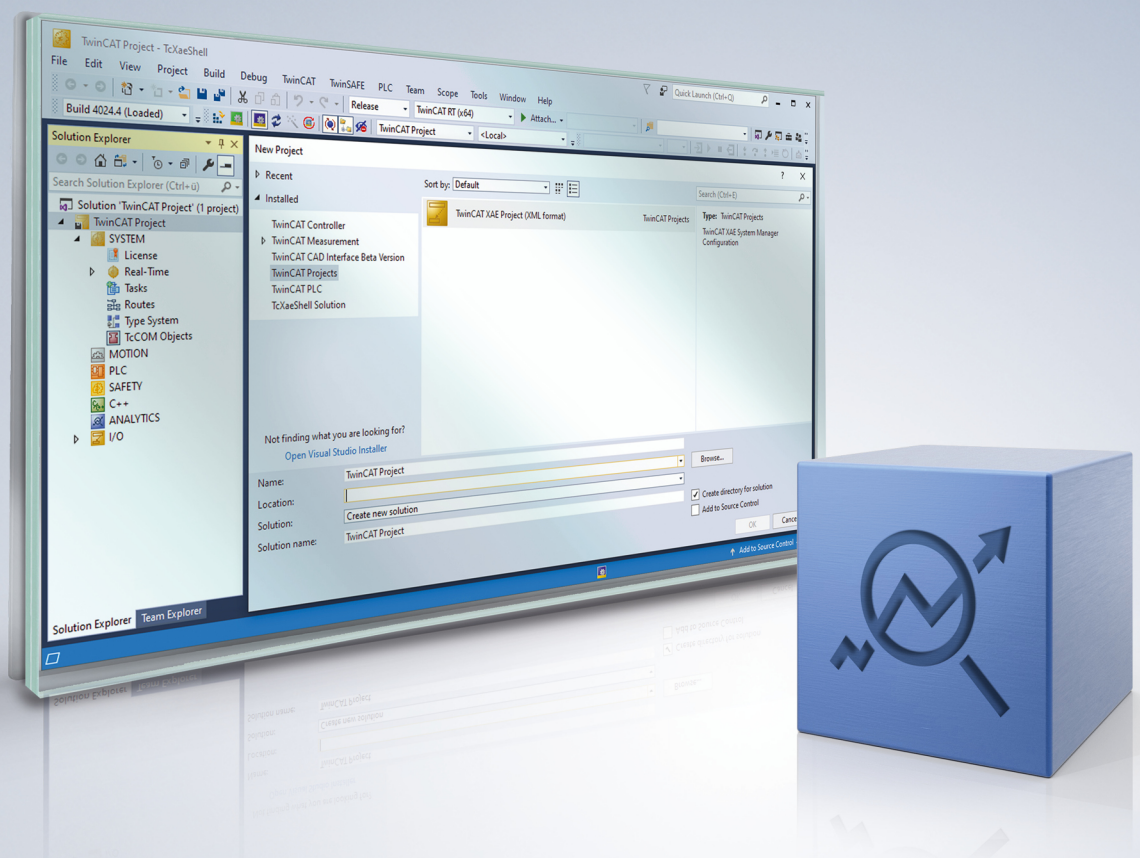
BECKHOFF New Automation Technology

Handbuch | DE

TE3500

TwinCAT 3 | Analytics Workbench



10.07.2025 | Version: 1.12.1

Inhaltsverzeichnis

1 Vorwort ... 5
1.1 Hinweise zur Dokumentation ... 5
1.2 Zu Ihrer Sicherheit ... 6
1.3 Hinweise zur Informationssicherheit ... 7
2 Übersicht ... 8
3 Installation ... 11
3.1 Systemanforderungen ... 11
3.2 Installation und Lizenzierung ... 11
3.3 Lizenzierung ... 15
4 Analytics Workflow - First Steps ... 18
4.1 Aufzeichnung von Maschinendaten ... 18
4.2 Kommunikation ... 21
4.3 Historisierung von Daten ... 22
4.4 Analyse der Daten ... 29
4.5 24-stündige Anwendung von Analytics ... 34
5 Technische Einführung ... 44
5.1 Grundkonzept ... 44
6 Konfiguration ... 46
6.1 Datenquellen ... 46
6.1.1 Wizard ... 48
6.2 Data Scout ... 54
6.2.1 Data Track Editor ... 54
6.3 Editor Basics ... 68
6.3.1 Netzwerke ... 68
6.3.2 Smart Link ... 72
6.3.3 Enable Disable ... 76
6.3.4 Static Values ... 78
6.3.5 Globale Parameter ... 79
6.4 Algorithmen ... 81
6.4.1 Analytics - Base ... 82
6.4.2 Analytics - Classification ... 137
6.4.3 Analytics - Clustering ... 163
6.4.4 Analytics - Compare ... 172
6.4.5 Analytics - Energy ... 196
6.4.6 Analytics - Math ... 202
6.4.7 Analytics - Preprocessing ... 216
6.4.8 Analytics – Reporting ... 218
6.4.9 Analytics - Specific ... 222
6.4.10 Analytics - Statistics ... 233
6.4.11 Analytics - Training Base ... 259
6.4.12 Analytics - Visualization Only ... 261
6.4.13 Analytics - XY Path Analysis ... 269
6.4.14 Analytics - Erweiterung der Algorithmik ... 279

6.5 Interaktion mit Scope ... 356
6.5.1 Scope Konfiguration im Netzwerk Template ... 360
6.6 Arbeit mit historischen Daten ... 363
6.7 Runtime-Deployment ... 365
6.7.1 Algorithmuseigenschaften ... 372
6.7.2 SPS-Code ... 374
6.8 HMI One-Click Dashboard ... 433
6.8.1 Dashboard Aufbau und Inhalte im Analytics Projekt verwalten ... 436
6.8.2 Angepasste und eigene Controls verwenden ... 446
6.8.3 Benutzerverwaltung und Zugriffsrechte konfigurieren ... 465
6.8.4 Dashboard-Konfiguration ... 468
6.8.5 Ändern eines generierten Dashboards im HMI-Engineering ... 475
6.8.6 Konfiguration des Dashboards zur Laufzeit im Webbrowser ... 475
6.8.7 Erzeugen von Diagrammen im Webbrowser ... 482
6.8.8 Umschaltung mehrerer Maschinen im HMI Dashboard ... 507
6.8.9 Integration einer Sprachumschaltung ... 513
6.8.10 Verwenden der Interactive Funktionen in Standard TwinCAT HMI-Projekten ... 516
6.9 Reporting Integration ... 518
6.9.1 Grundkonzept ... 518
6.9.2 Analytics Reporting ... 520
6.9.3 TwinCAT Reporting Server ... 528
6.10 Vergleich von Analyseergebnissen ... 539
6.10.1 Laden und Speichern von Projektkonfigurationen ... 542

7 API ... 544
7.1 Data Exchange API ... 544
7.1.1 .NET ... 544
7.1.2 Python ... 552

8 Beispiele ... 554
8.1 Pick-and-place-Anwendung ... 554
8.2 Pick-and-place-Anwendung: DenStream ... 563
8.3 Pick-and-place-Anwendung: Korrelation ... 565

9 Anhang ... 570
9.1 FAQ – Häufig gestellte Fragen und Antworten ... 570

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.
Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.
Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.
Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.
Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.
Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Kennzeichnungen führen.

EtherCAT® ist eine eingetragene Marke und patentierte Technologie, lizenziert durch die Beckhoff Automation GmbH, Deutschland.

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.
Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.
Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

Fremdmarken

In dieser Dokumentation können Marken Dritter verwendet werden. Die zugehörigen Markenvermerke finden Sie unter: https://www.beckhoff.com/trademarks.

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

⚠ GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

⚠ WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

⚠ VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt

Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem https://www.beckhoff.de/secguide.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter https://www.beckhoff.de/secinfo.

2 Übersicht

Die TwinCAT 3 Analytics Workbench ist ein TwinCAT 3 Engineering-Produkt zur Erstellung von kontinuierlichen Datenanalysen aus verschiedenen, dezentralen Maschinensteuerungen. Die Konfiguration der Workbench ist in Microsoft Visual Studio® integriert und als grafische Benutzeroberfläche angelegt. In einer Toolbox stehen viele Algorithmen, wie Zykluszeitüberwachung, Life-Count, Lebensdauer und Minimum/Maximum/Mittelwert, zur Konfiguration der Analyse zur Verfügung.

Zur einfachen Visualisierung der Signalverläufe enthält die TwinCAT 3 Analytics Workbench das TwinCAT 3 Scope View Professional TE1300: Die Analyseergebnisse kann der Benutzer per Drag-and-Drop aus dem Analytics-Konfigurator in das Charting-Tool ziehen, um signifikante Stellen im Datenstrom zu markieren. Solche Markierungen können einfache Minima und Maxima, Zählerstände, aber auch Ergebnisse eines logischen Operators sein, der Ergebnisse aus der Maschinensteuerung logisch miteinander verbindet, so dass man sie im Datenstrom wiederfinden kann. Damit ist eine Korrelation zu anderen Signalen im Scope View zyklusgenau möglich.

Die MQTT-Eingangsdaten werden über den TwinCAT Target Browser ausgewählt, wobei Live-Daten und über den TF3520 TwinCAT 3 Analytics Storage Provider auch historische Daten zur Verfügung stehen. Ist die erstellte Analyse vollständig und im grafischen Editor getestet, kann diese Konfiguration mit nur einigen Klicks in lesbaren SPS-Code mit zugehörigem HTML5-Dashboard umgewandelt werden. Der automatisch generierte SPS-Code und das ebenfalls automatisch erstellte Dashboard können direkt auf ein Gerät mit TF3550 TwinCAT 3 Analytics Runtime heruntergeladen werden und dort 24/7 parallel zu den eigentlichen Produktionsmaschinen laufen und Analyseergebnisse liefern. Die Darstellung über die TwinCAT 3 HMI zeigt Analyseergebnisse beispielsweise für Maschinenführer, Fertigungsleiter und/oder Maschinenbauer.

Funktionalität

Im Vergleich mit dem Analytics Service Tool bietet die Analytics Workbench ergänzende Funktionen u.a. mit der Code- und Dashboard-Generierung. Alle Funktionen des Service Tools sind auch in der Workbench enthalten.

Komponenten

- Analytics-Konfigurator
- Base Analytics-Algorithmen
- Analytics PLC Library
- Analytics Storage Provider Recorder inkl. SPS-Bibliothek
- TwinCAT Scope (TE1300 und TF3300)
- IoT-Konnektivität
- HMI Engineering (TE2000)

Features	TE3500 Analytics Workbench		TE3520 Analytics Service Tool	
	7-Days-Trial	Full License	7-Days-Trial	Full License
General:				
Analysis configurator	✓	✓	✓	✓
Analysis channels	max 5	unlimited	max 5	unlimited
Analysis moduls/ algorithm	max 3	unlimited	max 3	unlimited
Long time records >1h	✗	✓	✗	✓
Interaction with Scope View	✓	✓	✓	✓
Storage Provider Recorder	✓	✓	✓	✓
MQTT	✓ (max 5)	✓	✓ (max 5)	✓
Analytics File	✓ (max 5)	✓	✓ (max 5)	✓
ADS	✓ (no auto deployment)	✓ (no auto deployment)	✓	✓
Data export tool	✓	✓	✓	✓
Basic data export formats	✓ (max 5)	✓	✓ (max 5)	✓
Extended data export formats	✗	✓	✗	✓
Analytics Data Scout:				
Load data	✓	✓	✓	✓
Toolbar snipping buttons	✗ *	✓	✗ *	✓
Export data to Analytics File	✓	✓	✓	✓
Deploy Runtime:				
Deploy Wizard	✓	✓	/	/
Add PLC Code to existing Sln	✗	✓	/	/
Merge PLC Code	✗	✓	/	/
Auto Create Bootproject	✗	✓	/	/
Auto Activate Runtime	✗	✓	/	/
Stream Results	✗	✓	/	/
HMI Dashboard	✓ (defaults)	✓	/	/
Reset Algorithm in PLC Code	✗	✓	/	/
Reporting Integration:				
24/7-Reporting with Reporting Algorithm	✓	✓	/	/

Features	TE3500 Analytics Workbench		TE3520 Analytics Service Tool	
On-Demand-Reporting				
Adding Additional Information	✔ (max 1)	✔	✔ (max 1)	✔
Select specific Analytics Information	✘	✔	✘	✔
✔	Full support			
✘	No support			
/	No feature of this product			
*	By existing TE1300 Scope Professional full support			

3 Installation

3.1 Systemanforderungen

Für eine ordnungsgemäße Funktion von TwinCAT Analytics müssen die folgenden Systemanforderungen erfüllt sein.

Unterstützte Betriebssysteme

Windows 10

TwinCAT

Mindestens TwinCAT 3.1 Build 4022.29 für Engineering mit TwinCAT Analytics Service Tool und Workbench.

.NET Framework

Für das Engineering ist ein .NET Framework 4.7.2 erforderlich.

