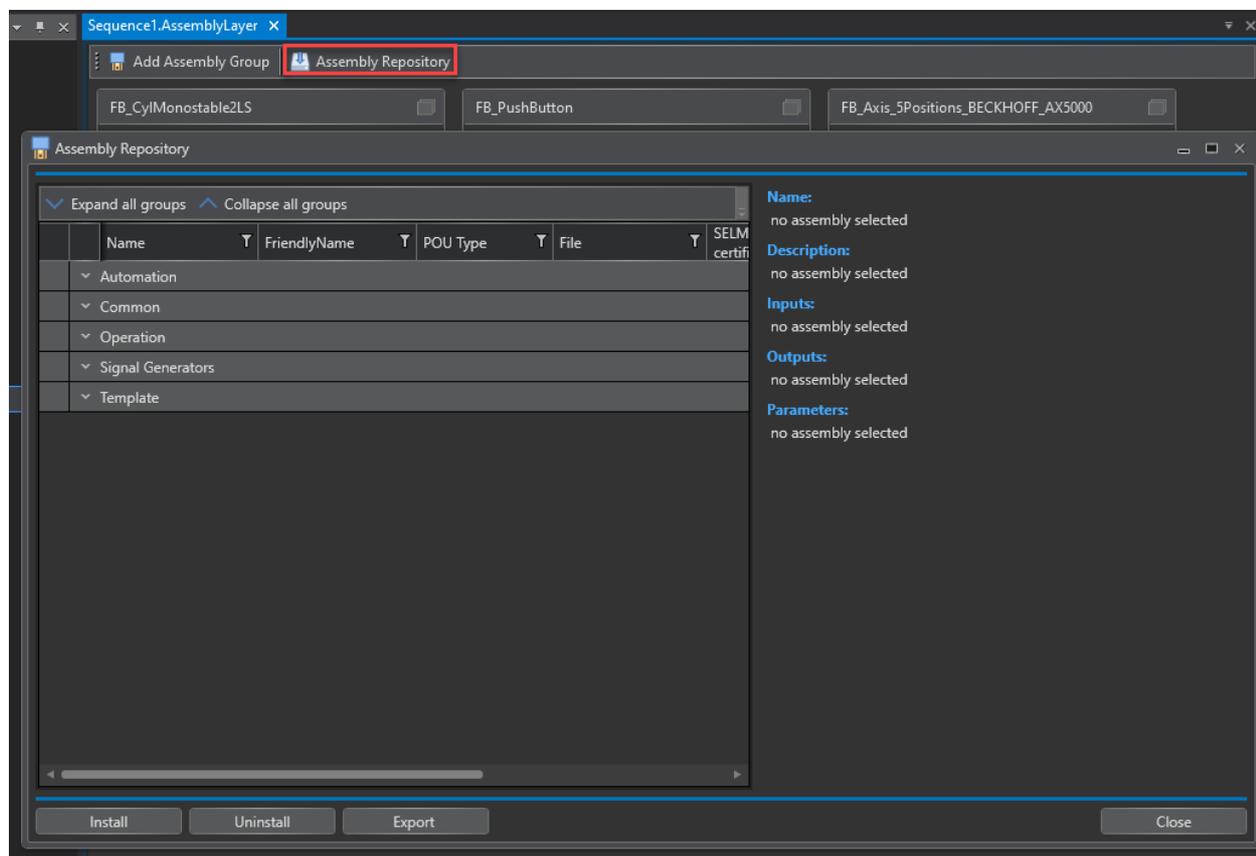
Die Zonen werden automatisch in die Systemebene integriert und die grundlegenden Eigenschaften sind bereits vorausgefüllt, was einen erheblichen Arbeitsaufwand einspart. Es müssen lediglich die Prozess-definierten Eigenschaften ergänzt werden. Falls der Treiber dies unterstützt, besteht auch die Möglichkeit, entsprechende Parameter und CMZ's automatisch zu erstellen.



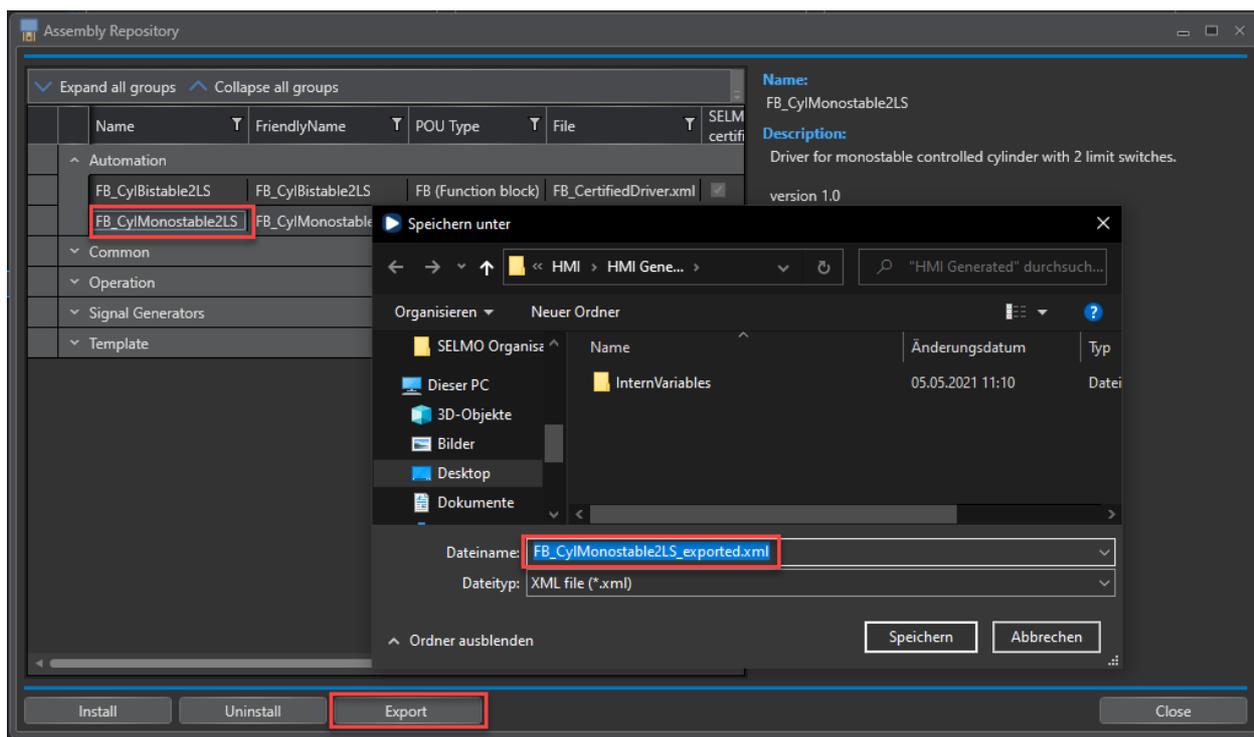
Studio

Assembly Repository

Im Assembly Repository haben Sie einen Überblick über alle verfügbaren Treiber für Ihr Projekt. Sie haben die Möglichkeit, neue Treiber, die nach der Selmo-Struktur erstellt wurden, zu importieren, zu exportieren oder aus der Sammlung zu entfernen.