Visual Studio Entwicklungsumgebung

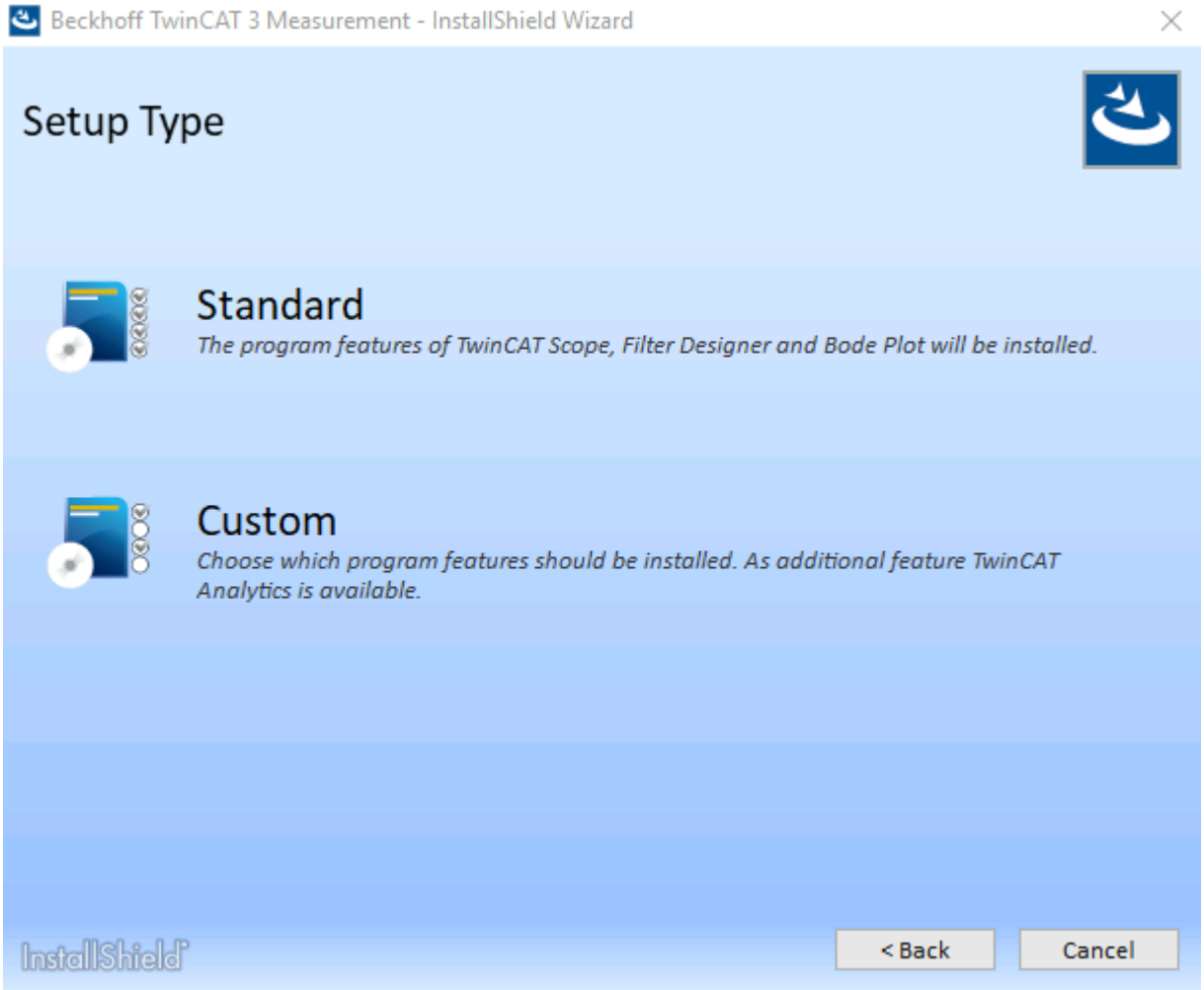
- Visual Studio® 2015
- Visual Studio® 2017
- Visual Studio® 2019
- TwinCAT XAE Shell

Im Allgemeinen ist die Verwendung der Visual Studio® Shell ausreichend. Wenn Sie das Setup „Full“ wählen, wird die TwinCAT XAE Shell automatisch installiert. Das Setup „Update“ bietet lediglich eine Aktualisierung der Analytics-Quellen und keine Visual Studio® Shell.

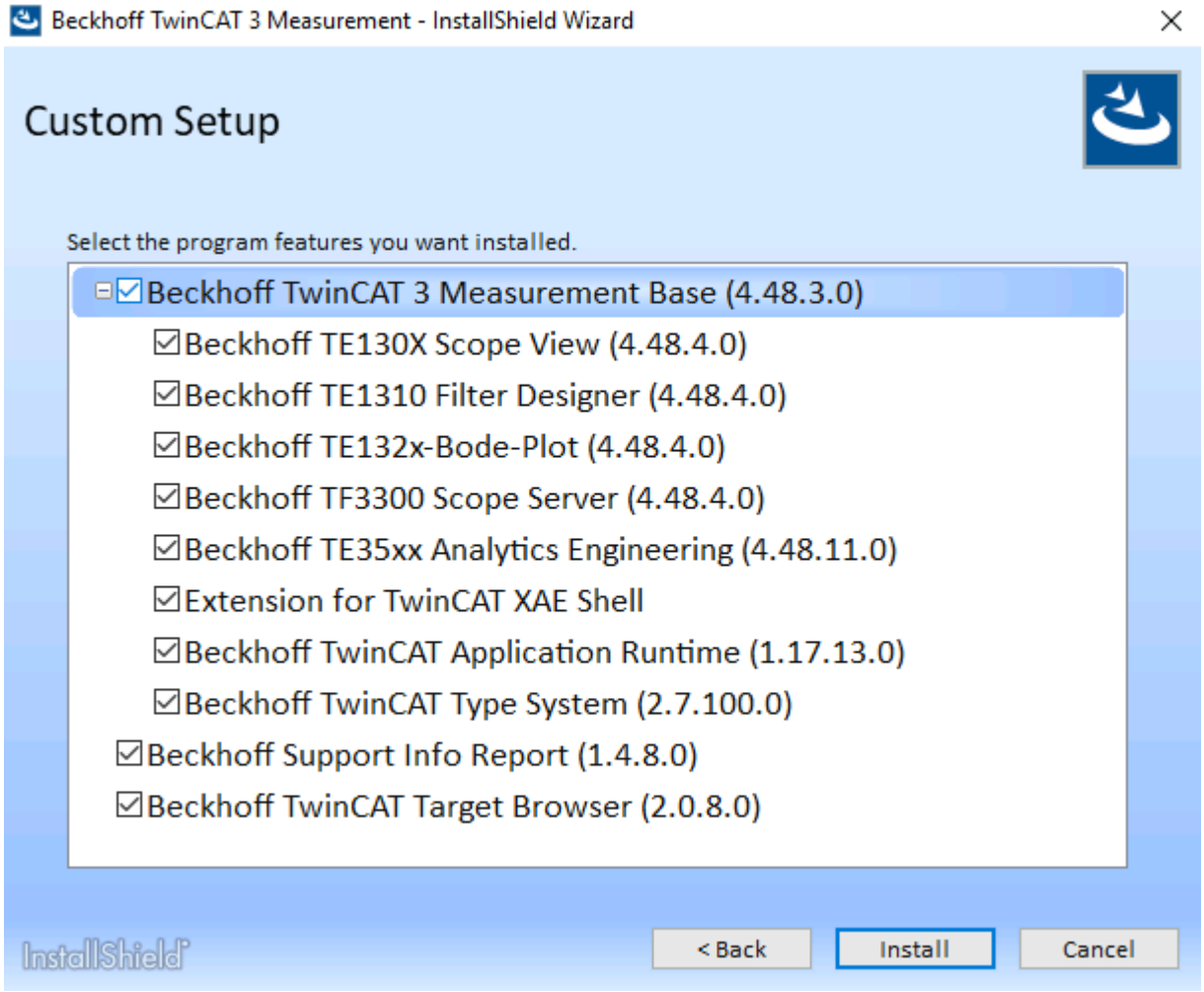
3.2 Installation und Lizenzierung

Das Setup von TwinCAT Analytics ist Teil des Setups der TwinCAT Measurement Suite. Sie haben die Möglichkeit zwischen zwei Setup-Varianten zu wählen:
Das TC3-Measurement-Full Setup bringt die TwinCAT Measurement Komponenten und eine Visual Studio® Umgebung mit.
Das TC3-Measurement-Update Setup bringt dagegen ausschließlich die Measurement Komponenten mit.

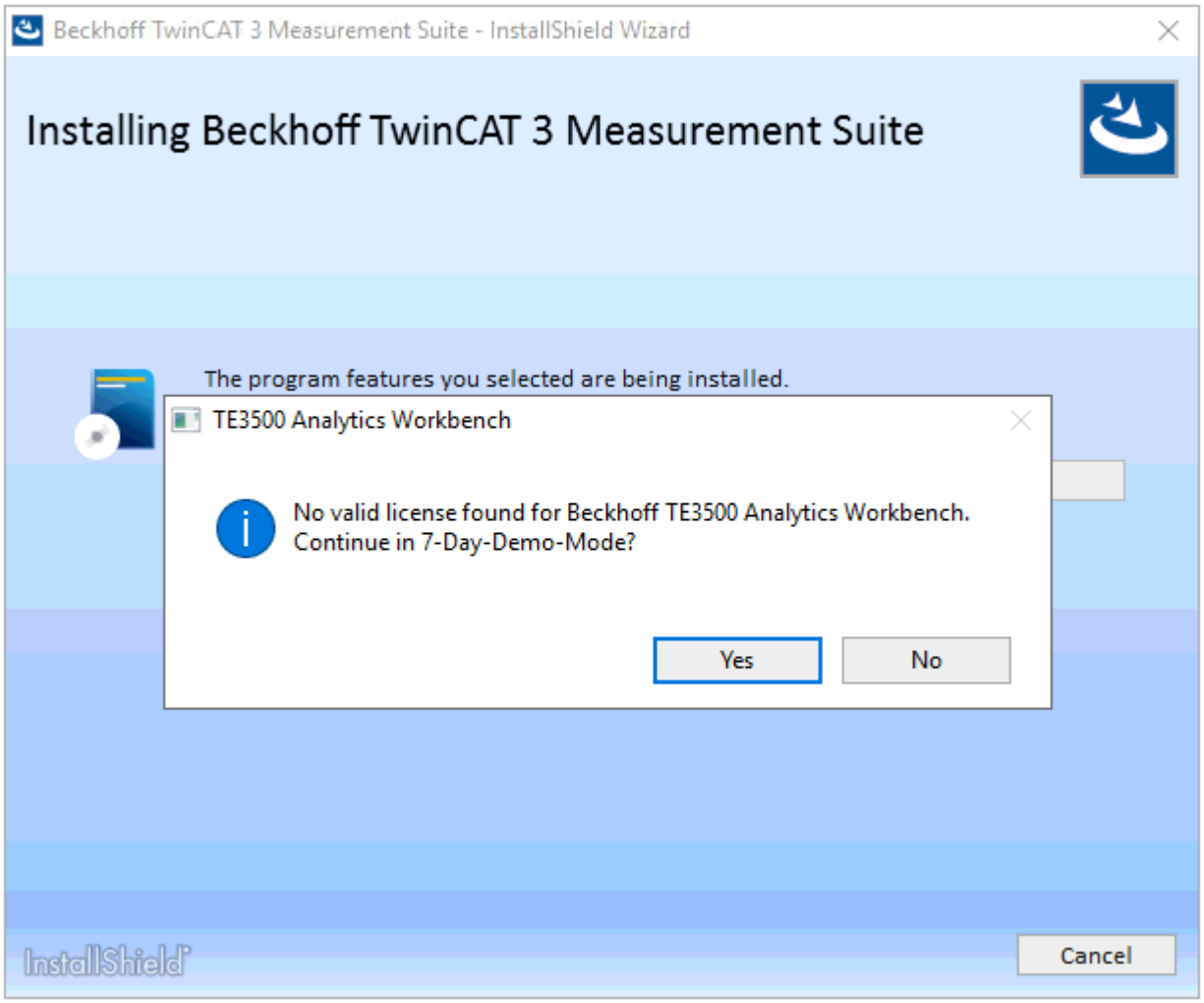
Nach der Bestätigung der Lizenzbedingungen haben Sie die Wahl zwischen Standard und Custom. Bei Standard werden automatisch alle Measurement-Komponenten inklusive TwinCAT Analytics Engineering installiert.



Bei Custom haben Sie die Möglichkeit einzelne Komponenten abzuwählen, damit diese nicht auf Ihrem System installiert werden.



Das Analytics-Setup prüft während des Prozesses, ob Ihr System über die notwendigen Analytics Engineering Lizenzen verfügt. Falls nicht, kann eine Demo-Lizenz aktiviert werden. Diese Demo-Lizenz kann beliebig oft verlängert werden. Allerdings sind nach der Installation im Demo-Modus keine weiteren funktionalen Updates auf diesem System durch neuere Setups mehr möglich. Um diese Updates einspielen zu können, muss zunächst eine Lizenz erworben werden. Eine Übersicht über funktionale Einschränkungen in der Demo-Version finden Sie hier [▶ 9].



Die Lizenzen der Analytics Engineering Tools TE3500 Analytics Workbench und TE3520 Analytics Service Tool sind immer mit einer Wartungslizenz verbunden. Nach dem erstmaligen Erwerb ist die Wartungslizenz für 12 Monate gültig. In diesem Zeitraum sind funktionale Aktualisierungen der Software möglich. Nach dem Ablauf der 12 Monate ist die Software weiter uneingeschränkt nutzbar. Nur neue Funktionen können nicht mehr eingespielt werden, ohne die Wartungslizenz zu verlängern. Die Installation weist darauf mit einer entsprechenden Meldung hin. Eine Verlängerung der Wartungslizenz ist über die Produkte TE3501 und TE3521 jederzeit möglich.

Über zukünftige Funktionen neuerer Versionen erkundigen Sie sich gerne bei Ihrem zuständigen Beckhoff Vertriebsmitarbeiter.

Setup erfordert Lizenz

Updates von TwinCAT Analytics Engineering-Tools sind nur mit einer gültigen Wartungslizenz möglich!

TwinCAT 3-Lizenzen für Nicht-Beckhoff-Geräte

Wenn Sie einen IPC von einem anderen Hersteller als Beckhoff verwenden (TwinCAT 3 Plattformeben >= 90), wird ein TwinCAT 3 Lizenz-Dongle, wenn auch keine Voraussetzung für eine erfolgreiche Lizenzierung von TwinCAT Analytics, nachdrücklich empfohlen!

TwinCAT Analytics Workflow

Um den vollständigen Analytics Workflow zu nutzen, benötigen Sie das TE2000 HMI Engineering-Setup. Dieses ist derzeit nicht in das Analytics Workbench-Setup integriert. Das bedeutet, dass Sie es einzeln installieren müssen. In einer der nächsten Setup-Versionen wird es integriert sein, so dass dieser Schritt dann nicht mehr notwendig sein wird.

3.3 Lizenzierung

Die TwinCAT 3 Function ist als Vollversion oder als 7-Tage-Testversion freischaltbar. Beide Lizenztypen sind über die TwinCAT-3-Entwicklungsumgebung (XAE) aktivierbar.

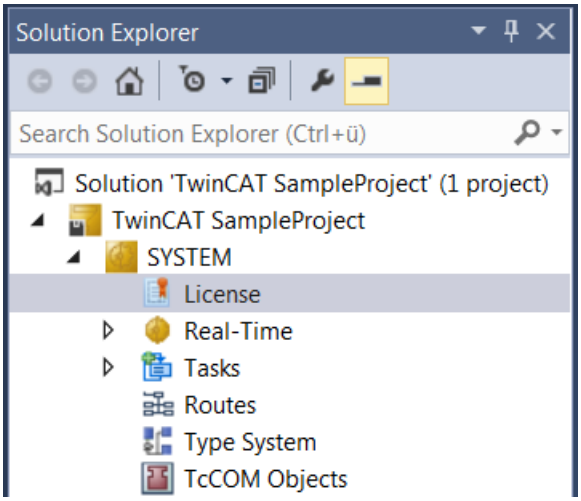
Lizenzierung der Vollversion einer TwinCAT 3 Function

Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „TwinCAT-3-Lizenzierung“.

Lizenzierung der 7-Tage-Testversion einer TwinCAT 3 Function

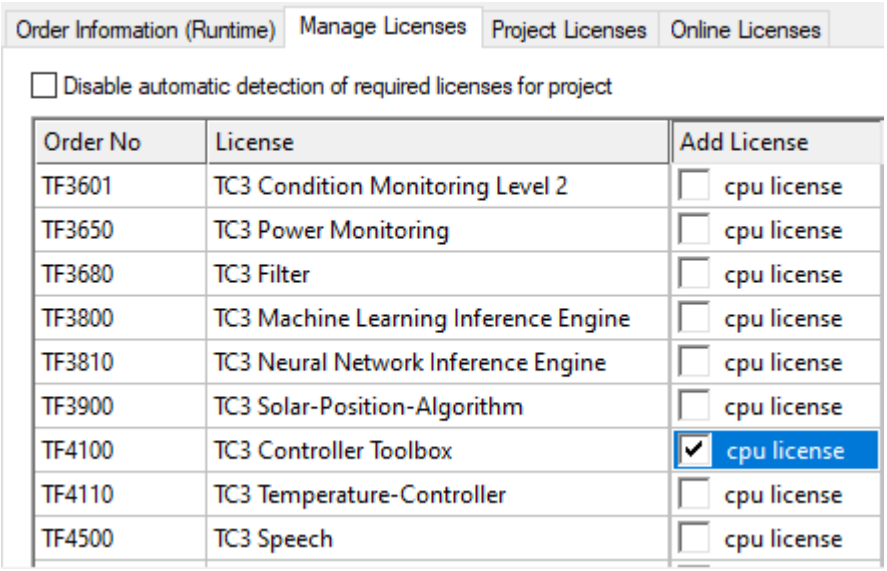
Eine 7-Tage-Testversion kann nicht für einen TwinCAT-3-Lizenz-Dongle freigeschaltet werden.

1. Starten Sie die TwinCAT-3-Entwicklungsumgebung (XAE).
2. Öffnen Sie ein bestehendes TwinCAT-3-Projekt oder legen Sie ein neues Projekt an.
3. Wenn Sie die Lizenz für ein Remote-Gerät aktivieren wollen, stellen Sie das gewünschte Zielsystem ein. Wählen Sie dazu in der Symbolleiste in der Drop-down-Liste **Choose Target System** das Zielsystem aus.
   ⇨ Die Lizenzierungseinstellungen beziehen sich immer auf das eingestellte Zielsystem. Mit der Aktivierung des Projekts auf dem Zielsystem werden automatisch auch die zugehörigen TwinCAT-3-Lizenzen auf dieses System kopiert.
4. Klicken Sie im **Solution Explorer** im Teilbaum **SYSTEM** doppelt auf **License.**

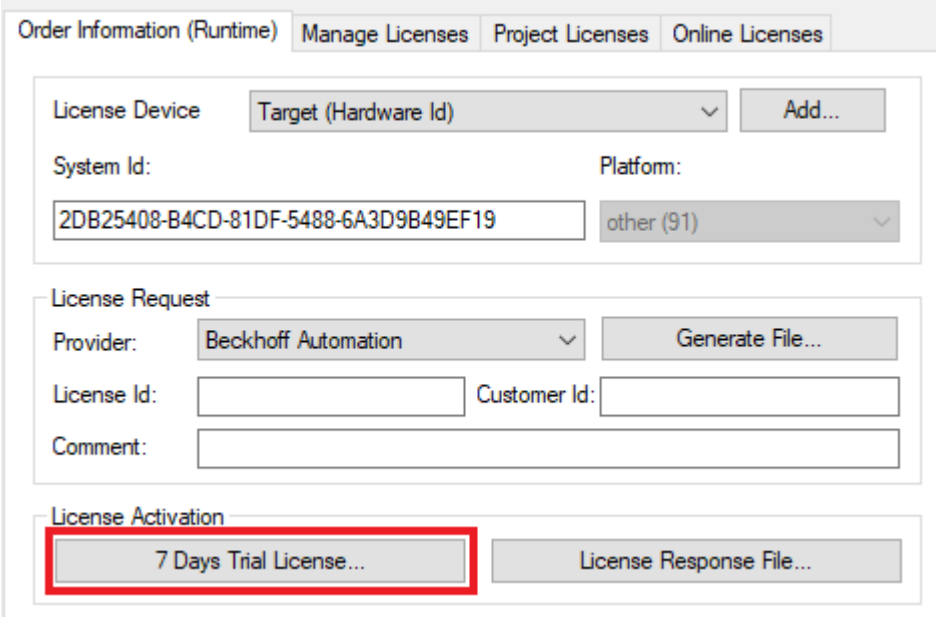


   ⇨ Der TwinCAT-3-Lizenzmanager öffnet sich.

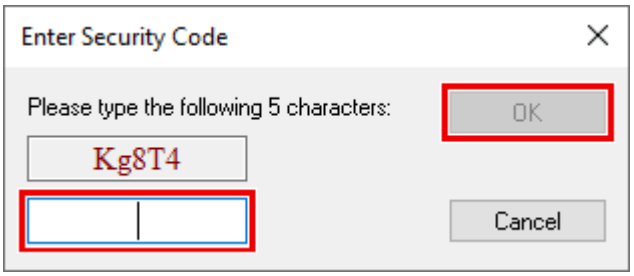
5. Öffnen Sie die Registerkarte **Manage Licenses.** Aktivieren Sie in der Spalte **Add License** das Auswahlkästchen für die Lizenz, die Sie Ihrem Projekt hinzufügen möchten (z. B. „TF4100 TC3 Controller Toolbox“).



6. Öffnen Sie die Registerkarte **Order Information (Runtime).**
   ⇨ In der tabellarischen Übersicht der Lizenzen wird die zuvor ausgewählte Lizenz mit dem Status „missing“ angezeigt**.**
7. Klicken Sie auf **7 Days Trial License...**, um die 7-Tage-Testlizenz zu aktivieren.



   ⇨ Es öffnet sich ein Dialog, der Sie auffordert, den im Dialog angezeigten Sicherheitscode einzugeben.



8. Geben Sie den Code genauso ein, wie er angezeigt wird, und bestätigen Sie ihn.
9. Bestätigen Sie den nachfolgenden Dialog, der Sie auf die erfolgreiche Aktivierung hinweist.

⇨ In der tabellarischen Übersicht der Lizenzen gibt der Lizenzstatus nun das Ablaufdatum der Lizenz an.

10. Starten Sie das TwinCAT-System neu.

⇨ Die 7-Tage-Testversion ist freigeschaltet.

4 Analytics Workflow - First Steps

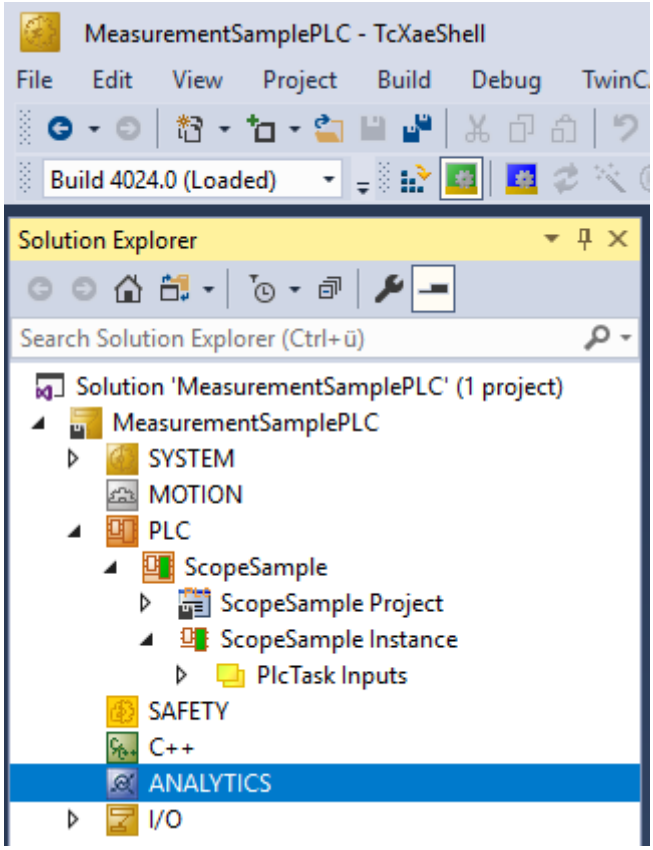
Diese Dokumentation präsentiert Schritt für Schritt den vollständigen TwinCAT Analytics Workflow. Von der Datenerfassung über die Kommunikation und Historisierung bis hin zur Auswertung und Analyse der Daten sowie zur Präsentation der Daten im webbasierten Dashboard.

4.1 Aufzeichnung von Maschinendaten

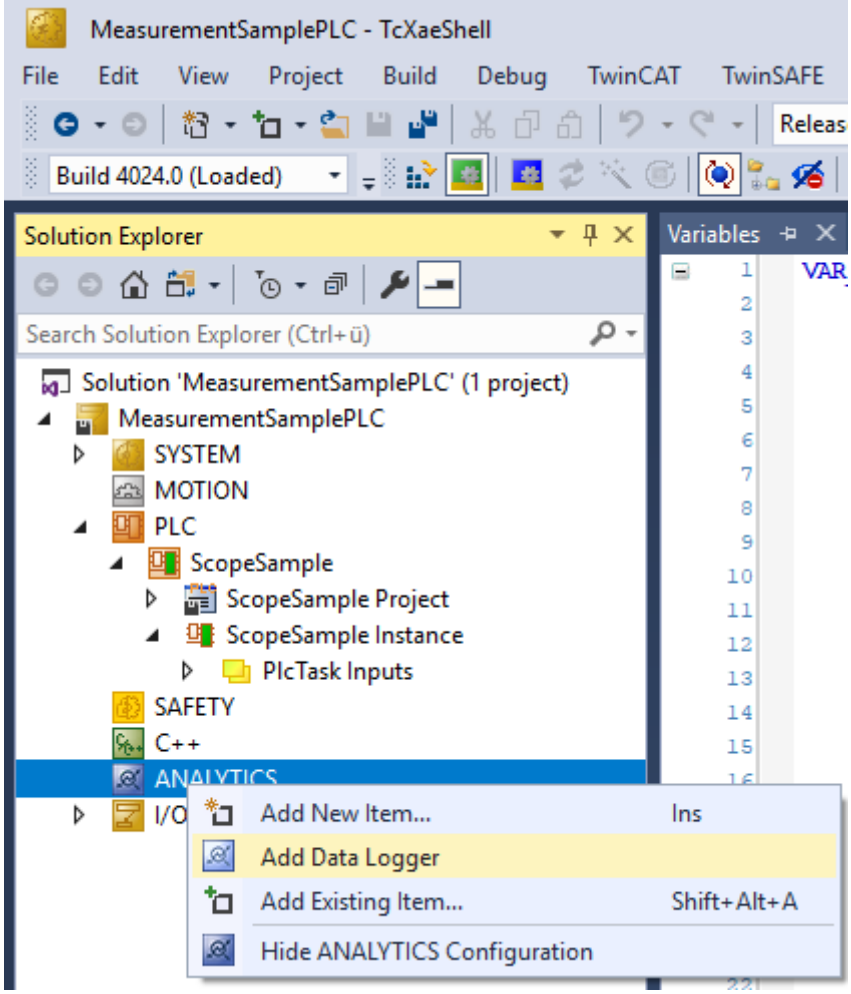
Maschinenseitig ist der Analytics Logger die Aufzeichnungseinrichtung von Prozessdaten des Maschinenabbilds, SPS, NC usw. Der Logger arbeitet im Echtzeitkontext von TwinCAT.

Der TwinCAT Analytics Logger wird mit TwinCAT XAE und XAR installiert. Der Logger kann als MQTT-Client fungieren, um die aufgezeichneten Daten an einen nativen MQTT-Message-Broker zu übermitteln, oder die Daten im selben Datenformat in einer lokalen Binärdatei zu speichern. Bei Verwendung als MQTT-Client kann der Logger kurze Unterbrechungen der Verbindung zum Message-Broker mit Hilfe einer Ringpuffer-Funktionalität überbrücken. Auch für den lokalen Binärdateispeicher kann ein Ringpuffer konfiguriert werden.

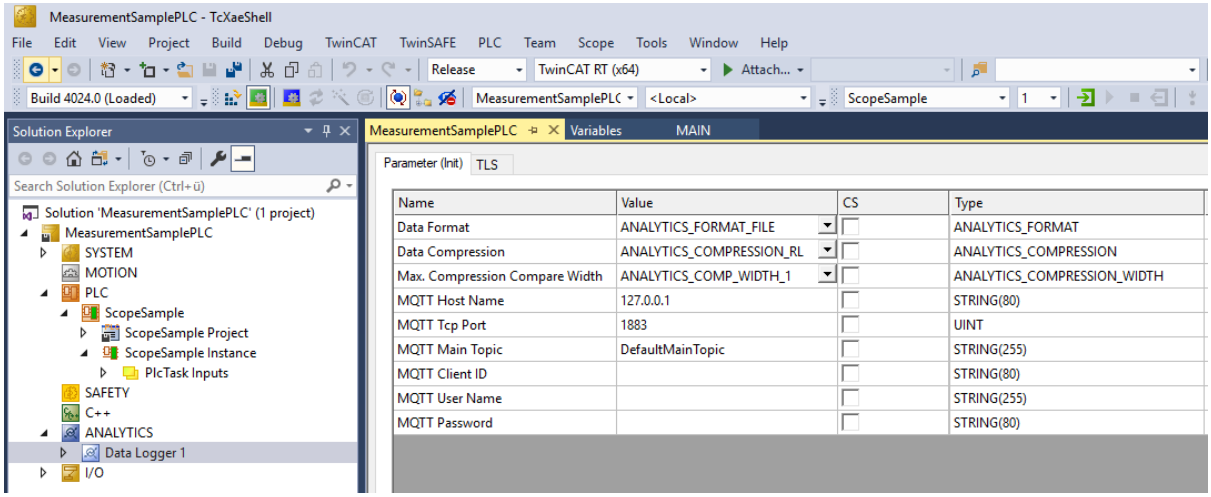
- Um den Analytics Logger zu konfigurieren, müssen Sie in Ihrem vorhandenen TwinCAT-Projekt zum Analytics-Baumknoten navigieren.



- Klicken Sie mit der rechten Maustaste auf diesen Knoten und klicken Sie auf **Add Data Logger**, um Ihrer Konfiguration eine neue Instanz hinzuzufügen.



- Für die Konfiguration der Grundeinstellungen doppelklicken Sie auf das neue Tree Item.



Sie können Ihre spezifischen Analytics Logger-Einstellungen vornehmen.

-Datenformat: Binärdatei oder MQTT-Stream.

-FILE-Format: Der Analytics Logger speichert die Daten in lokalen Binärdateien und alle anderen Einstellungen sind nicht mehr notwendig. Die Dateien werden in *C:\TwinCAT\3.1\Boot\Analytics* gespeichert.