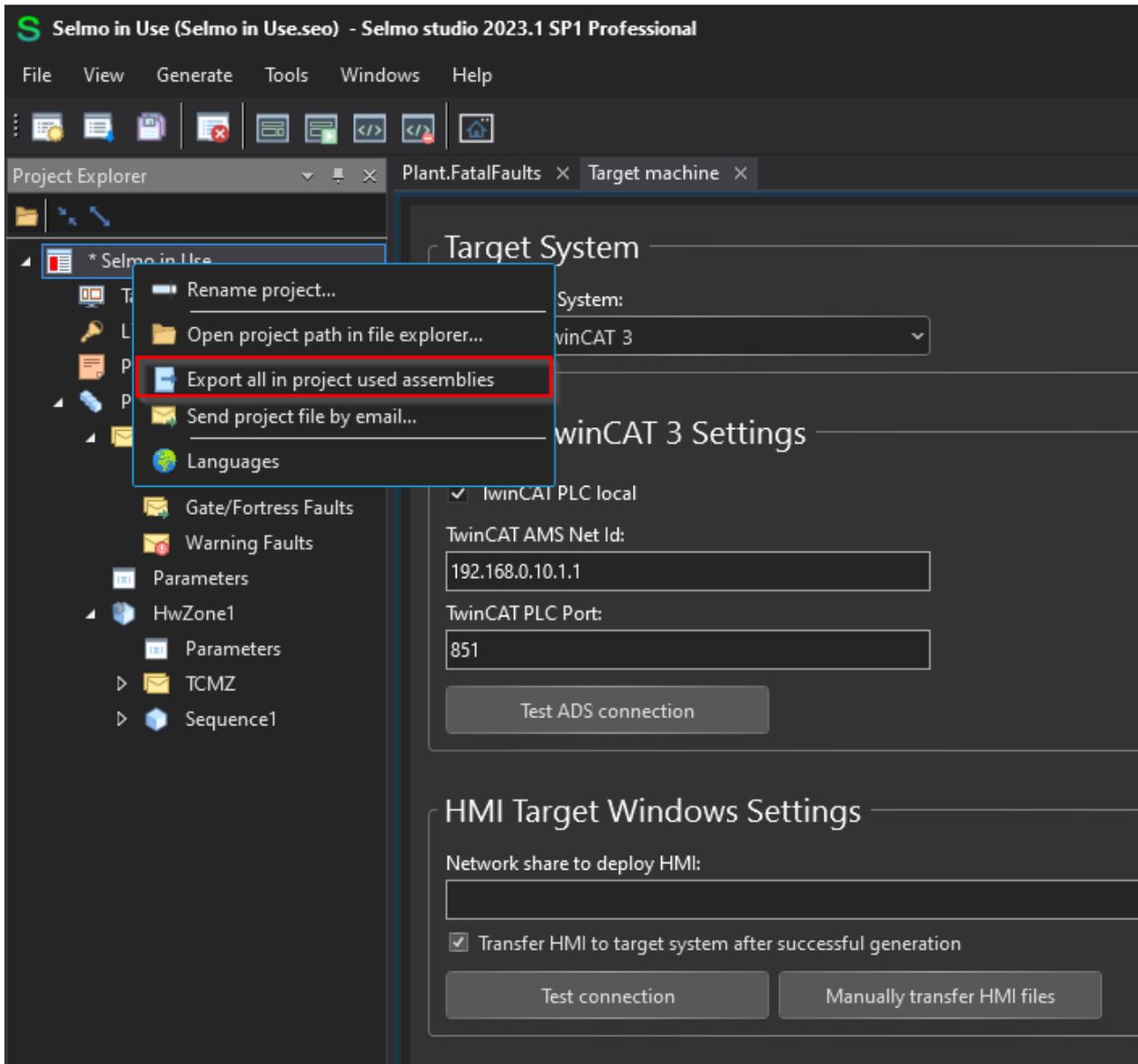


Studio



Studio

Nachdem das Projekt abgeschlossen wurde bzw. während der Projektierung, können alle verwendeten Assemblies exportiert werden.



Studio

1.9 HMI

HMI steht für Human-Machine Interface und bezieht sich auf die Schnittstelle zwischen einem menschlichen Bediener und einer Maschine, einem System oder einem Prozess. Die HMI kann eine grafische Benutzeroberfläche sein, die es dem Bediener ermöglicht, Daten zu überwachen, Einstellungen zu ändern oder Aktionen auszuführen.

Bei Selmo wird die HMI Entwicklung auf eine effiziente und automatisierte Weise durchgeführt. Selmo hat ein fortschrittliches System entwickelt, das die Generierung der HMI direkt aus dem Prozessmodell ermöglicht. Traditionell erforderte die Erstellung einer HMI, also der Benutzerschnittstelle zwischen Mensch und Maschine, einen zeitaufwändigen Prozess. Entwickler mussten manuell Grafiken, Bedienelemente und Interaktionen entwerfen und programmieren. Dies war nicht nur arbeitsintensiv, sondern auch fehleranfällig und zeigte oft eine Diskrepanz zwischen dem Prozessmodell und der tatsächlichen HMI.

Selmo hat dieses Problem erkannt und eine innovative Lösung entwickelt, um den HMI-Entwicklungsprozess zu optimieren. Durch die Verbindung des Prozessmodells mit der HMI-Generierung werden die HMI-Elemente automatisch aus den Daten des Prozessmodells erzeugt. Das Prozessmodell dient als grundlegende Struktur für die HMI. Es enthält Informationen über den Ablauf des Prozesses, die verschiedenen Zustände und die relevanten Parameter. Basierend auf diesen Informationen generiert das System von Selmo automatisch die entsprechenden Benutzerschnittstellen. Die Vorteile dieser automatisierten HMI-Generierung sind vielfältig.

Erstens spart sie Entwicklern eine erhebliche Menge an Zeit und Aufwand. Statt jede einzelne grafische Komponente manuell zu erstellen, können sie sich auf die Erstellung des Prozessmodells konzentrieren, während die HMI automatisch generiert wird.

Zweitens verbessert die automatisierte Generierung die Konsistenz und Genauigkeit der HMI. Da sie direkt aus dem Prozessmodell abgeleitet wird, ist die HMI immer auf dem neuesten Stand und spiegelt den tatsächlichen Prozess korrekt wider. Dadurch werden potenzielle Fehler und Inkonsistenzen vermieden, die bei manueller Entwicklung auftreten könnten.

Darüber hinaus ermöglicht die automatisierte HMI-Generierung eine bessere Zusammenarbeit zwischen den verschiedenen Teammitgliedern. Entwickler, Ingenieure und Bediener können gemeinsam am Prozessmodell arbeiten und Änderungen vornehmen, die automatisch in der generierten HMI aktualisiert werden.

Der Selmo Ansatz zur automatisierten HMI-Generierung hat das Potenzial, die Effizienz und Qualität der HMI-Entwicklung in der Industrieautomatisierung erheblich zu verbessern. Durch die Integration von Prozessmodell und HMI wird die Entwicklung beschleunigt, Fehler reduziert und die Zusammenarbeit erleichtert.

Selmo ist damit ein Vorreiter in der Branche und treibt den Fortschritt in der Human-Machine-Interface-Technologie voran.

1.9.1 Sequence Control

Bei Selmo ist die HMI (Human-Machine Interface) mit einer leistungsstarken Sequence Control-Funktion ausgestattet, die es ermöglicht, für jede Sequence im Prozess ein individuelles Sequence Control zu erstellen. Diese Funktion ermöglicht die präzise Steuerung, Überwachung und Diagnose von Abläufen innerhalb des Produktionsprozesses. Das Sequence Control bietet eine klare Übersicht über die Schritte und Zonen, die für den reibungslosen Ablauf einer bestimmten Sequence erforderlich sind.

Eine besonders nützliche Funktion des Sequence Control ist die Single Step Funktion. Diese Funktion ermöglicht es dem Bediener, einen Schritt manuell auszuführen, um den Ablauf zu testen oder zu überprüfen. Dadurch kann der Bediener den Prozess schrittweise durchgehen und sicherstellen, dass jeder Schritt ordnungsgemäß funktioniert, bevor er zum nächsten Prozessschritt übergeht.

Darüber hinaus bietet das Sequence Control-System eine Schrittauswahl Funktion. Diese Funktion ermöglicht es dem Bediener, zwischen verschiedenen Schritten in einer Sequence zu wählen, basierend auf den aktuellen Anforderungen oder Bedingungen. Dies ist besonders nützlich, wenn bestimmte Schritte übersprungen oder wiederholt werden müssen, um den Anforderungen des Prozesses gerecht zu werden.

Eine weitere praktische Eigenschaft des Sequence Control ist die automatische Suche eines gültigen Schrittes. Dies bedeutet, dass das System automatisch nach dem nächsten gültigen Schritt sucht, der basierend auf den aktuellen Bedingungen oder vordefinierten Regeln ausgeführt werden soll. Dies spart Zeit und stellt sicher, dass der Prozess reibungslos abläuft, ohne dass der Bediener manuell nach dem nächsten Schritt suchen muss.

Insgesamt ermöglicht das Sequence Control eine präzise Kontrolle und Diagnose von Abläufen. Es bietet eine klare Übersicht über die erforderlichen Schritte und Zonen, erleichtert die Schrittauswahl und ermöglicht eine einfache Suche nach gültigen Schritten. Diese Funktionen tragen dazu bei, die Effizienz des Produktionsprozesses zu verbessern, menschliche Fehler zu minimieren und eine zuverlässige Ausführung der Sequenzen zu gewährleisten.

Selmo powered HMI - codesys_test_2024.4

Selmo No alarms 11:56:36
Monday, April 22, 2024

Hardware Zones: **HwZone1**

HwZone Controls

Automatic Mode: Automatic Release

Automatic Release

EOC mode

Lamp test

Any sequences ready to start
All sequences ready to start
Automatic Lamp
Safety Gate Lamp
Emergency Stop Lamp
EOC reached Lamp

Overview **Sequence1**

Sequence Automatic Release

Step decrement Step next valid Step increment

Single Step On Step Single Ghost Mode Step Reset

Parameters

Parameter Timer 1:

Previous step: 2: Step 2

Actual step: **3: Timer 1**

Next step: 4: Step 3

Waiting for **Timer 1**
Time left: 1.5s

Global reset
Global sequence step reset
Global automatic release
Lamp test

Language: English

Features: Show historic data, Manage process set value snapshots, Take a screenshot

Settings: Switch Fullscreen/Window, Close HMI, Minimize HMI

HMI

Sequence starten

Um den Automatikbetrieb der Sequenz zu starten, muss der Automatic-Mode-Schalter aktiviert werden.

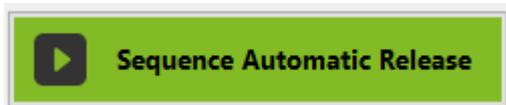


Ist kein Fehler bzw. CMZ aktiv und somit die 'Waiting for' blau eingefärbt, kann der Automatik-Betrieb über den Button,

gestartet werden. Damit werden alle sich in dieser HwZone befindlichen Sequences freigegeben.



Falls Sie nur diese Sequence freigeben möchten, steht Ihnen dafür ein spezieller Button zur Verfügung.



Mit dem Button besteht die Möglichkeit, alle Sequenzen in allen Hardware-Zonen zu starten, in denen der Automatic-Mode-Schalter auf "Automatik" steht.