-BINARY-Format: Die Daten werden an den konfigurierten MQTT-Message-Broker gesendet. Sie können mehrere Logger in einem TwinCAT-Projekt vorsehen, um Daten an verschiedene MQTT-Message-Broker zu übermitteln.

-Datenkompression: ein (Standard) oder aus.

-Max. Kompression: Modus der Kompression.

-MQTT-Hostname

-MQTT-Tcp-Port

-MQTT-Maintopic für eigene Hierarchieebenen, damit die Identifikation einfach bleibt.

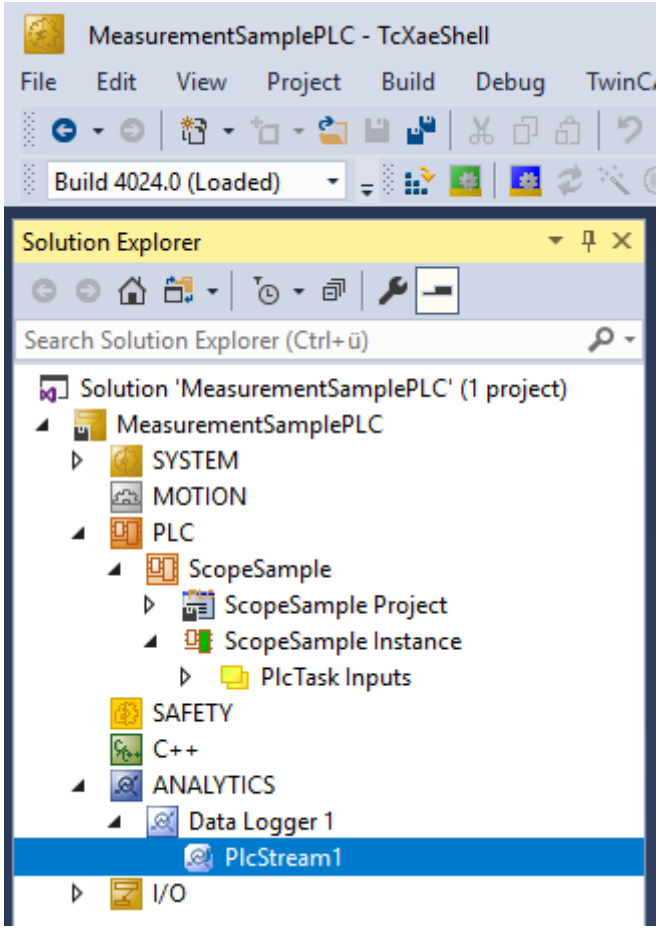
-MQTT-Client-ID, sollte im Netzwerk eindeutig sein.

-MQTT-Benutzername

-MQTT-Passwort für die Authentifizierung beim Message-Broker.

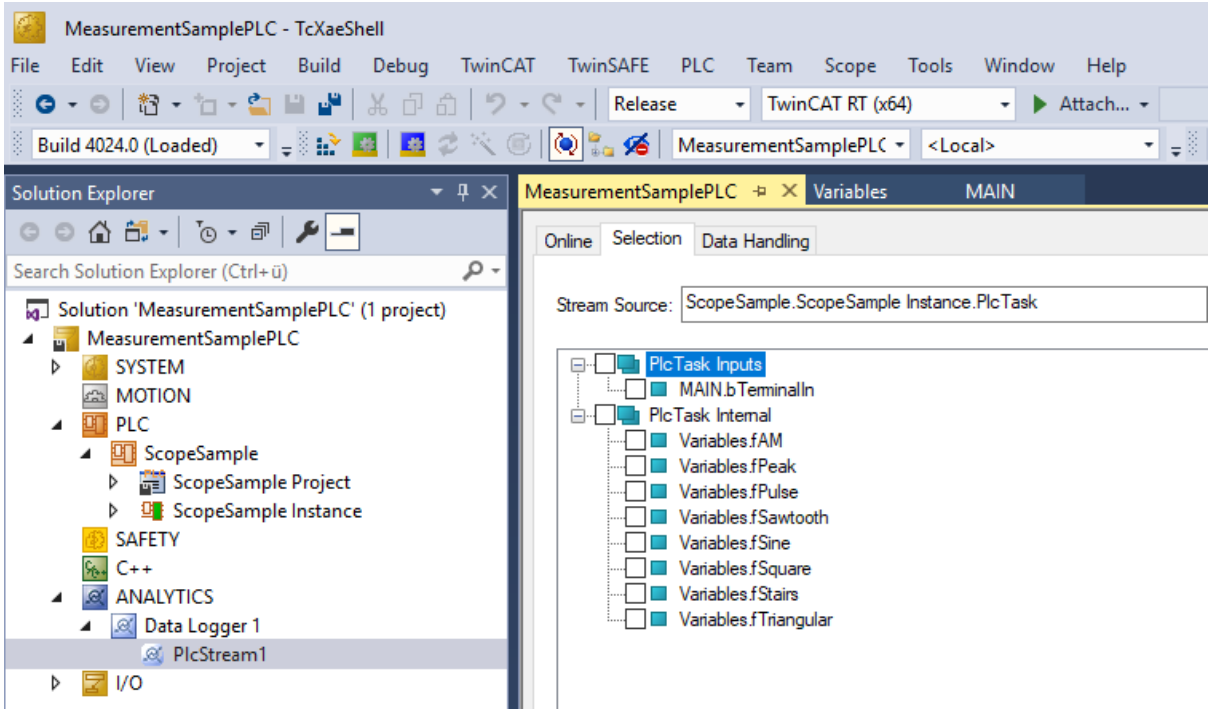
-Auf der Registerkarte **TLS** (Transport Layer Security) können die Sicherheitseinstellungen konfiguriert werden. TLS ist ein sicherer Kommunikationskanal zwischen Client und Server. Bei Verwendung von Zertifikaten ist der TCP-Port 8883 ausschließlich für MQTT über TLS reserviert! Der Analytics Logger unterstützt die Modi CA Certificates, CA Certificates & Client Certificate und Preshared Key (PSK).

- Wenn Variablen in Ihrer SPS-Anwendung in der Deklaration mit dem Attribut {attribute 'TcAnalytics'} gekennzeichnet sind, werden sie automatisch als Stream unter dem Daten-Logger-Baumknoten angezeigt.

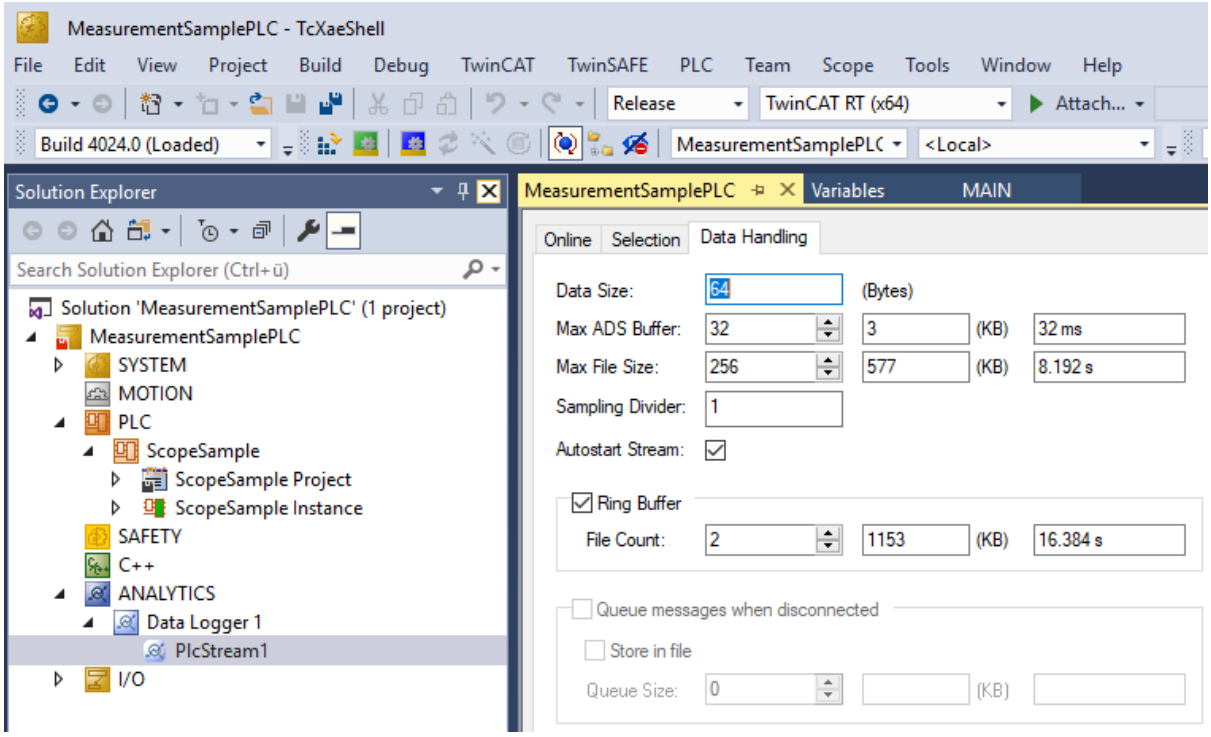


Ein zusätzlicher Geräte-Stream wird angezeigt, wenn Ihre Konfiguration ein EtherCAT-Prozessabbild liefert.

- In dem Stream gibt es einen Karteireiter **Selection**, wo die aufzuzeichnenden Variablen ausgewählt werden können.

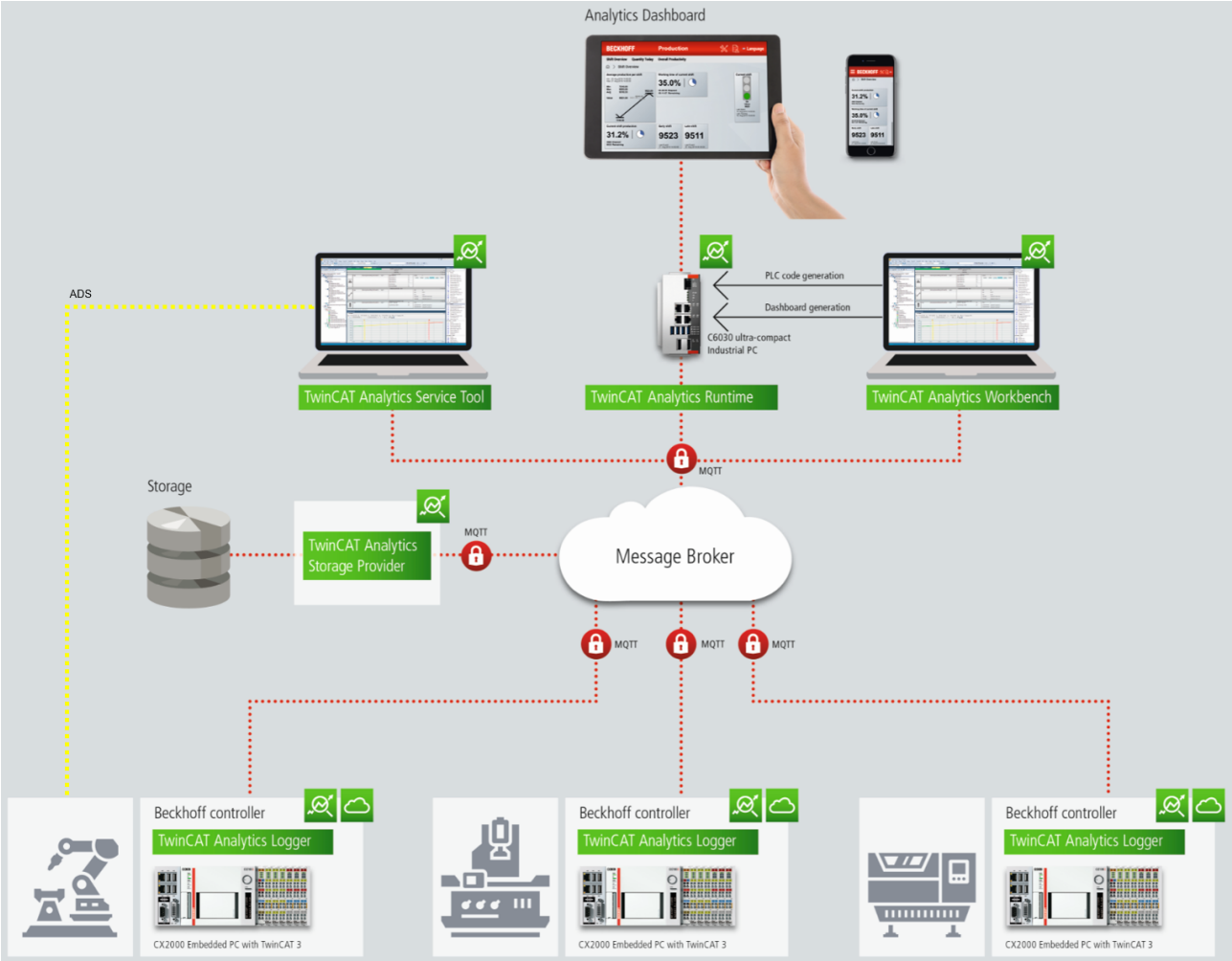


- Schließlich ist es im Karteireiter **Data Handling** möglich, die Paketgröße für die Frames zu ändern oder den Ringpuffer für Verbindungsunterbrechungen und Datei zu konfigurieren.