Wie in der 'Waiting for' angezeigt, ist es notwendig den Button, für die eingestellte Zeit des Startup-Delay im Selmo-Studio,

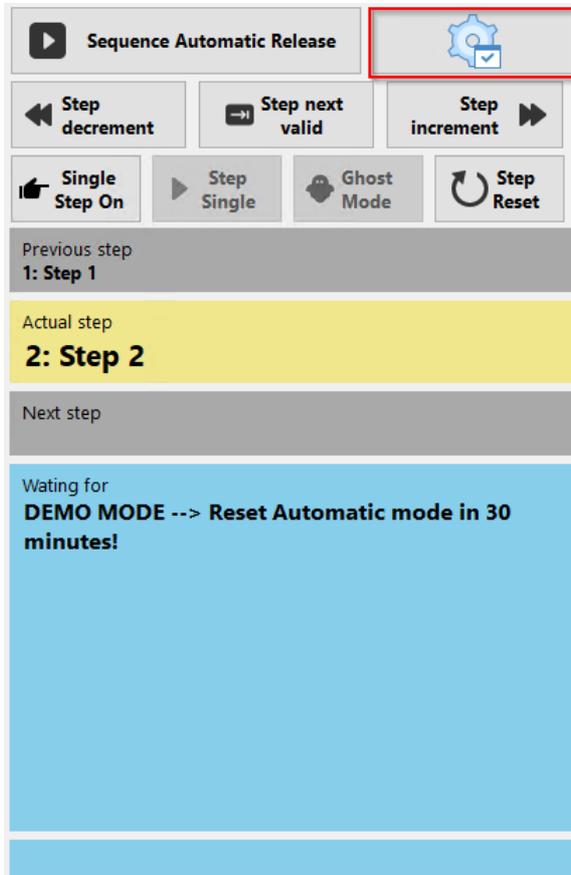
gedrückt zu halten. Der Default-Wert beträgt 3s.

The screenshot displays the Selmo HMI interface. At the top, it shows 'Selmo' and 'No alarms' on the left, and the time '13:55:17' and date 'Monday, May 6, 2024' on the right. Below the header, there are three hardware zones: 'HwZone3', 'HwZone2', and 'HwZone1'. The 'HwZone Controls' section on the left includes 'Automatic Mode', 'HwZone Automatic Release', 'EOC mode', and 'Lamp test'. The main area is titled 'Overview' and 'Sequence2'. It features a 'Sequence Automatic Release' button, 'Step decrement', 'Step next valid', and 'Step increment' buttons. Below these are 'Single Step On', 'Step Single', 'Ghost Mode', and 'Step Reset' buttons. The sequence status is shown as 'Actual step 1: Step 1' and 'Next step 2: Step 2'. A blue box contains the text: 'Waiting for Internal: Press sequence automatic release or global automatic release for 3s Internal: Waiting 'Step Single' button pressed (only in Single Step Mode)'. On the right side, there are buttons for 'Global reset', 'Global sequence step reset', 'Global automatic release', and 'Lamp test'. Below these are 'Language' (English), 'Features' (Show historic data, Manage process set value snapshots, Take a screenshot), and 'Settings' (Switch Fullscreen/Window, Close HMI, Minimize HMI).

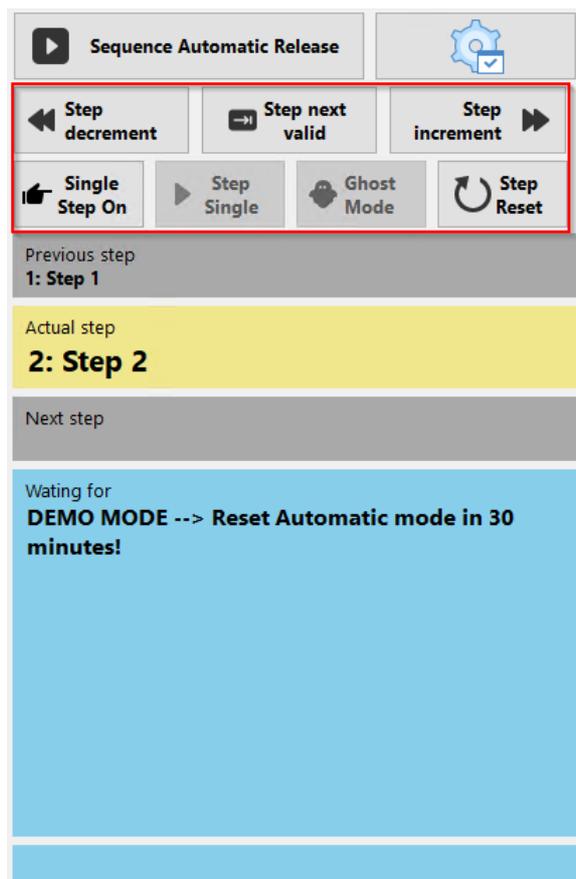
HMI

Sequence Control

Durch Klicken auf das Zahnrad-Symbol öffnet sich die Benutzeroberfläche der Sequence Control.



HMI



HMI

Hinweis

Es ist wichtig zu beachten, dass die Schritt Funktionen "Step decrement", "Step next valid", "Step increment" und "Step Reset" nur im manuellen Modus verfügbar sind und Ihnen die Möglichkeit gibt, die Sequenz schrittweise zu steuern. Sobald der Automatic Mode aktiviert ist, übernimmt das System die automatische Ausführung der Schritte gemäß den vorgegebenen Bedingungen und Regeln.

Step decrement

Der "Step decrement" ist eine Funktion, die es ermöglicht, einen Schritt in der Sequence rückgängig zu machen. Durch die Verwendung dieser Funktion können Sie zuvor ausgeführte Schritte in der Sequence wieder zurückgehen und so den vorherigen Zustand wiederherstellen. Es ist praktisch, um Fehler zu korrigieren oder bestimmte Aktionen zu wiederholen, die möglicherweise übersprungen oder falsch ausgeführt wurden.

Step next valid

Der "Step next valid" ist eine Funktion, mit der Sie den nächsten möglichen Zustand suchen können, der zum aktuellen Maschinenzustand passt. Diese Funktion ist besonders nützlich, wenn Sie in einer Sequence bestimmte Schritte durchlaufen und sicherstellen möchten, dass der nächste Schritt gültig ist und zum aktuellen Zustand der Maschine passt. Durch Betätigen des "Step next valid"-Buttons analysiert die Sequence den aktuellen Zustand der Maschine und sucht den nächsten Schritt, der basierend auf den festgelegten Bedingungen und Regeln für den Maschinenzustand gültig ist. Auf diese Weise können Sie sicherstellen, dass die Schritte in der Sequence entsprechend der aktuellen Maschinensituation ausgeführt werden. Diese Funktion ermöglicht eine präzise und automatische Navigation durch die Sequence, indem sie sicherstellt, dass nur diejenigen Schritte ausgeführt werden, die mit dem aktuellen Zustand der Maschine kompatibel sind. Dadurch wird die Effizienz verbessert und potenzielle Fehler oder Inkonsistenzen vermieden.

Step increment

Der "Step increment" ist eine Funktion, mit der Sie einen Schritt nach vorne in der Sequence machen können. Durch das Drücken des "Step increment"-Buttons können Sie zur nächsten Aktion in der Sequence springen und den Fortschritt vorantreiben. Diese Funktion ist besonders nützlich, wenn Sie bestimmte Schritte in der Sequence überspringen möchten, entweder weil Sie sicher sind, dass sie bereits erfolgreich abgeschlossen wurden oder aus anderen Gründen. Durch das Voranschreiten in der Sequence können Sie Zeit sparen und den Arbeitsablauf beschleunigen, indem Sie direkt zum nächsten relevanten Schritt gelangen. Es ist wichtig zu beachten, dass beim Verwenden des "Step increment" darauf geachtet werden sollte, dass der übersprungene Schritt nicht wesentliche Aktionen oder Bedingungen enthält, die den weiteren Verlauf der Sequence beeinflussen könnten. Eine sorgfältige Überprüfung der Sequence Struktur und des aktuellen Zustands der Maschine ist ratsam, um sicherzustellen, dass das

Überspringen eines Schritts keine unerwarteten Konsequenzen hat. Insgesamt bietet der "Step increment" eine praktische Möglichkeit, den Fortschritt in einer Sequence zu steuern und gezielt zu den relevanten Schritten zu gelangen, die als nächstes ausgeführt werden sollen.

Step Reset

Der "Step Reset" ist eine Funktion, mit der Sie den Schrittzähler zurücksetzen können. Durch das Drücken des "Step Reset"-Buttons wird der Zähler auf den Anfangswert zurückgesetzt. Diese Funktion ist besonders nützlich, wenn Sie den Fortschritt in der Sequence zurücksetzen und von vorne beginnen möchten. Indem Sie den Schrittzähler zurücksetzen, haben Sie die Möglichkeit, die Sequence erneut abzuspielen, ohne den bisherigen Fortschritt zu berücksichtigen. Dies kann hilfreich sein, wenn Sie eine Fehlerbehebung durchführen möchten oder eine Sequence erneut testen möchten, ohne die vorherigen Durchläufe zu berücksichtigen. Es ist wichtig zu beachten, dass beim Zurücksetzen des Schrittzählers auch andere Variablen oder Zustände, die im Zusammenhang mit dem Schrittfortschritt stehen, zurückgesetzt werden können. Dies hängt von der spezifischen Implementierung der Sequence ab. Der "Step Reset" bietet eine einfache und effektive Möglichkeit, den Schrittfortschritt in der Sequenz zurückzusetzen und bei Bedarf von vorne zu beginnen.

Hinweis

Der Single-Step-On Mode kann unabhängig von der Betriebsart ein- oder ausgeschaltet werden. Der Bestätigung-Button, wie der "Step Single", funktioniert jedoch nur wenn die Automatik freigegeben ist.

Single Step On

Durch das Einschalten des "Single Step On" wird der Single-Step-Modus aktiviert. Im Single-Step-Modus wird die Sequence so konfiguriert, dass sie nach jedem einzelnen Schritt anhält und auf Ihre Bestätigung wartet, bevor sie zum nächsten Schritt übergeht. Dieser Modus ermöglicht es Ihnen, die Sequence Schritt für Schritt zu durchlaufen und jede Aktion einzeln zu überprüfen oder zu testen. Nachdem ein Schritt ausgeführt wurde, müssen Sie den "Step Single"-Button drücken, um die Ausführung fortzusetzen und zum nächsten Schritt zu gelangen. Dadurch haben Sie die volle Kontrolle über den Ablauf und können jeden Schritt sorgfältig überwachen. Der Single-Step-Modus ist besonders nützlich bei der Fehlersuche, beim Testen oder bei der schrittweisen Überprüfung einer Sequence. Sie können den Zustand der Maschine oder andere relevante Parameter nach jedem Schritt überprüfen und sicherstellen, dass alles wie erwartet funktioniert, bevor Sie zum nächsten Schritt übergehen.

Step Single

Der "Step Single" ist ein Bestätigungsbutton, der im Single-Step-Modus verwendet wird, um die Schritte schrittweise weiter zuschalten. Im Single-Step-Modus stoppt die Sequence nach jedem Schritt und wartet auf Ihre Bestätigung, bevor sie zum nächsten Schritt übergeht. Wenn Sie bereit sind, zum nächsten Schritt zu wechseln, drücken Sie den "Step Single"-Button, um die Ausführung fortzusetzen und den nächsten Schritt auszulösen. Durch das Drücken des "Step Single"-Buttons haben Sie die Kontrolle darüber, wie schnell die Sequence fortschreitet. Sie können den Fortschritt Schritt für Schritt steuern und jede Aktion einzeln überprüfen, bevor Sie zum nächsten Schritt übergehen. Dies ermöglicht eine detaillierte Überwachung und Prüfung des Ablaufs und ist besonders nützlich bei der Fehlersuche, beim Testen oder bei der schrittweisen Überprüfung der Sequence. Der "Step Single"-Button ermöglicht es Ihnen, den Single-Step-Modus effektiv zu nutzen und den Ablauf der Sequence in Ihrem eigenen Tempo voranzutreiben.

Ghost Mode

Der Ghost-Modus kann zu beispielsweise für Inbetriebnahmezwecken verwendet werden. Wenn beispielsweise ein Sensor in einer Zone fehlt, kann der Ghost-Modus für diese Zone aktiviert werden, um das gesamte Modell dennoch zu testen. Bei der Zone im Ghost-Modus startet eine Verzögerung, bis der Sequenzcheck erfüllt ist, um dann fortzufahren.

Ghost Mode	True
Ghost Mode Delay	1000

In der HMI ist der Ghost-Mode-Button grau hinterlegt, was bedeutet, dass der Ghost-Mode separat aktiviert werden muss. Der Ghost-Mode kann nur über den Safety Function Key aktiviert werden, der als Schüsselschalter fungiert und nur für autorisierte Personen zugänglich ist. Der Safety Function Key existiert pro Hardwarezone und muss in Codesys unter den IOs der Hardwarezone auf True gesetzt werden, um den Button in der HMI starten zu können.

Ausdruck	Datentyp	Wert	Vorber...	Adresse	Komm...
xManualModeKeySwitch	BOOL	FALSE			zone ma...
xAutomaticModeKeySwitch	BOOL	TRUE			zone au...
xSafetyFunctionKeySwitch	BOOL	FALSE			zone Sa...
xEmergencyStopRelay	BOOL	FALSE			zone em...
xFaultReset	BOOL	FALSE			zone fa...

The screenshot displays the Selmo HMI interface. At the top left, the 'Selmo' logo is visible. Below it, the 'Hardware Zones' section shows 'HwZone1' with a red box around its status indicator. The 'HwZone Controls' section includes buttons for 'Automatic Mode', 'HwZone Automatic Release' (highlighted with a red box), 'EOC mode', and 'Lamp test'. A legend on the left side lists various status indicators: 'Any sequences ready to start', 'All sequences ready to start', 'Automatic Lamp', 'Safety Gate Lamp', 'Emergency Stop Lamp', and 'EOC reached Lamp'. The main control area shows 'Overview' and 'Sequence1' tabs, with 'Sequence1' highlighted. Below the tabs are several control buttons: 'Sequence Automatic Release', 'Step decrement', 'Step next valid', 'Step increment', 'Single Step On', 'Step Single', 'Ghost Mode' (highlighted with a red box), and 'Step Reset'. The sequence progress is shown as: 'Previous step 2: Step 2', 'Actual step 3: Timer 1', and 'Next step 4: Step 3'. A large blue area at the bottom indicates 'Waiting for Timer 1 Timer' with 'Time left: 2.8s'.

Schritt Information



HMI

Preview step

Dieser Schritt zeigt die vorherige Stufe oder den vorherigen Zustand an.

Actual step

Dieser Schritt zeigt den gegenwärtigen Zustand oder den aktuellen Fortschritt an.

Next step

Dieser Schritt zeigt den nächsten Schritt oder die nächste erforderliche Aktion an.

Zonen Information



HMI

Waiting for

Die definierten Bedingungen bzw. Zonen für diesen Schritt werden angezeigt, um den nächsten Schritt zu erreichen. Dies ermöglicht dem Bediener sofort zu erkennen, welches Signal fehlt, und entsprechend zu handeln.

z.B.: Sequence-Check Zone 1

⊖ Zone 1

Monitoring-Check Zone 2

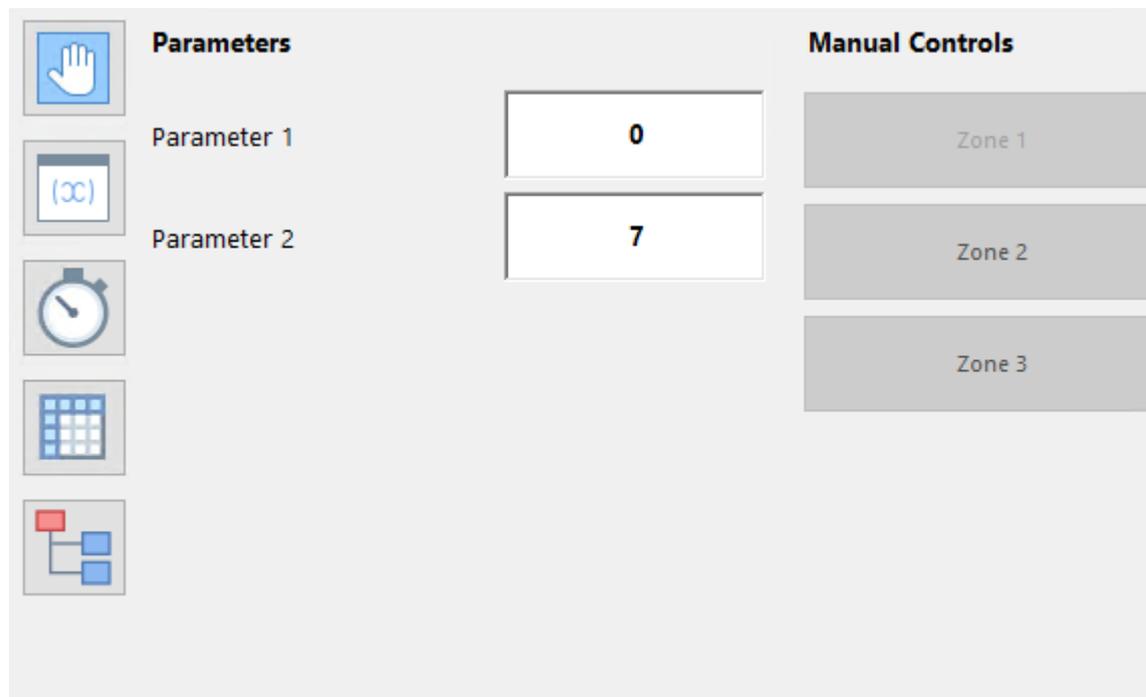
⚠ Zone 2



HMI

Wenn die Automatik beendet oder durch einen Interlock unterbrochen wird, werden die Zoneninformationen (Interlock- oder Sequence Check) gespeichert. Dies dient einer verbesserten Analyse. Diese Einträge können durch drücken des Global Reset Button gelöscht werden.

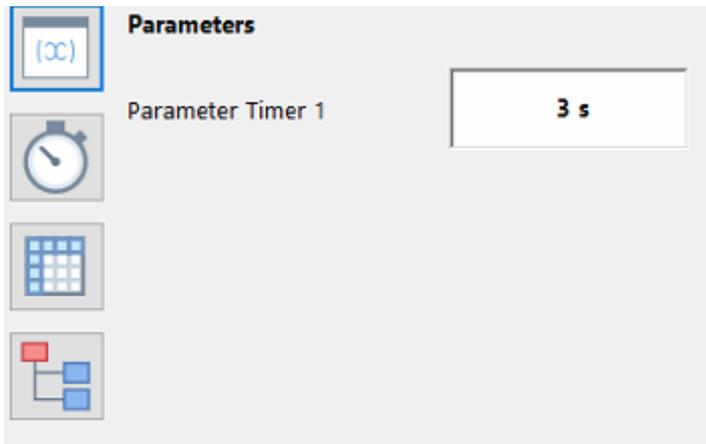
Handbetrieb



HMI

Die im Selmo-Studio festgelegten Zonen mit dem "[HMI Button](#)⁹¹" ermöglichen es, diese von der HMI aus zu steuern, solange der Automatik-Betrieb nicht aktiv ist. Es ist wichtig zu beachten, dass der Handbetrieb nur dann möglich ist, wenn keine Störung (TCMZ, CMZ) aktiv ist.