4.2 Kommunikation

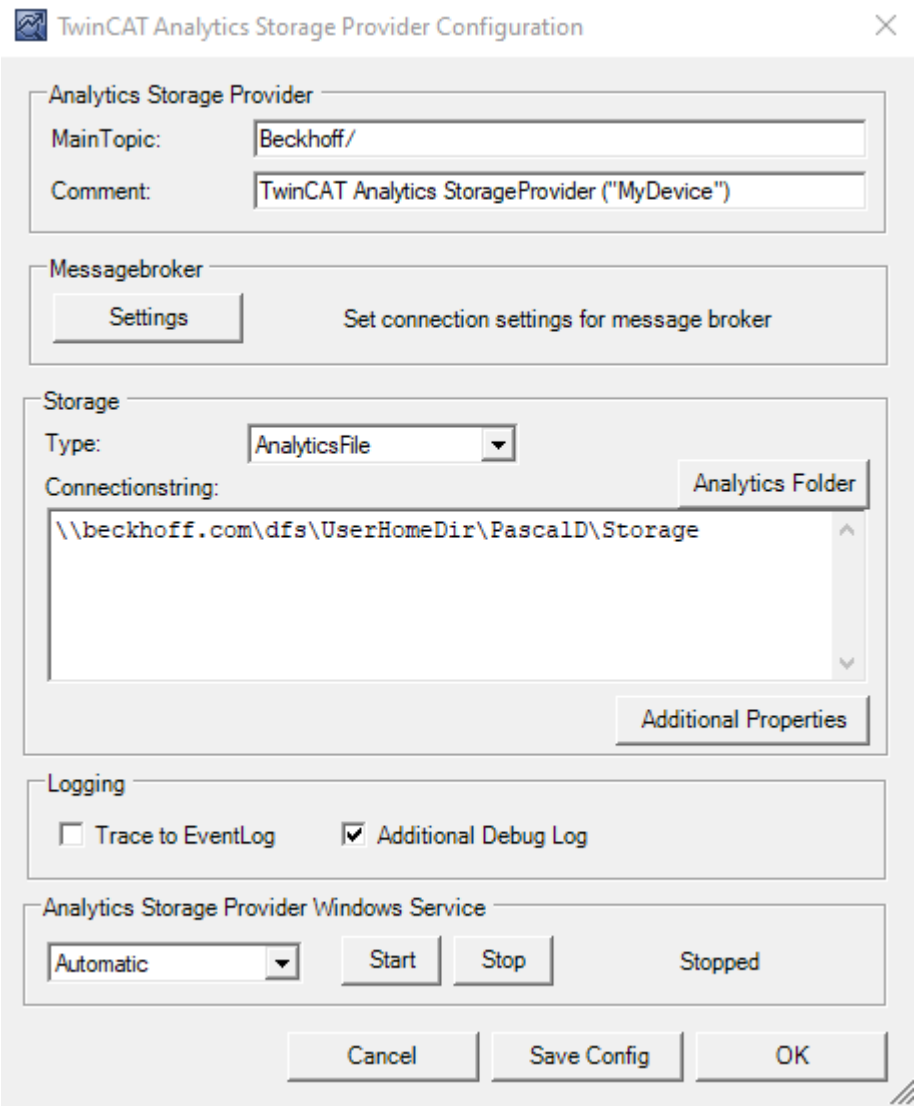
Derzeit kann der Analytics-Workflow vollständig über MQTT abgebildet werden. Die Engineering-Tools können auch über ADS auf die Daten der Maschinen zugreifen und Analysen durchführen.



Wenn man für das IoT-Kommunikationsprotokoll MQTT wählt, muss irgendwo im Netzwerk ein nativer MQTT-Message-Broker eingerichtet werden (VM in einem Cloud-System ist ebenfalls möglich). Dieser Message-Broker sorgt für eine Entkopplung der verschiedenen Anwendungen im Analytics-Workflow.

4.3 Historisierung von Daten

Nach der Installation des TwinCAT Analytics Storage Providers kann der im Hintergrund laufende Dienst konfiguriert werden. Hierzu finden Sie die Anwendung TwinCAT.Analytics.StorageProvider.Configurator im Ordner *C:\TwinCAT\Functions\TF3520-Analytics-StorageProvider\Tools*.

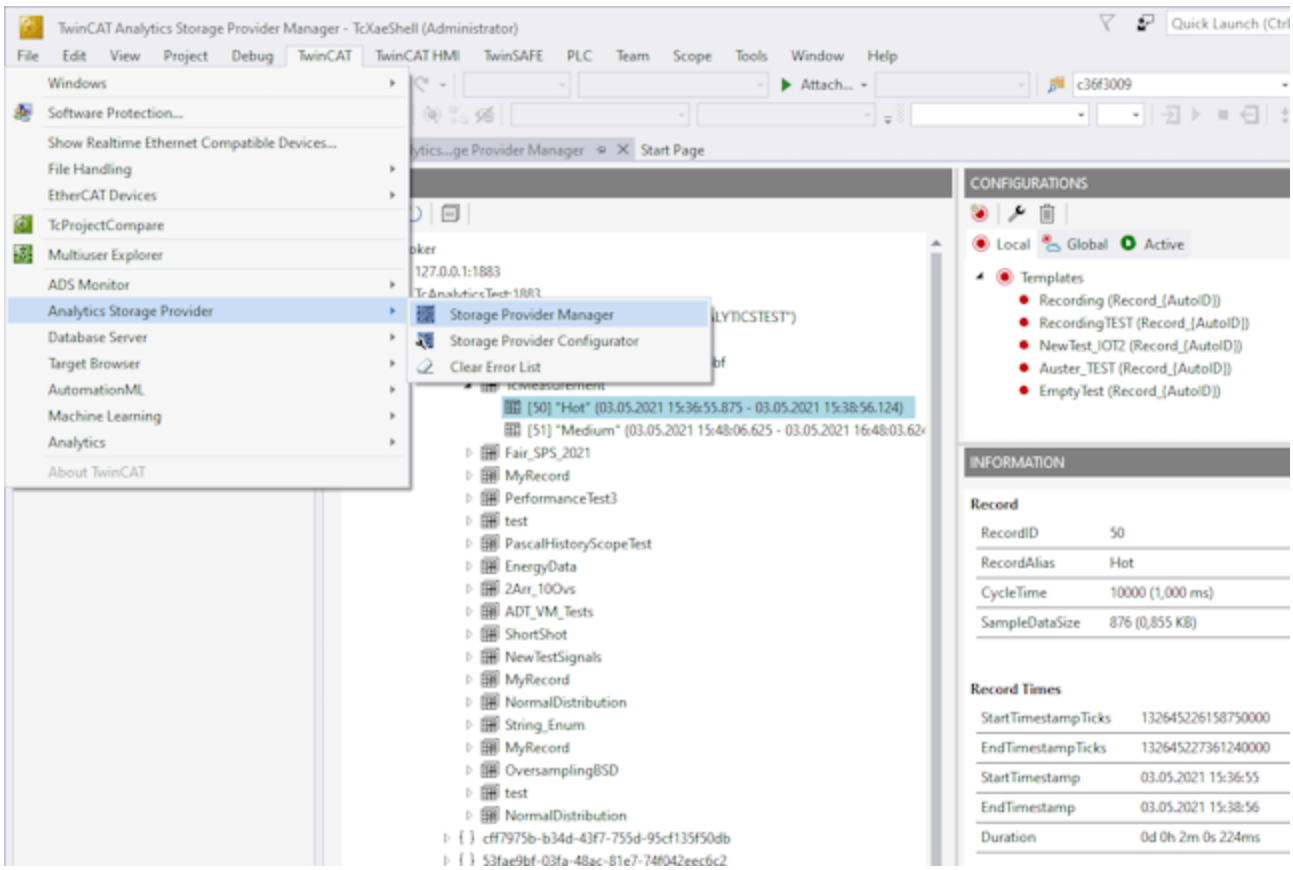


Der Hauptteil des Topics kann in der Konfiguration ebenso festgelegt werden wie der Kommentar, der zur Identifizierung dient, wenn mehr als ein Storage Provider beim Message-Broker registriert ist.

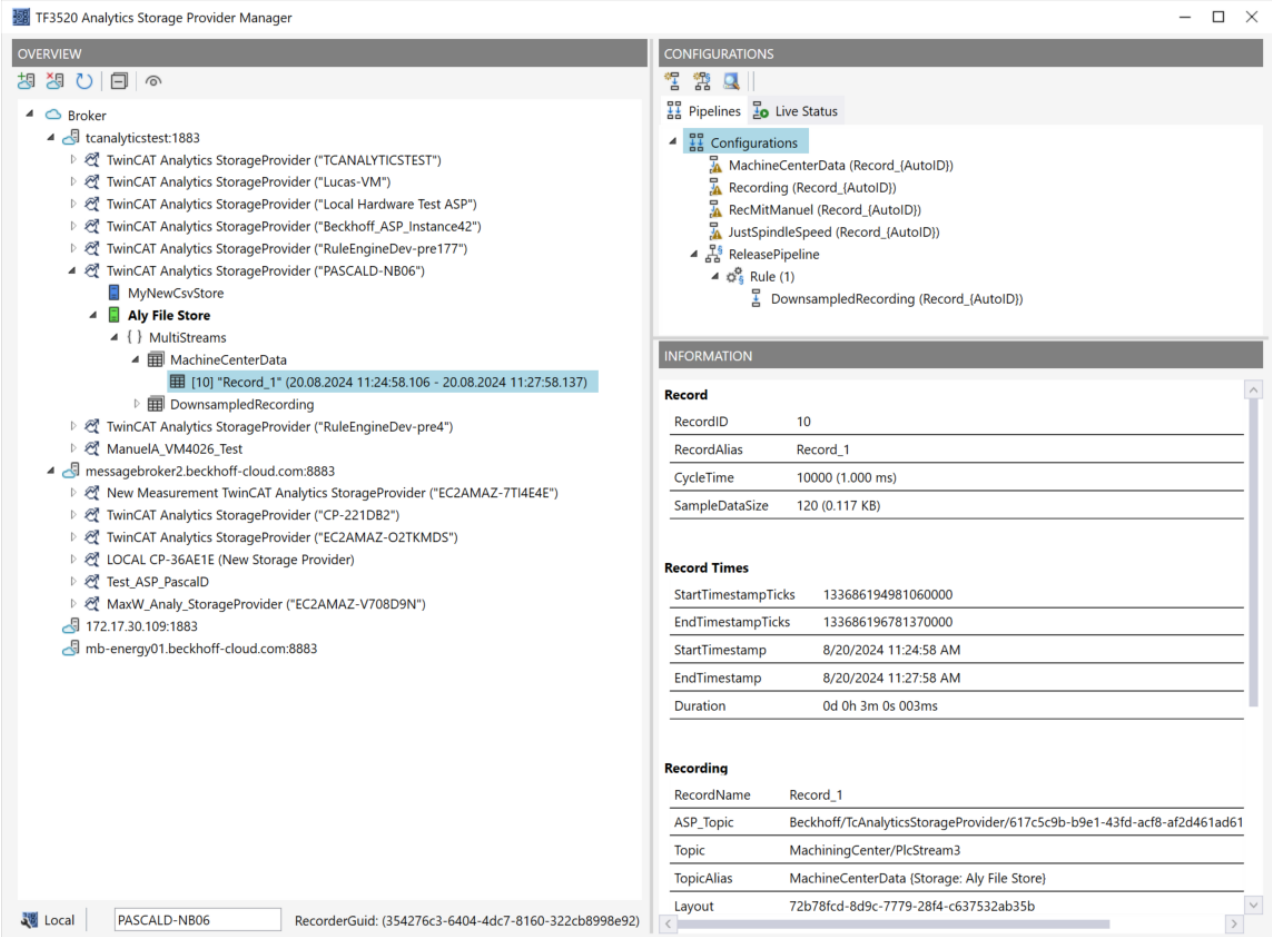
Sie können die Message-Broker-Einstellungen vornehmen und sich für einen Speichertyp entscheiden:

- Analytics File (Binärdatei)
- CSV Datei
- Microsoft SQL (Binär / Klartext)
- InlfuxDB (Klartext)
- Microsoft Azure Blob (Azure-Cloud erforderlich)

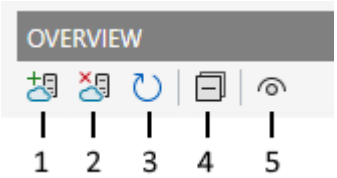
Zuletzt können Sie die Konfiguration speichern und den Dienst starten. Der nächste Schritt besteht in der Konfiguration der spezifischen Aufzeichnung. Hierzu sollten Sie in Ihrer Entwicklungsumgebung den **Storage Provider Manager** auswählen.



Mit dem Storage Provider Recorder können Aufnahme Definitionen erstellt, gestartet und verwaltet werden. Zusätzlich besteht die Möglichkeit, die Datenspeicher einzelner Analytics Storage Provider zu verwalten. Alle wichtigen Eigenschaften der gefundenen Analytics Storage Provider und historisierten Daten werden übersichtlich dargestellt.



Symbolleiste Manager Fenster („OVERVIEW“)

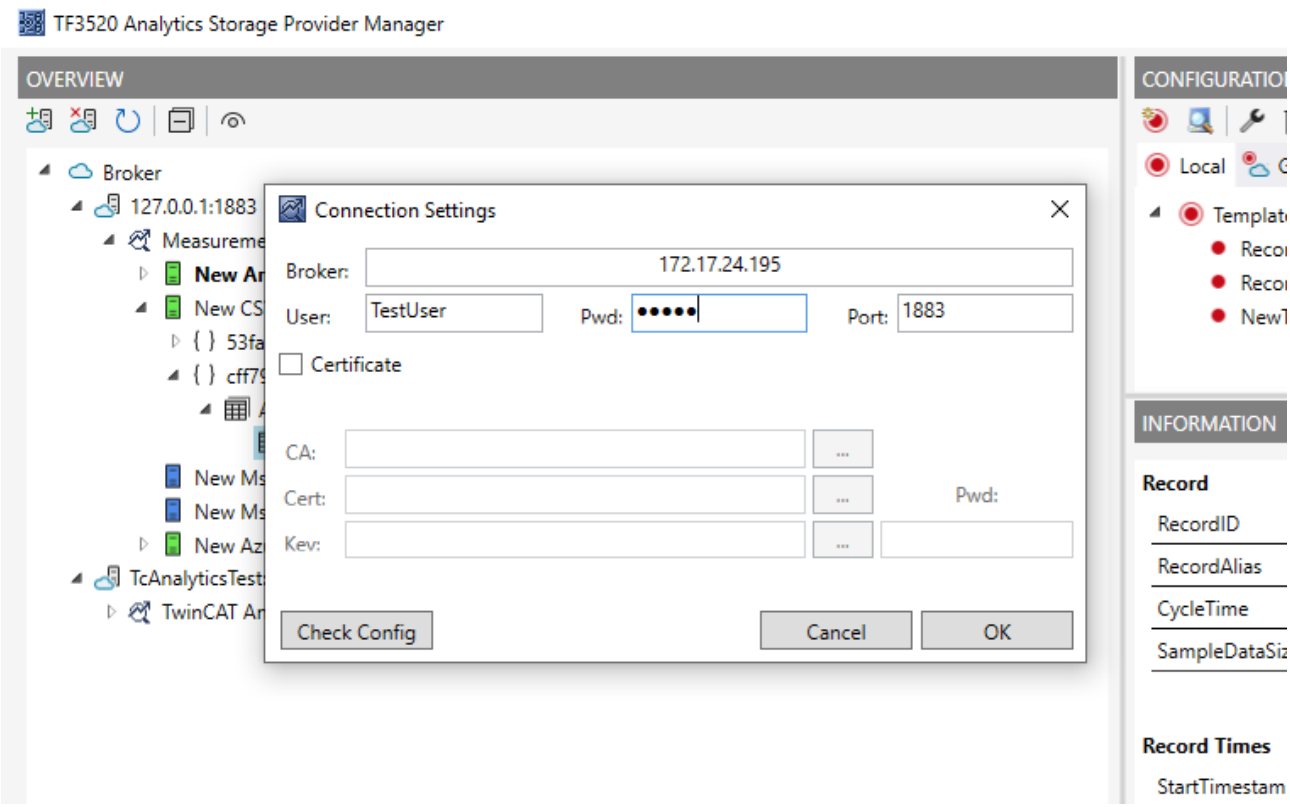


1	Neuen Broker hinzufügen
2	Ausgewählten Broker entfernen
3	Anzeige aktualisieren
4	Alle Knoten einklappen
5	Ansicht Wechsel zwischen Dark/Light Modus

Funktion Manager Fenster („OVERVIEW“)

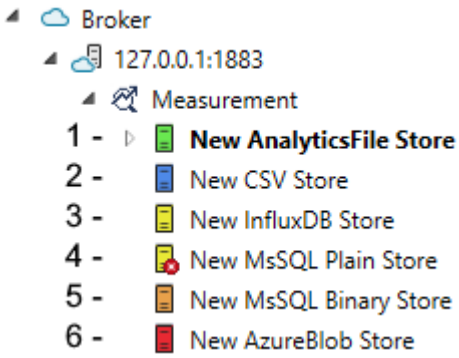
Vergeben Sie zunächst einen „RecorderAlias“. Dieser hilft die gestarteten Aufnahmen zu gruppieren und seine selbst gestarteten wieder zu finden.