Parameter



HMI

In diesem Bereich sind die [Parameter](#)¹³⁶ für diese Sequence zu finden, die im Selmo-Studio erstellt wurden.

Step Time Monitoring

	Actual	Last	Min	Avg	Max	Timeout Count	Timeout	Timeout Add	Disable Timeout
Step 1	0.000	0.790	0.768	0.938	0.791	0	300	5	<input type="checkbox"/>
Step 2	0.000	1.036	1.018	1.032	1.041	0	300	5	<input type="checkbox"/>
Timer 1	1.883	3.005	2.998	3.010	3.020	0	300	5	<input type="checkbox"/>
Step 3	0.000	1.280	1.268	1.276	1.291	0	300	5	<input type="checkbox"/>
Step 4	0.000	1.518	1.518	1.524	1.541	0	300	5	<input type="checkbox"/>

HMI

Bei der Auswertung der Zeit, die ein Schritt benötigt hat, können verschiedene Metriken verwendet werden, um ein umfassendes Bild der Leistung zu erhalten. Hier sind einige gängige Metriken, die in Verbindung mit dem Step Time Monitoring verwendet werden:

Actual (Tatsächlich): Dies ist die tatsächlich gemessene Zeit, die ein bestimmter Schritt für die Ausführung benötigt hat. Es handelt sich um den exakten Wert, der basierend auf den gemessenen Daten erfasst wurde.

Last (Letzte): Dies bezieht sich auf die Zeit, die der Schritt bei der letzten Ausführung benötigt hat. Es zeigt den zuletzt gemessenen Wert und kann verwendet werden, um Veränderungen in der Leistung über die Zeit zu verfolgen.

Min (Minimum): Dies ist der kleinste gemessene Wert für die Zeit, die der Schritt benötigt hat. Es gibt an, wie schnell der Schritt unter den besten Bedingungen ausgeführt werden kann.

Avg (Durchschnitt): Diese Metrik gibt den Durchschnittswert der Zeiten an, die der Schritt während einer bestimmten Periode oder einer bestimmten Anzahl von Ausführungen benötigt hat. Der Durchschnittswert kann Aufschluss über die typische oder durchschnittliche Leistung des Schritts geben.

Max (Maximum): Dies ist der größte gemessene Wert für die Zeit, die der Schritt benötigt hat. Es zeigt die längste Zeit, die der Schritt für die Ausführung benötigt hat und kann auf Engpässe oder ungewöhnliche Verzögerungen hinweisen.

Timeout Count (Timeout Zähler): Bezieht sich auf die Anzahl der Vorkommnisse, bei denen ein Schritt oder eine Operation innerhalb eines definierten Zeit-Limits oder Timeout-Werts nicht abgeschlossen wurde. Es zählt die Anzahl der Fälle, in denen die Ausführungszeit eines Schrittes den maximal zulässigen Zeitrahmen überschritten hat.

Durch die Kombination dieser Auswertungen - Actual, Last, Min, Avg, Max und Timeout Count - kann eine umfassende Analyse der Leistung und Ausführungszeiten einzelner Schritte ermöglicht werden. Dadurch können potenzielle Engpässe, ineffiziente Bereiche oder Probleme identifiziert werden, die optimiert werden müssen, um die Gesamtleistung zu verbessern.

Timeout

Indem ein Timeout-Wert festgelegt wird, kann das System die Ausführungszeit eines Schrittes überwachen und prüfen, ob er innerhalb des definierten Zeitlimits abgeschlossen wird. Wenn die Ausführungszeit den Timeout-Wert überschreitet, wird der Schritt als fehlgeschlagen oder nicht rechtzeitig abgeschlossen betrachtet.

Timeout Add

Timeout additional" bezieht sich auf einen zusätzlichen Wert, der zum bestehenden Timeout-Wert addiert wird. Dadurch wird der Timeout-Wert für eine bestimmte Schritt verlängert. Angenommen, Sie haben bereits einen Timeout-Wert von 10 Sekunden festgelegt und möchten einen zusätzlichen Wert von 5 Sekunden hinzufügen. In diesem Fall würde der neue Timeout-Wert 15 Sekunden betragen.

Disable Timeout

Die Deaktivierung des Timeout oder die Deaktivierung der Schrittzeitüberwachung bezieht sich auf den Vorgang, bei dem eine Funktion oder ein Prozess keine zeitlichen Begrenzungen oder Grenzwerte für die Ausführungsdauer hat.

Teach Mode

Der "Teach Mode" ermöglicht das Einlernen der Timeout-Zeiten, indem 5 min gemessen und die Timeout-Zeiten daraus abgeleitet werden.

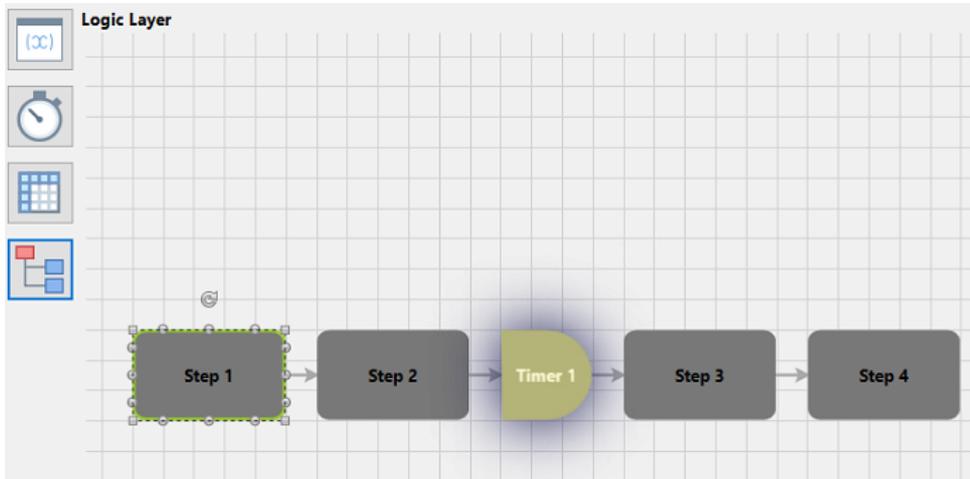
System Layer

Step/Zone	Zone 2	Zone 3	Timer 1	Zone 4	Zone 5
Step 1	1	0	0	0	0
Step 2	0	1	0	0	0
Timer 1	0	0	1	0	0
Step 3	0	0	0	1	0
Step 4	0	0	0	0	1

HMI

Ansicht des System-Layer, wie im Selmo-Studio definiert.

Logic Layer



Ansicht des Logic-Layer, wie im Selmo-Studio definiert, mit Highlighting des aktiven Schritts.

1.9.2 Alarm Handling

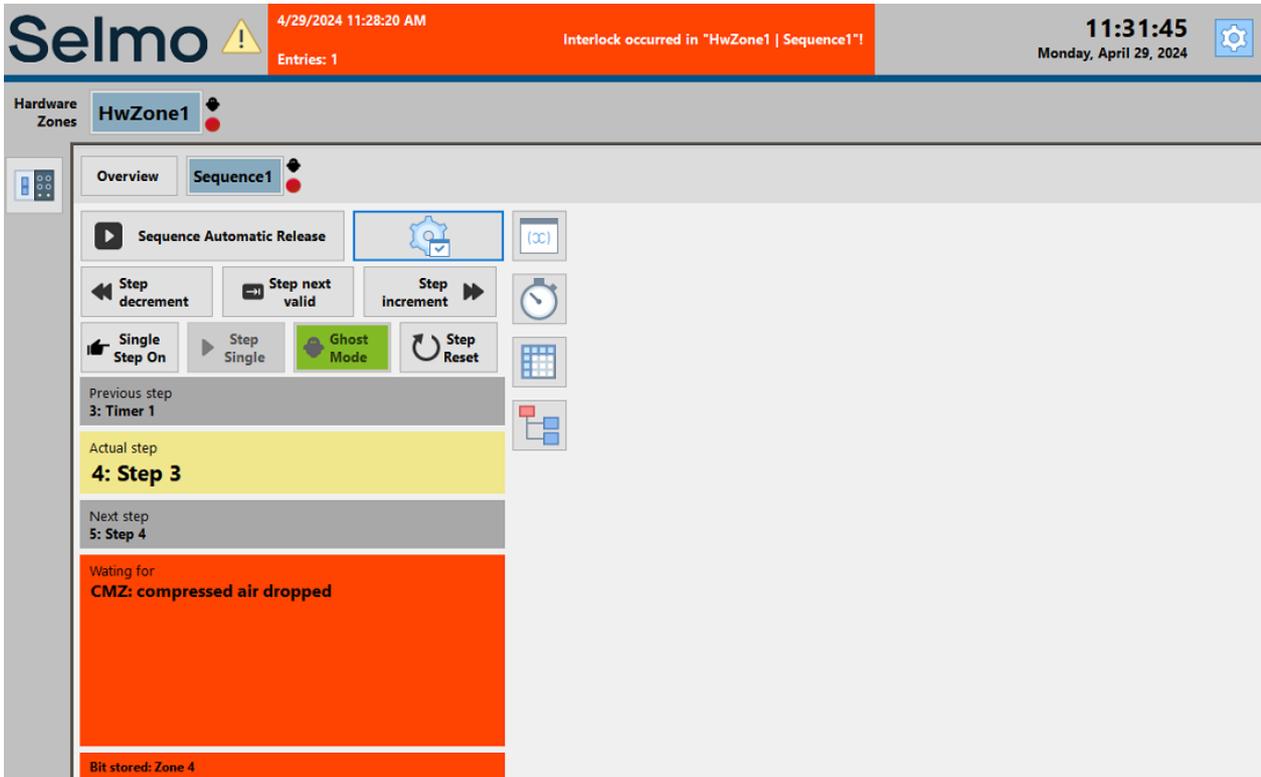
Alarm Handling bezieht sich auf den Prozess der Behandlung von Alarmen oder Warnmeldungen, die in einem System auftreten. Es beinhaltet die Identifizierung, Erfassung, Priorisierung, Verarbeitung und Behebung von Alarmzuständen.