Danach können ein oder mehrere Broker eingerichtet werden. Dies geschieht über die schon bekannte Eingabemaske für MQTT-Verbindungseigenschaften.



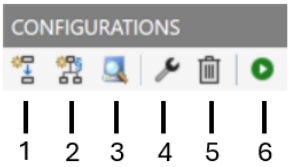
Sobald eine Verbindung zum Broker hergestellt werden konnte, werden alle darauf verbunden Analytics Storage Provider aufgelistet.

„Storage“ Status



1	Storage Online
2	Storage Offline
3	Storage startet auf
4	Storage startet mit Fehler. Es wird weiter versucht ihn zu starten.
5	Storage wird heruntergefahren
6	Storage befindet sich im Fehlerfall

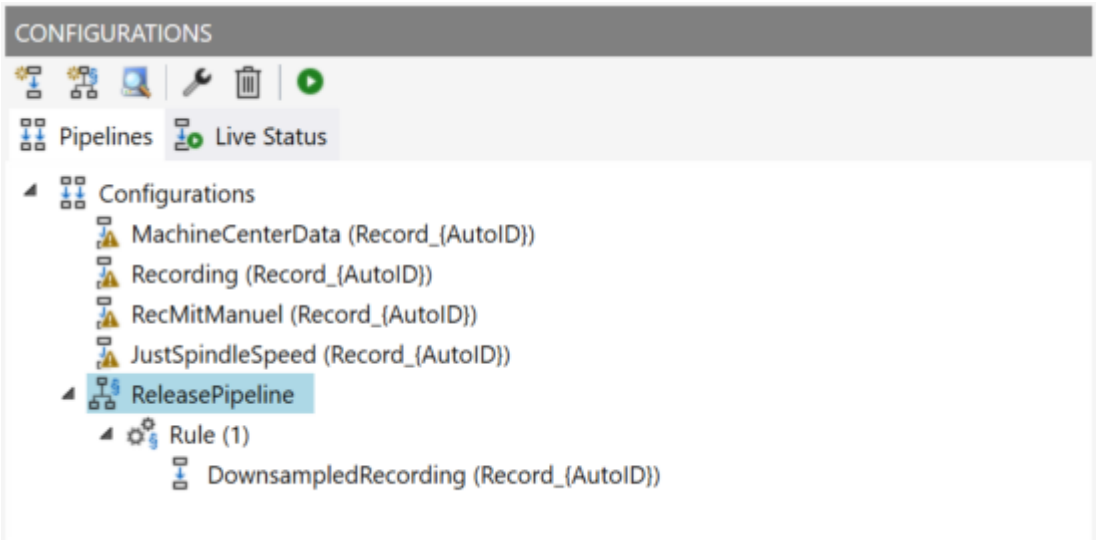
Symbolleiste Manager Fenster („CONFIGURATIONS“)



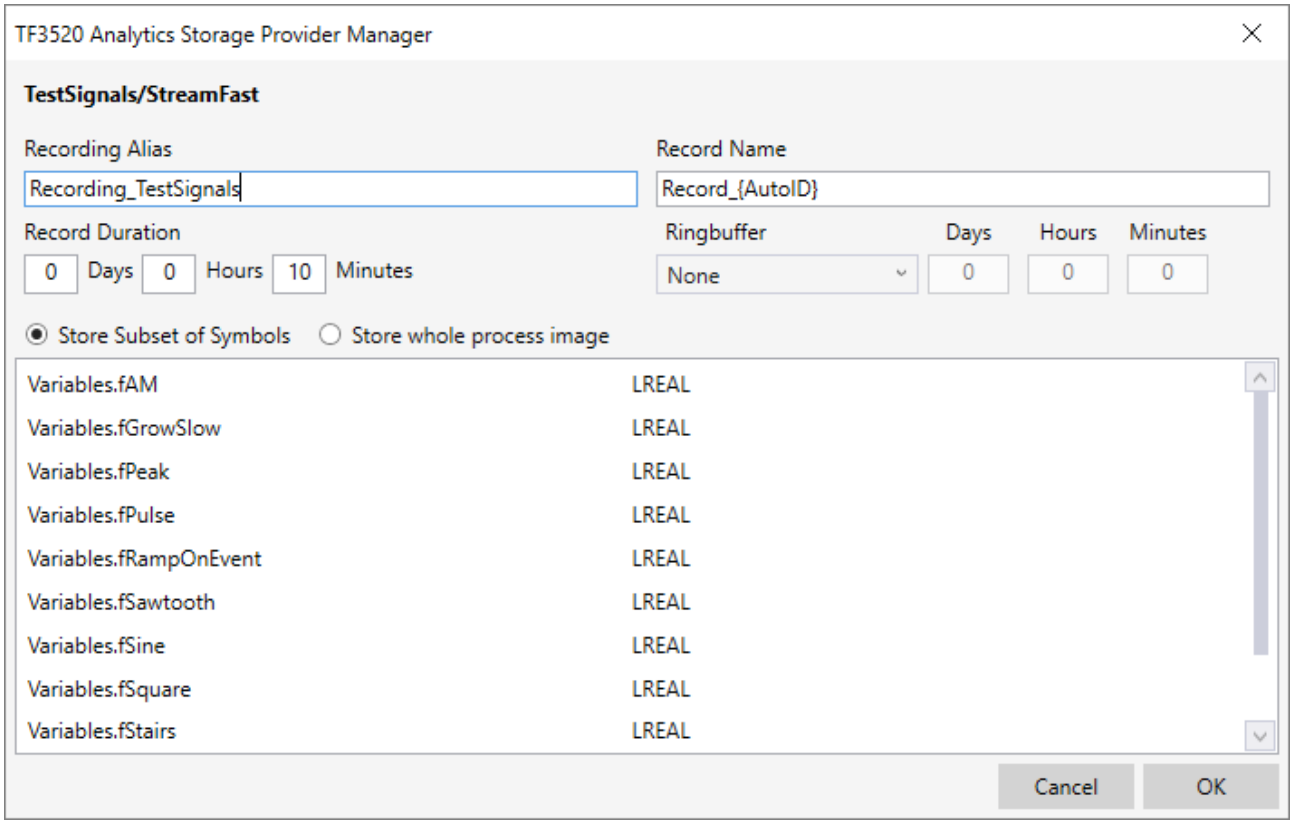
1	Eine neue Pipeline erzeugen
2	Eine neue Pipeline mit Rule Engine erzeugen
3	Target Browser öffnen für Beschaltung einfacher Pipelines
4	Eine ausgewählte Pipeline editieren
5	Eine ausgewählte Pipeline löschen
6	Eine ausgewählte Pipeline starten

Funktion Manager Fenster („CONFIGURATIONS“)

Das Fenster teilt sich in zwei Tabs auf. Pipelines und Live Status. Unter Pipelines finden Sie die Konfigurationen Ihrer Pipelines. Von hier können Sie neue Pipelines definieren, bestehende editieren, löschen oder starten.



Um eine neue einfache Pipeline zu erzeugen, drücken Sie den Button **Neue Pipeline erzeugen**. Folgender Dialog öffnet sich.



Sie können nun die Symbole, die aufgezeichnet werden sollen, vom Target Browser per Drag-and-Drop in den Dialog ziehen. Vergeben Sie zusätzlich ein Recording Alias und einen Record Name.

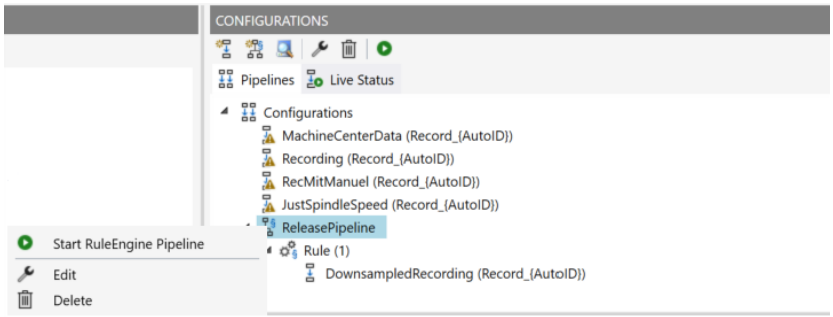
Für den Record Name stehen verschiedene Platzhalter zur Verfügung:

"{AutoID}"	
"{Topic}"	
"{SystemID}"	
"{Layout}"	
"{CycleTime}"	
"{SampleSize}"	
"{RecordStart}"	

Sie können auch Aufzeichnungsnamen und eine Dauer konfigurieren (anderenfalls läuft die Aufzeichnung endlos, bis sie manuell gestoppt wird). Ein Ringpuffer kann nach Speicherplatz oder Zeit eingestellt werden.

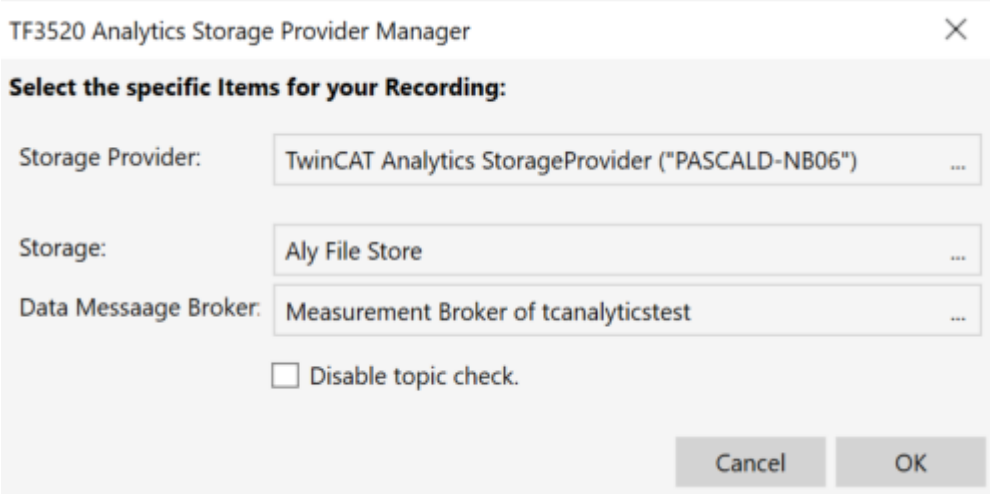
Mit **OK** werden die Eingaben bestätigt und einen neue lokale Aufnahmedefinition wird erzeugt.

Es ist nun möglich diese Definition direkt über die Toolbar oder das Kontextmenü zu starten.



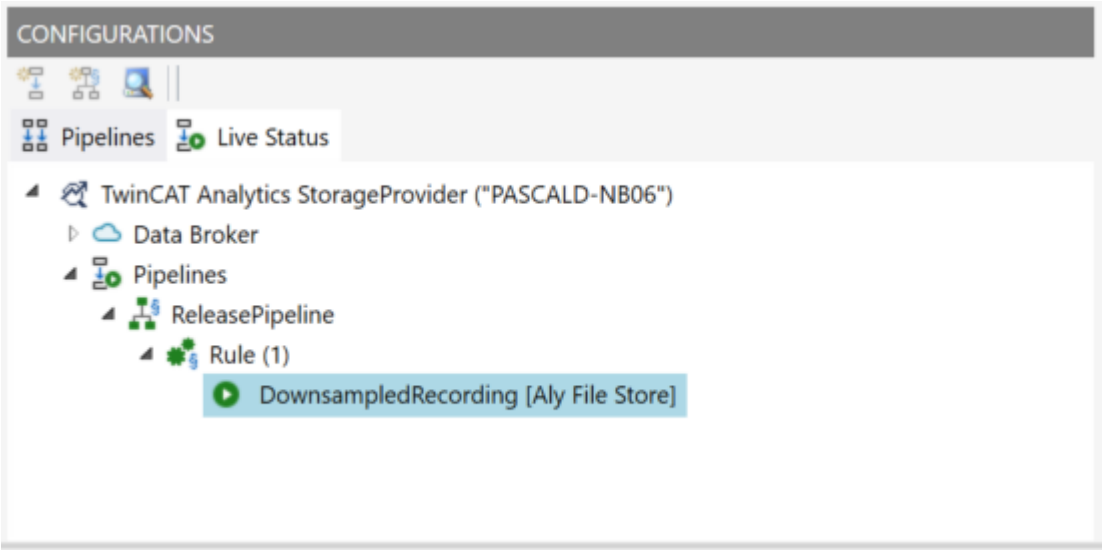
Es besteht aber auch die Möglichkeit die Definition global zugänglich zu machen. Dies kann über das Kontextmenü mit dem Eintrag **Publish Recording** gemacht werden.

Danach öffnet sich folgender Dialog:



Hier kann nun der gewünschte Analytics Storage Provider ausgewählt werden, über den die Definition veröffentlicht werden soll. Zusätzlich wird der Definition ein Storage und ein Data Broker des ausgewählten Analytics Storage Providers zugewiesen. Nach der Auswahl wird die Aufnahmedefinition mit **OK** bestätigt und an dem ausgewählten Analytics Storage Provider veröffentlicht. Dadurch kann sie von jedem Storage Provider Manager, der mit dem MQTT-Broker verbunden ist, gefunden werden.