Das Alarm Handling umfasst folgende Schritte:

1. **Alarmerkennung:** Das System erkennt eine Abweichung von normalen Betriebszuständen und generiert einen Alarm.
2. **Alarmanzeige:** Der Alarm wird auf einer Benutzeroberfläche oder in einem Überwachungssystem angezeigt. Dies kann in Form von visuellen Hinweisen, akustischen Signalen oder Meldungen erfolgen.
3. **Alarmpriorisierung:** Die Alarmerkennung wird nach ihrer Dringlichkeit oder Schweregrad priorisiert, um festzustellen, welche Alarmerkennung zuerst behandelt werden müssen.
4. **Alarmbenachrichtigung:** Das Alarm-Management-System informiert relevante Personen oder Teams über das Auftreten eines Alarms. Dies kann durch E-Mail-Benachrichtigungen, SMS-Nachrichten oder andere Kommunikationsmittel erfolgen.
5. **Alarmanalyse:** Die Ursache des Alarms wird untersucht, um festzustellen, was das Problem verursacht hat. Dies kann eine manuelle Überprüfung der Systeme, Protokolle oder Sensordaten erfordern.
6. **Alarmreaktion:** Basierend auf der Analyse wird eine geeignete Reaktion eingeleitet, um das Problem zu lösen oder die Auswirkungen zu minimieren. Dies kann das Ergreifen von Maßnahmen zur Fehlerbehebung, das Einleiten von Wartungsarbeiten oder das Herbeirufen von Fachpersonal umfassen.
7. **Alarmverfolgung und Dokumentation:** Der gesamte Prozess des Alarmhandlings wird dokumentiert, einschließlich der Zeitpunkte der Alarmerkennung, der durchgeführten Maßnahmen und der Ergebnisse.

Ein effektives Alarm Handling ermöglicht es, potenzielle Probleme frühzeitig zu erkennen, schnell zu reagieren und Betriebsstörungen zu minimieren, um einen reibungslosen Betrieb des Systems sicherzustellen.

Wenn eine Alarm in Form eines CMZ (Constantly Monitored Zone) oder TCMZ (Total Constantly Monitored Zone) auftritt, wird dies in der Alarmleiste und in der "Waiting for"-Box der Sequence angezeigt. Gleichzeitig ändert sich die Hintergrundfarbe der "Waiting for" Anzeige auf Rot, und in der Sequence Overview wird eine rote Lampe aktiviert, um die Störung zu signalisieren. In jedem Fall müssen Fehler mit einem "Reset" bestätigt werden, bevor der Automatik-Betrieb wieder aufgenommen werden kann.



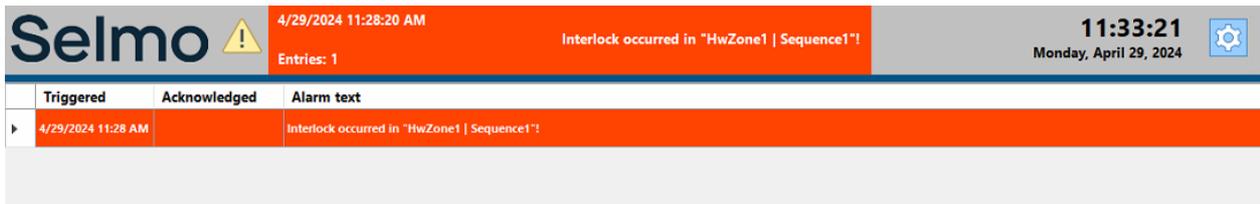
HMI

Warnungen hingegen werden nur in der Alarmleiste angezeigt, die orange hinterlegt sind. Diese Warnungen haben keinen Einfluss auf den Automatik-Betrieb.



HMI

Durch einen Doppelklick auf die Alarmleiste öffnet sich die Alarmübersicht auf der HMI. In der Alarmliste werden sowohl Störungen als auch Warnungen angezeigt, solange sie aktiv sind. Hier werden alle relevanten Informationen zu den Alarmsignalen, wie Art der Störung oder Warnung, Zeitpunkt des Auftretens und eventuell zugeordnete Informationen, aufgelistet. Die Alarmliste ermöglicht eine übersichtliche Darstellung und Nachverfolgung aller aktuellen Alarmzustände, sodass das Bedienpersonal schnell über den Status informiert ist und gegebenenfalls entsprechende Maßnahmen ergreifen kann.



Triggered	Acknowledged	Alarm text
4/29/2024 11:28 AM		Interlock occurred in "HwZone1 Sequence1"!

HMI

In der Alarm-Historie werden die aufgetretenen Alarme mit Zeitstempel protokolliert und abgespeichert. Dabei werden alle vergangenen Alarmereignisse erfasst und chronologisch geordnet. Dadurch ermöglicht die Alarm-Historie eine nachträgliche Analyse und Rückverfolgung vergangener Alarmzustände. Das Speichern der Alarme mit Zeitstempeln hilft bei der Identifizierung von Trends, dem Erkennen von wiederkehrenden Problemen und der Durchführung von Nachuntersuchungen. Die Alarm-Historie dient somit als wertvolles Werkzeug zur Überwachung und Auswertung der Systemleistung und unterstützt bei der kontinuierlichen Verbesserung der Anlagen- oder Maschineneffizienz.

Triggered	Acknowledged	Alarm text
4/29/2024 11:28 AM		Interlock occurred in "HwZone1 Sequence1"

alarm count: 1

HMI

Um neben den TCMZ und CMZ auch Warnungen und Interlocks oder System Fehler anzuzeigen, ist es erforderlich, die Option "Show warning and interlock messages" und oder "Show system alarms" zu aktivieren. Durch Aktivierung dieser Option werden alle Warnungen und Interlocks und oder Systemfehler in der Benutzeroberfläche angezeigt.

Selmo powered HMI - codesys_test_2024.4

Selmo  4/29/2024 11:28:20 AM Interlock occurred in "HwZone1 | Sequence1"! 09:09:52
Tuesday, April 30, 2024 

Entries: 1

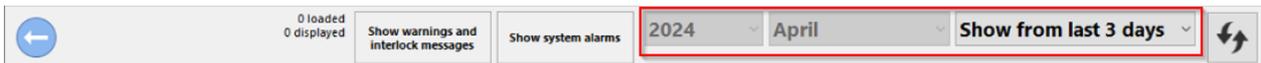
Triggered	Acknowledged	Deactivated	Alarm text
4/29/2024 11:28:20 AM	4/29/2024 11:49:46 AM		Interlock occurred in "HwZone1 Sequence1"
4/29/2024 11:28:20 AM			Interlock occurred in "HwZone1 Sequence1"

3 loaded
3 displayed

Show warnings and interlock messages Show system alarms 2024 April Show from last 3 days 

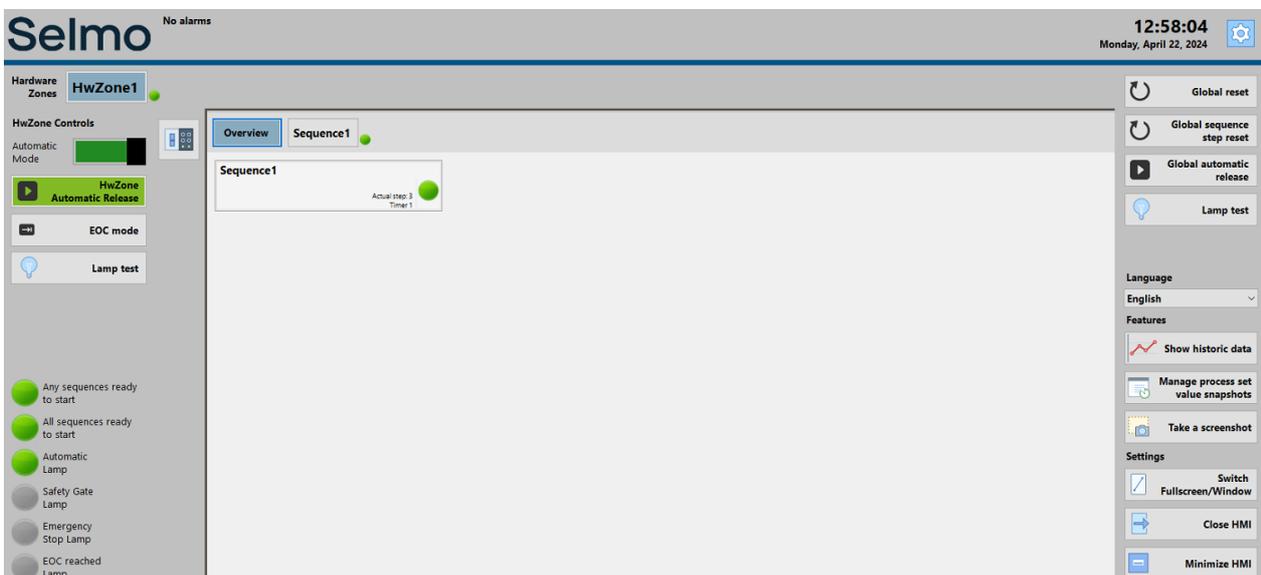
HMI

Es besteht die Möglichkeit, die Alarmer nach Jahr, Monat oder für die letzten 3, 7 oder 30 Tage abzurufen. Dadurch kann das Bedienpersonal gezielt auf vergangene Alarmereignisse zugreifen und spezifische Zeiträume auswählen, um Alarmmuster, Trends oder besondere Vorkommnisse zu analysieren. Die flexible Abrufmöglichkeit der Alarmer nach verschiedenen Zeitintervallen ermöglicht eine detaillierte Untersuchung der Alarmhistorie und erleichtert die Fehlerdiagnose, die Performance-Analyse und die kontinuierliche Verbesserung des Systems.