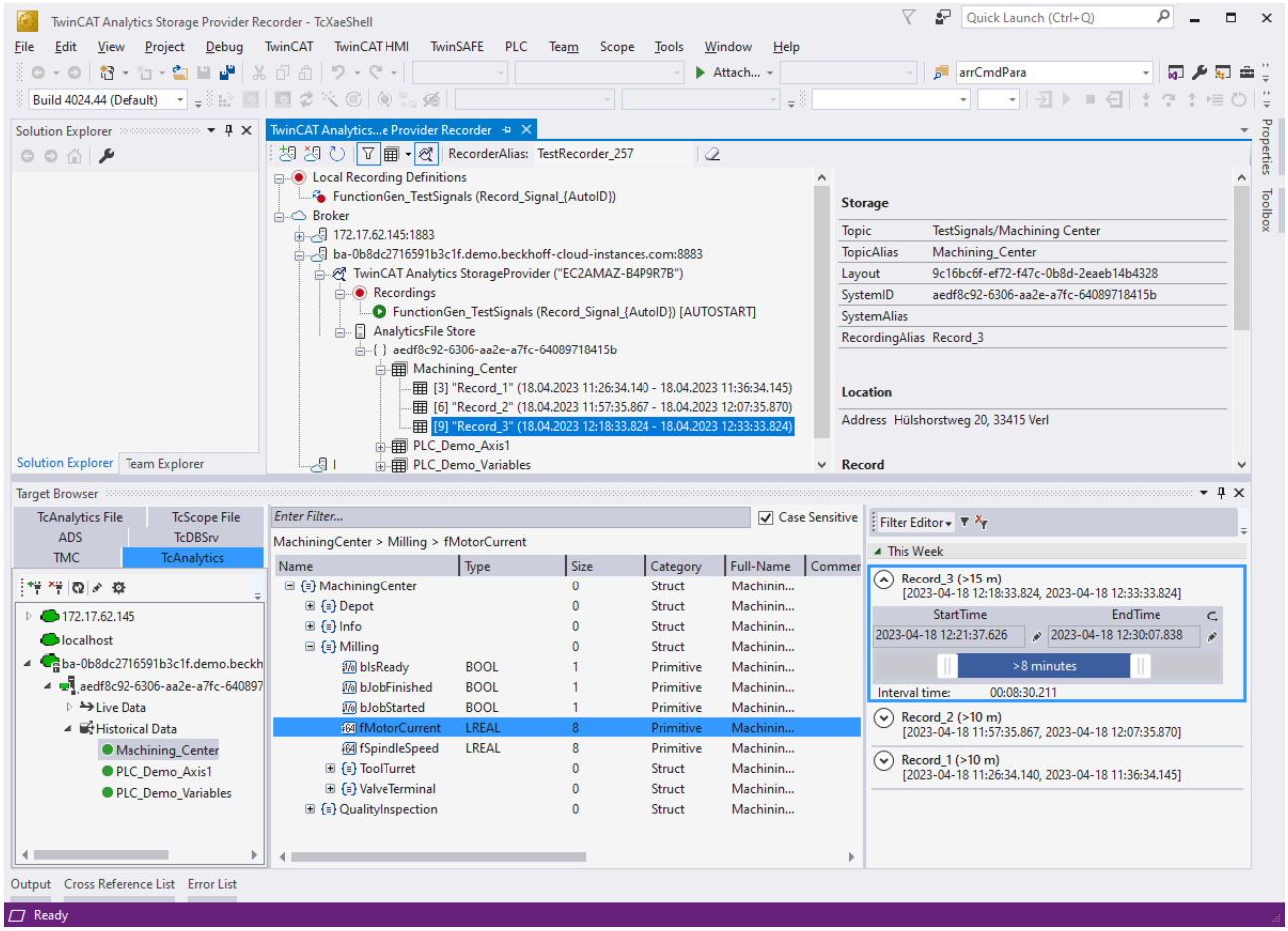
Nach dem Start einer Pipeline springt die Ansicht automatisch in den zweiten Tab, den Live Status.



Hier werden alle aktiven Recordings von allen Usern aufgelistet. Die Recordings können in diesem Tab beendet werden, es ist auch möglich zu dem resultierenden Record zu springen.

Historisierte Daten verwenden

Nach und auch während der Aufzeichnung können Sie die historischen Daten als Eingang für Ihre Analyse im Target Browser auswählen. Im Target Browser finden Sie für die historischen Daten eine neue Steuerung auf der rechten Seite. Dort können Sie die Zeitspanne für Ihre Daten auswählen.

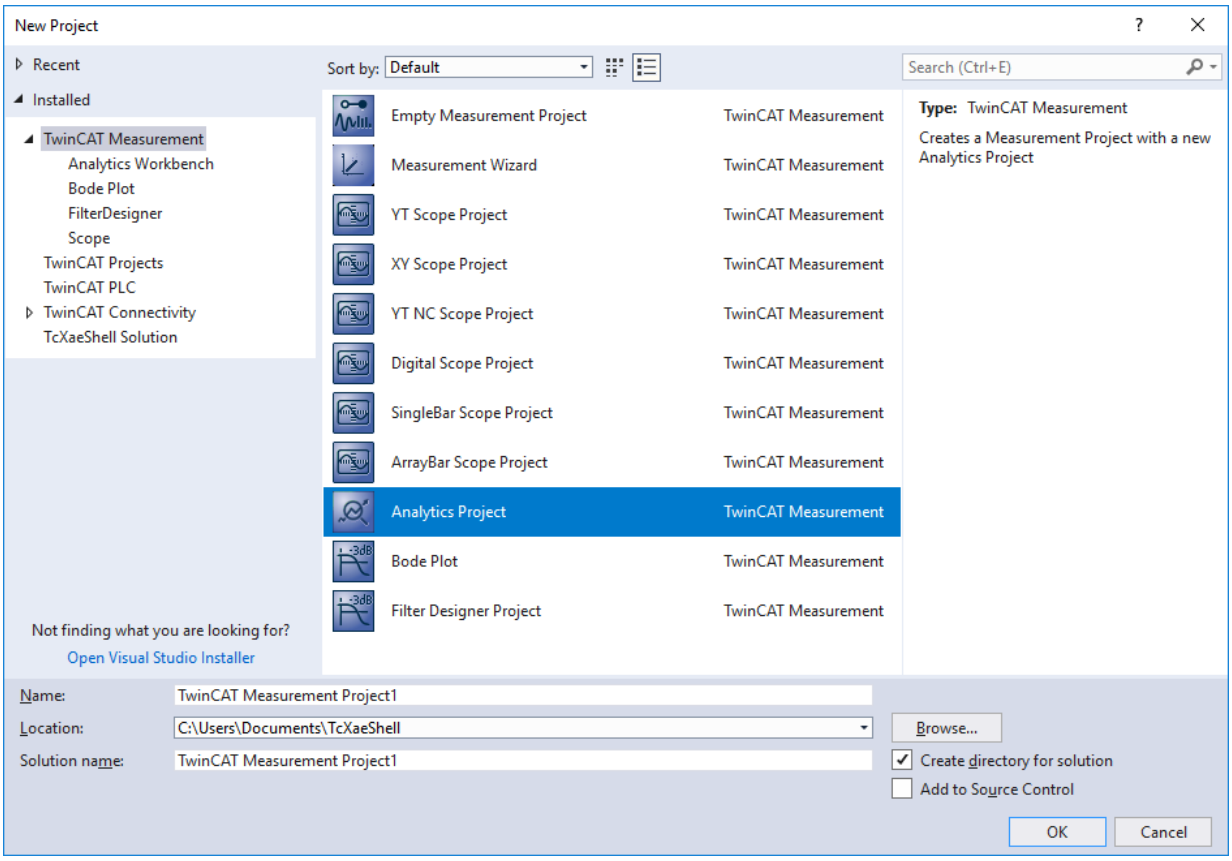


4.4 Analyse der Daten

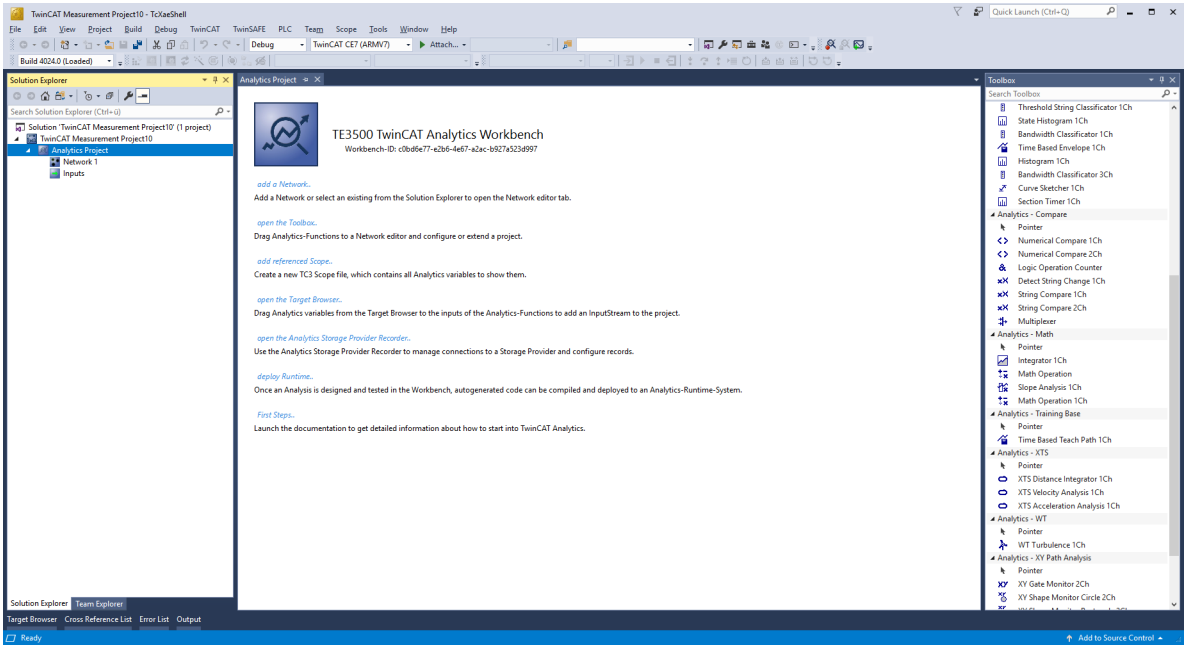
✓ Öffnen Sie Ihre TwinCAT Engineering-Umgebung, um die Analyse der Daten zu starten.

1. Öffnen Sie **Visual Studio® > File > New > Project…**

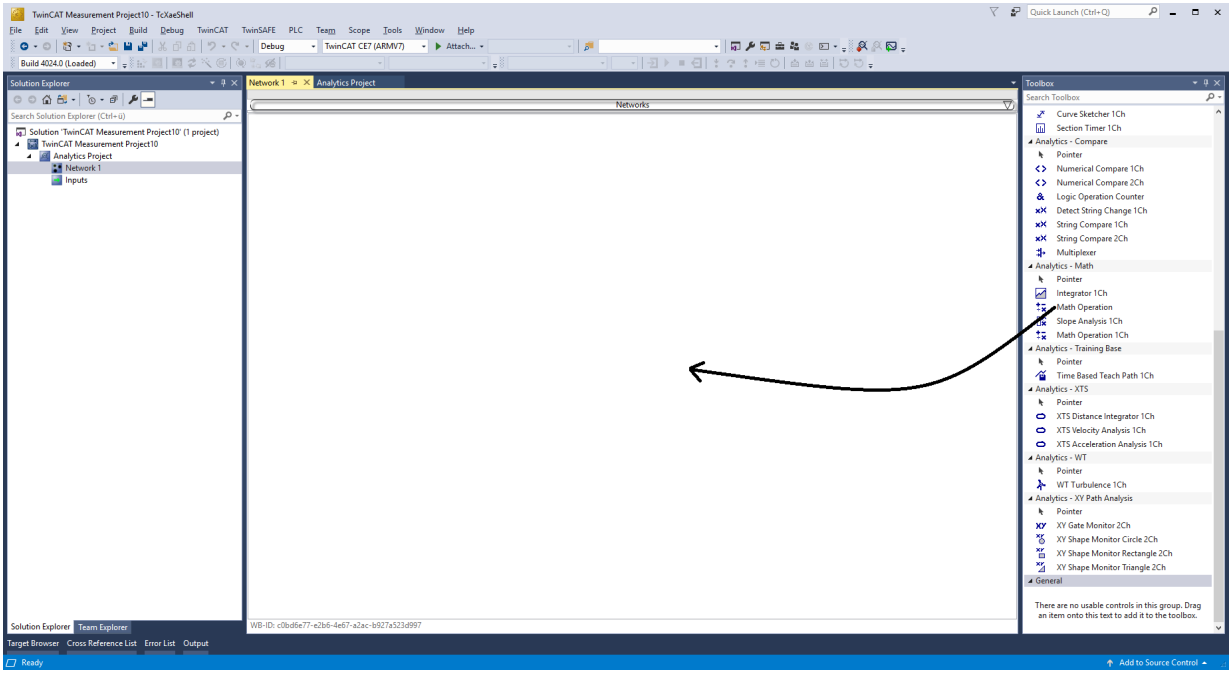
2. Wählen Sie aus **TwinCAT Measurement** die **Analytics-Projektvorlage** aus.



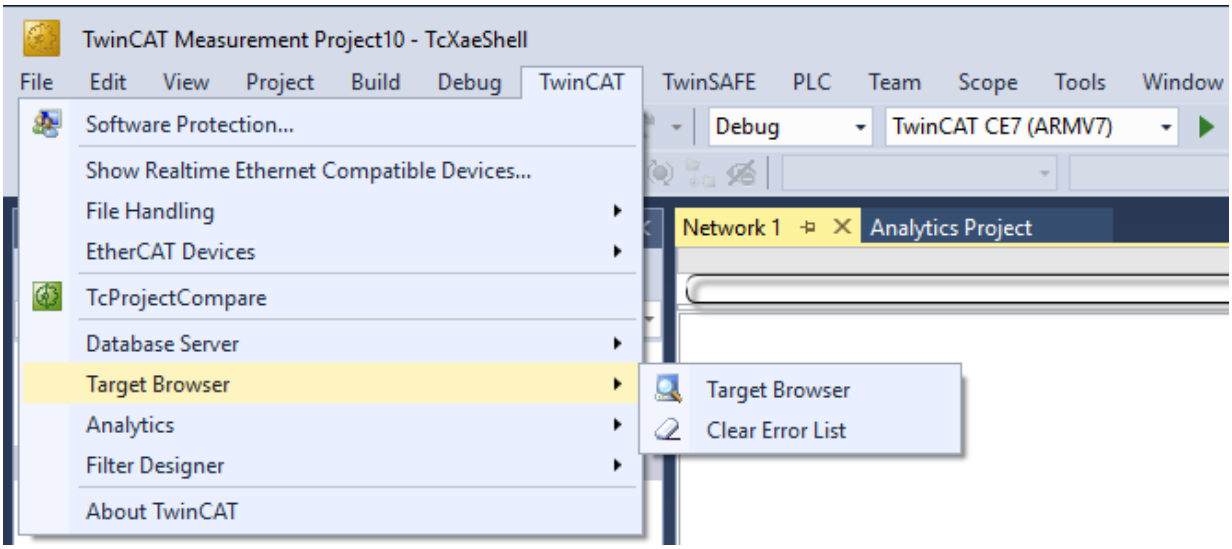
⇨ Das neue Projekt wird im Projektmappen-Explorer angezeigt. Nach einem Klick auf das Baumknotenelement **Analytics Project** öffnet sich ein Startfenster, in dem Sie Ihre erste Aktion auswählen können. Von hier aus können Sie ein Netzwerk hinzufügen, die **Toolbox** öffnen, den **Target Browser** öffnen oder den **Analytics Storage Provider Recorder** öffnen. In den folgenden Schritten werden Sie all diese Aktionen durchführen.