HMI

1.9.3 Plant



HMI

Global Reset

Ein globales Quittieren von anstehenden Fehlern.

Global sequence step reset

Setzt alle Sequences auf Schritt 1 zurück.

Global automatic release

Freigeben aller Sequences, welche "Ready to start" sind.

Muss für die definierte Zeit des Startup-Delay, default 3s, gedrückt werden.

Lamp test

Kontrolle aller Status Lampen auf der Plant Ebene.

Ist in der Hardware Zone genauer beschrieben.

Language

Sprachumschaltung, mit dem im Selmo-Studio definierten Sprachen, aus dem Dropdown-Menü.

Show historic data

Diese Funktion ermöglicht es, historische Daten anzuzeigen. Dies ist nur dann möglich wenn man Parameter auf der Plant-, Hardwarezone-, und Sequence Ebene anlegt. Die Parameter welche angezeigt werden sollen, müssen von der rechten Seite auf die linke gezogen werden.

The screenshot shows the Selmo HMI interface. At the top, it displays 'No alarms', the time '09:48:52', and the date 'Tuesday, May 7, 2024'. The main area is titled 'Hardware Zones HwZone1'. On the left, there are four parameter configuration panels, each with a 'Y-Axis' dropdown set to 'Primary', a 'Linecolor' selection, and a 'Line thickness' of 1. The central chart area shows a grid with time on the x-axis (from 07.05.24 00:00:00 to 09:46:30) and parameter values on the y-axis. On the right, a tree view shows the hierarchy: Plant > HwZone1 > Sequence1 > Parameter1, Parameter2. The right sidebar contains a 'Language' dropdown set to 'English' and a 'Show historic data' button.

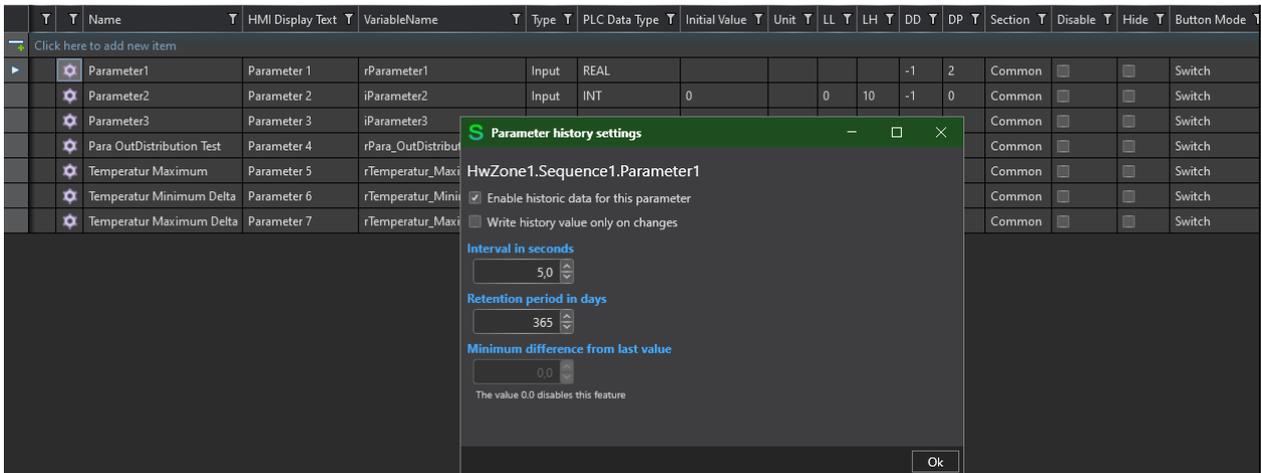
Um diese Funktion nutzen zu können muss im Selmo Studio bei den 'Parameters' ein Doppelklick auf das Zahnradsymbol getätigt werden.

The screenshot shows the 'Parameters' table in Selmo Studio. The table has the following columns: Name, HMI Display Text, VariableName, Type, PLC Data Type, Initial Value, Unit, LL, LH, DD, DP, Section, Disable, Hide, and Button Mode. The rows are:

Name	HMI Display Text	VariableName	Type	PLC Data Type	Initial Value	Unit	LL	LH	DD	DP	Section	Disable	Hide	Button Mode
Parameter1	Parameter 1	rParameter1	Input	REAL					-1	2	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Parameter2	Parameter 2	iParameter2	Input	INT	0		0	10	-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Parameter3	Parameter 3	iParameter3	Input	INT	0		0	10	-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Para OutDistribution Test	Parameter 4	rPara_OutDistribution_Test	Input	REAL					-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Temperatur Maximum	Parameter 5	rTemperatur_Maximum	Input	REAL	80				-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Temperatur Minimum Delta	Parameter 6	rTemperatur_Minimum_Delta	Input	REAL	5				-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Temperatur Maximum Delta	Parameter 7	rTemperatur_Maximum_Delta	Input	REAL	6				-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch

A red box highlights the gear icon in the first column for 'Parameter1' through 'Parameter7'.

Das Zahnrad Symbol öffnet ein separates Fenster. Hier ist es wichtig, dass 'Enable historic data for this parameter' angehakt wird, damit der Parameter in 'Show historic data' in der HMI nutzbar und sichtbar ist. Bei dem Fenster hat man auch die Möglichkeit die 'Interval in seconds' Zeit auszuwählen und auch die Verweildauer kann ausgewählt werden. Die Minstdifferenz zum letzten Wert kann auch ausgewählt werden.



Nachdem die Einstellungen gesetzt wurden wird das Zahnrad Symbol grün eingefärbt und zeigt somit an, dass es für 'Show historic data' in der HMI nutzbar ist.

Name	HMI Display Text	VariableName	Type	PLC Data Type	Initial Value	Unit	LL	LH	DD	DP	Section	Disable	Hide	Button Mode
Parameter1	Parameter 1	rParameter1	Input	REAL					-1	2	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Parameter2	Parameter 2	iParameter2	Input	INT	0	0	10	-1	0		Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Parameter3	Parameter 3	iParameter3	Input	INT	0	0	10	-1	0		Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Para OutDistribution Test	Parameter 4	rPara_OutDistribution_Test	Input	REAL					-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Temperatur Maximum	Parameter 5	rTemperatur_Maximum	Input	REAL	80				-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Temperatur Minimum Delta	Parameter 6	rTemperatur_Minimum_Delta	Input	REAL	5				-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch
Temperatur Maximum Delta	Parameter 7	rTemperatur_Maximum_Delta	Input	REAL	6				-1	0	Common	<input type="checkbox"/>	<input type="checkbox"/>	Switch

Manage process set value snapshots

Bietet die Möglichkeit, alle Parameter in der HMI, abzuspeichern und wiederherzustellen.

Take a screenshot

Screenshot der aktuellen Ansicht von der HMI erstellen.

Switch Fullscreen/Window

Umschalten zwischen Fullscreen und Fenster-Modus der HMI.

Close HMI

HMI-Anwendung schließen.

Minimize HMI

Minimieren der HMI in die Taskleiste.

Plant Parameter

Hier werden alle Parameter die in der Plant definiert wurde angezeigt.

The screenshot displays the Selmo HMI interface. At the top, the title bar reads "Selmo powered HMI - Selmo in Use". The main header area includes the "Selmo" logo, "No alarms" status, the time "12:28:57", the date "Tuesday, December 19, 2023", and a red-bordered icon representing the Plant Parameter function. Below the header, the interface is divided into several sections. On the left, under "Hardware Zones", "HwZone1" is selected. The main area shows "Overview", "Sequence1", and "Sequence2" tabs. Under "Sequence1", there is a status indicator "Actual step: 1 Step 1" with a warning icon. Under "Sequence2", there is a status indicator "Actual step: 1 Step 1". On the right side, the "Parameters" section displays "Parameter 1" with a value of "0".

HMI

Selmo solution

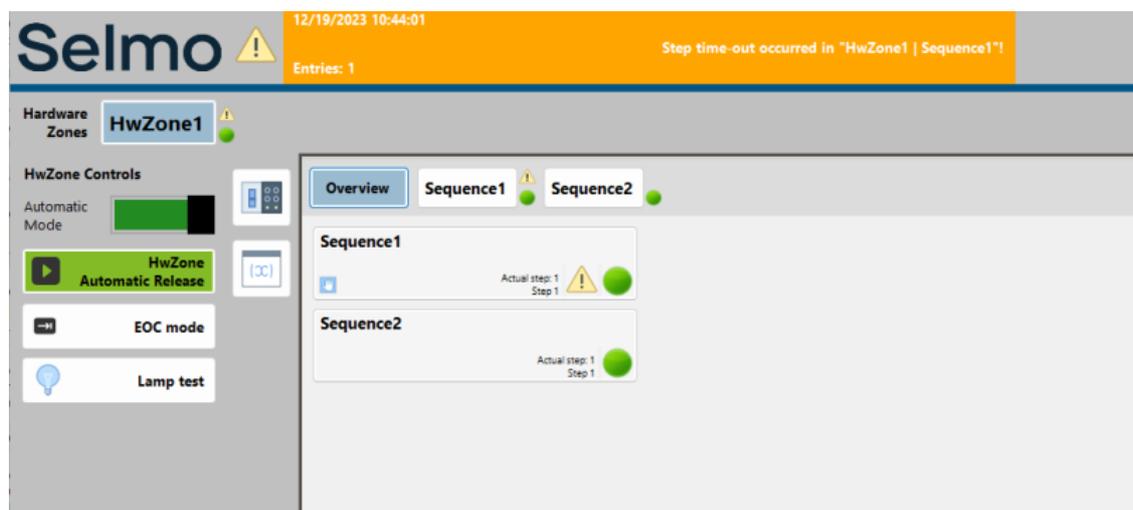
Hardware Zone

2 Hardware Zone

In der Übersicht werden alle Sequences der entsprechenden Hardwarezone dargestellt. Jede Sequence hat ein eigenes Informationsfenster, in der der aktuelle Schritt, der Status (Betriebsart, Alarm) und der Timeout angezeigt wird.



Das Warnzeichen symbolisiert ein Timeout, einen Einzelschritt und den EOC-Modus, es zeigt an dass ein manueller Eingriff benötigt wird.



HMI

Sequence Status-Lamps

Grün: Die Sequence läuft im Automatik-Betrieb.

Rot: Die Sequence hat eine Störung.

Grau: Handbetrieb ist aktiv.



Hardwarezone Control

A screenshot of the Hardwarezone Control interface. At the top, it shows "Hardware Zones" with a blue box labeled "HwZone1" and a green indicator. Below this is the "HwZone Controls" section, which includes a slider for "Automatic Mode" (set to green), a "HwZone Automatic Release" button with a play icon, an "EOC mode" button with a right arrow icon, and a "Lamp test" button with a lightbulb icon. At the bottom, there is a legend for status lamps: a green circle for "Any sequences ready to start", a green circle for "All sequences ready to start", a green circle for "Automatic Lamp", a grey circle for "Safety Gate Lamp", a grey circle for "Emergency Stop Lamp", and a grey circle for "EOC reached Lamp".

Hardware Zones **HwZone1**

HwZone Controls

Automatic Mode

HwZone Automatic Release

EOC mode

Lamp test

- Any sequences ready to start
- All sequences ready to start
- Automatic Lamp
- Safety Gate Lamp
- Emergency Stop Lamp
- EOC reached Lamp

Lamp test

Der Lamp test ist eine Standardfunktion und beim Durchführen des Lamp tests wird die Funktion der Leuchten kontrolliert. Nachdem der Button geklickt wurde, sollten alle Status Lampen, welche in den Sequenzen verwendet wurden (Automatic Lamp, Saftey Gate Lamp, Emergency Stop Lamp und EOC reached Lamp) leuchten.

Der Lamp test ist auf der Hardware Zone Ebene und der Plant Ebene durchführbar. Bei der Plant Ebene leuchten alle Lampen.

Selmo powered HMI - codesys_test_2024.4

Selmo

No alarms

11:56:36
Monday, April 22, 2024

Hardware Zones **HwZone1**

HwZone Controls

Automatic Mode

HwZone Automatic Release

EOC mode

Lamp test

Any sequences ready to start

All sequences ready to start

Automatic Lamp

Safety Gate Lamp

Emergency Stop Lamp

EOC reached Lamp

Overview **Sequence1**

Sequence Automatic Release

Step decrement Step next valid Step increment

Single Step On Step Single Ghost Mode Step Reset

Parameters

Parameter Timer 1 **3 s**

Previous step
2: Step 2

Actual step
3: Timer 1

Next step
4: Step 3

Waiting for
Timer 1 Timer
Time left: 1.5s

Global reset

Global sequence step reset

Global automatic release

Lamp test

Language
English

Features

Show historic data

Manage process set value snapshots

Take a screenshot

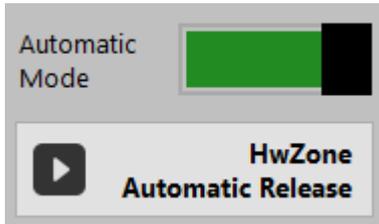
Settings

Switch Fullscreen/Window

Close HMI

Minimize HMI

Betriebsart Anwahl



Automatik Mode

Der Automatik-Modus bezeichnet einen Betriebszustand, in dem ein System, eine Maschine oder ein Gerät automatisch arbeitet, ohne dass menschliche Eingriffe oder Kontrolle erforderlich sind. Im Automatik-Modus führt das System vorprogrammierte Abläufe, Prozesse oder Sequences eigenständig aus, basierend auf voreingestellten Parametern, Algorithmen oder Regeln. Dadurch werden Aufgaben automatisiert und effizient erledigt, ohne dass kontinuierliche menschliche Überwachung oder Eingriffe erforderlich sind.

Der grüne Balken zeigt an, dass der Automatik-Modus aktiviert wurde. In diesem Modus arbeitet das System eigenständig gemäß den voreingestellten Parametern oder Regeln.

Der graue Balken zeigt an, dass der Handbetrieb gewählt wurde. In diesem Modus kann die Maschine oder Funktion manuell durch menschliche Eingriffe gesteuert werden.

Automatik Release

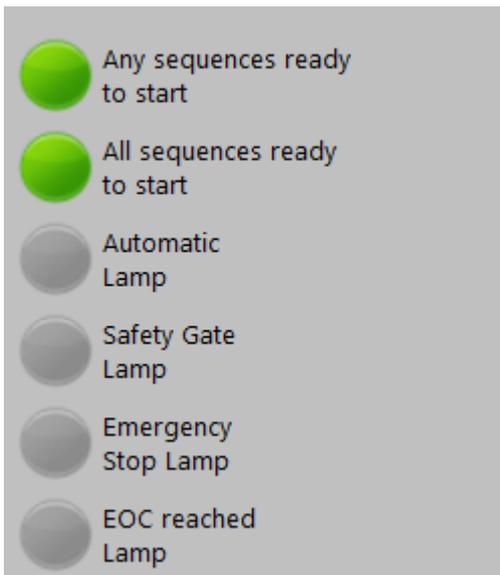
Der "Automatik Release"-Button ermöglicht die Freigabe der Betriebsart "Automatik". Wenn dieser Button für mindestens 3 Sekunden (Einstellbare Zeit) betätigt wird, wird das System in den Automatik-Modus versetzt, wodurch es in der Lage ist, automatisch und eigenständig zu arbeiten. Dabei führt das System vordefinierte Abläufe, Sequences oder Funktionen aus, ohne dass ein Benutzer kontinuierlich eingreifen oder Anweisungen geben muss.

EOC-Mode

Durch Aktivieren des "End of Cycle" Modus werden die Sequences bis zum EOC-Schritt ausgeführt und anschließend wird der Automatik-Betrieb angehalten.



Hardwarezone Status-Lamps



Any sequences ready to start

Gibt an, ob eine Sequence in der Hardwarezone bereit ist, um den Automatik-Betrieb zu starten.

All sequences ready to start

Gibt an, ob alle Sequences in der Hardwarezone bereit sind, um den Automatik-Betrieb zu starten.

Automatic Lamp

Blinken bedeutet, dass mindestens eine Sequence bereit ist, um zu starten.

Dauerleuchten zeigt an, dass die Automatik aktiv ist.

Safety Gate Lamp

Die Safety-Gate-Lampe signalisiert, ob der Safety-Gate-Schalter aktiviert ist.

Emergency Stop Lamp

Die Emergency Stop Lampe signalisiert, ob ein Not-Halt aktiviert ist.

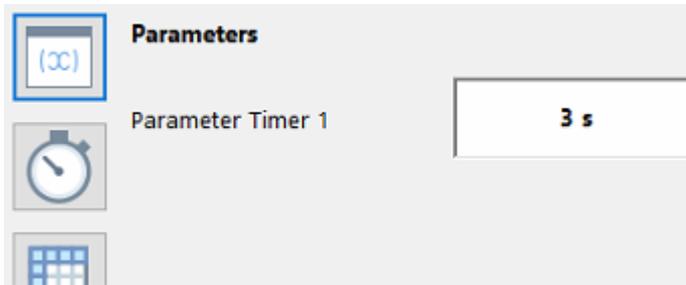
EOC reached Lamp

Die EOC-reached-Lampe blinkt, wenn der EOC-Modus aktiviert wurde.

Wenn die EOC-reached-Lampe dauerhaft leuchtet, bedeutet dies, dass alle Sequences in der Hardwarezone den EOC-Schritt erreicht haben.

Hardwarezone Parameter

Hier werden alle für die Hardwarezone definierten Parameter angezeigt



- H -

HMI Button 91