3. Es ist sinnvoll, als Erstes die **Toolbox** von Visual Studio® zu öffnen. Dort finden Sie alle unterstützten Algorithmen von TwinCAT Analytics. Die Algorithmen müssen in Netzwerke gruppiert und organisiert werden. Rechtsklicken Sie auf **Analytics Project**, um ein neues Netzwerk hinzufügen, oder fügen Sie mit Hilfe der Startseite ein Netzwerk hinzu. Das erste Netzwerk wird immer standardmäßig generiert.

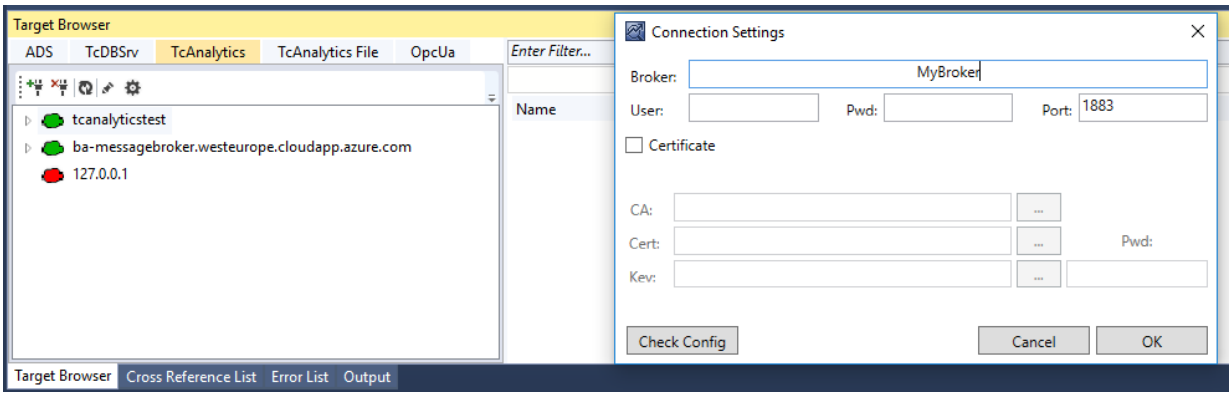


4. Wenn Sie auf das Netzwerk klicken, wird ein Editor geöffnet. Nun können Sie den gewünschten Algorithmus per Drag-and-Drop in die Editor-Oberfläche ziehen.
5. Nach der Auswahl des Algorithmus müssen Sie Eingangsvariablen mit den Modulen (Algorithmus) verbinden. Öffnen Sie dazu den **Target Browser**.
   TwinCAT > Target Browser > Target Browser

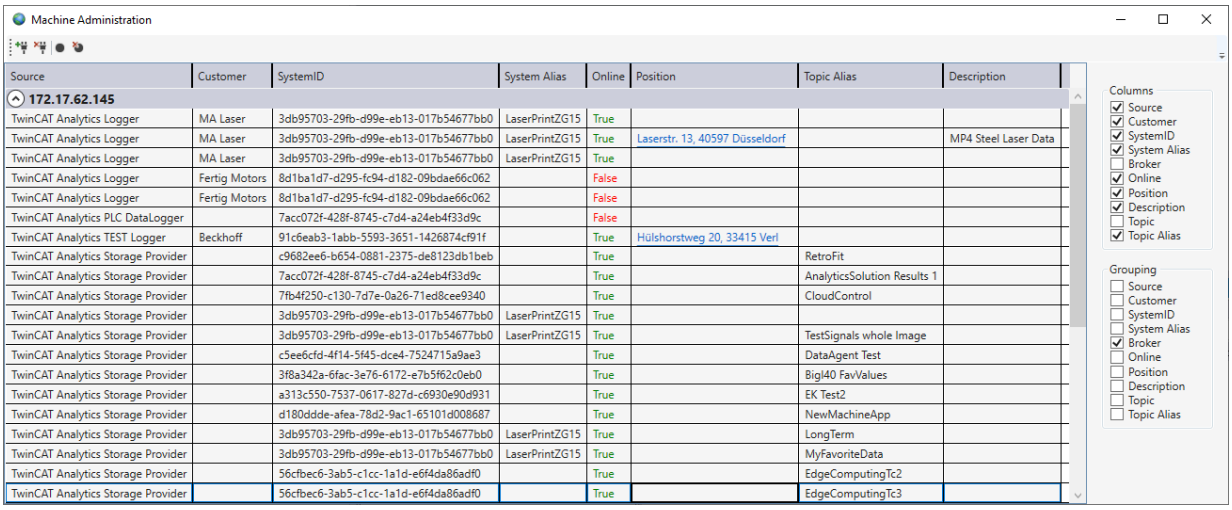


6. Wählen Sie nun den Karteireiter **TcAnalytics** oder **TcAnalyticsFile** im Target Browser aus. Mit dem Karteireiter **TcAnalytics** (MQTT) geht es weiter.

7. Klicken Sie auf das grün markierte Icon in der Symbolleiste dieser Analytics-Erweiterung. Es öffnet sich ein Fenster, in dem Sie die Konnektivitätsdaten Ihres Message-Brokers angeben können.

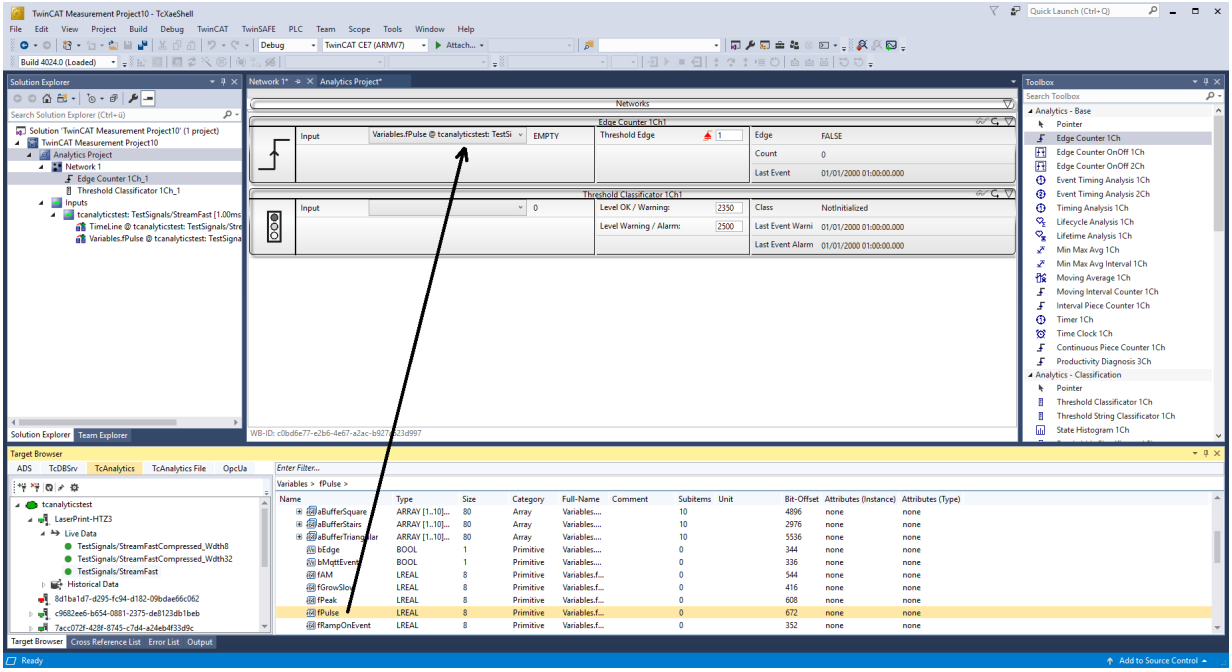


8. Wählen Sie Ihren MQTT-Analytics-Client aus (TwinCAT Analytics Logger, TwinCAT IoT Data Agent oder Beckhoff EK9160). Für jede Steuerung gibt es eine eindeutige ID. Diese ID wird im Target Browser angezeigt.
9. Über einen Klick auf das **Zahnrad-Icon**, gelangen Sie zur Seite Machine Administration. Hier können Sie einen System Alias-Namen vergeben, der im Target Browser anstelle der ID angezeigt wird.

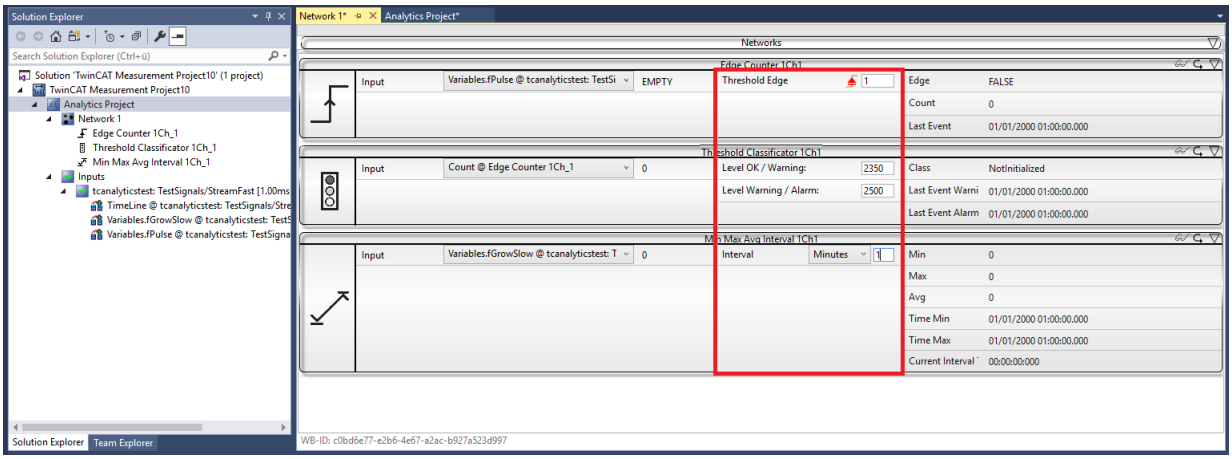


Source	Customer	SystemID	System Alias	Online	Position	Topic Alias	Description
172.17.62.145							
TwinCAT Analytics Logger	MA Laser	3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True			
TwinCAT Analytics Logger	MA Laser	3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True	Laserstr. 13, 40597 Düsseldorf		MP4 Steel Laser Data
TwinCAT Analytics Logger	MA Laser	3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True			
TwinCAT Analytics Logger	Fertig Motors	8d1ba1d7-d295-fc94-d182-09bdae66c062		False			
TwinCAT Analytics Logger	Fertig Motors	8d1ba1d7-d295-fc94-d182-09bdae66c062		False			
TwinCAT Analytics PLC DataLogger		7acc072f-428f-8745-c7d4-a24eb4f33d9c		False			
TwinCAT Analytics TEST Logger	Beckhoff	91c6eab3-1abb-5593-3651-1426874cf91f		True	Hülshorstweg 20, 33415 Verl		
TwinCAT Analytics Storage Provider		c9682ee6-b654-0881-2375-de8123db1beb		True		RetroFit	
TwinCAT Analytics Storage Provider		7acc072f-428f-8745-c7d4-a24eb4f33d9c		True		AnalyticsSolution Results 1	
TwinCAT Analytics Storage Provider		7fb4f250-c130-7d7e-0a26-71ed8cee9340		True		CloudControl	
TwinCAT Analytics Storage Provider		3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True			
TwinCAT Analytics Storage Provider		3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True		TestSignals whole Image	
TwinCAT Analytics Storage Provider		c5ee6cfd-4f14-5f45-dce4-7524715a9ae3		True		DataAgent Test	
TwinCAT Analytics Storage Provider		3f8a342a-6fac-3e76-6172-e7b5f62c0eb0		True		Bigl40 FavValues	
TwinCAT Analytics Storage Provider		a313c550-7537-0617-827d-c6930e90d931		True		EK Test2	
TwinCAT Analytics Storage Provider		d180ddde-afea-78d2-9ac1-65101d008687		True		NewMachineApp	
TwinCAT Analytics Storage Provider		3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True		LongTerm	
TwinCAT Analytics Storage Provider		3db95703-29fb-d99e-eb13-017b54677bb0	LaserPrintZG15	True		MyFavoriteData	
TwinCAT Analytics Storage Provider		56cfbec6-3ab5-c1cc-1a1d-e6f4da86adf0		True		EdgeComputingTc2	
TwinCAT Analytics Storage Provider		56cfbec6-3ab5-c1cc-1a1d-e6f4da86adf0		True		EdgeComputingTc3	

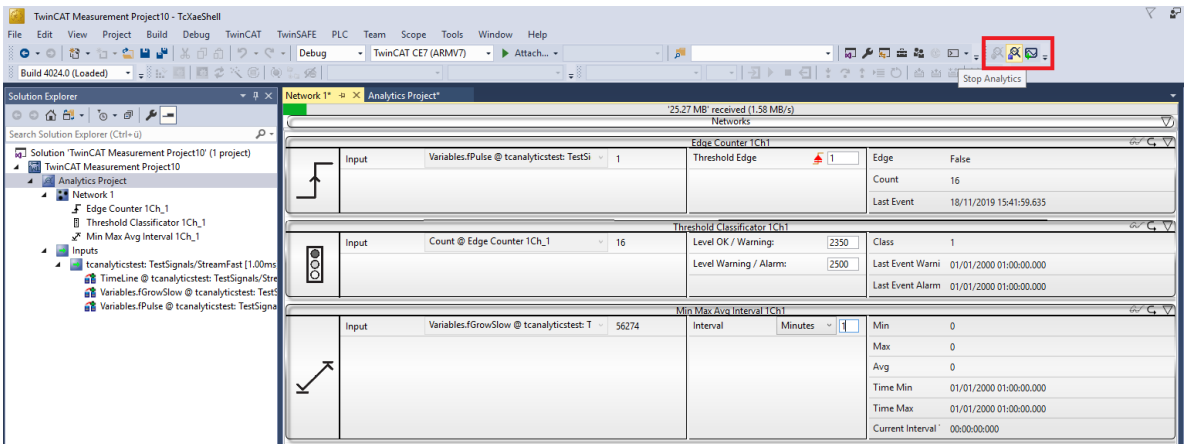
10. Im nächsten Schritt können Sie zwischen Live-Daten und historischen Daten für jeden MQTT Analytics-Client wählen. Die historischen Daten werden in diesem Fall von dem TwinCAT Analytics Storage Provider bereitgestellt.



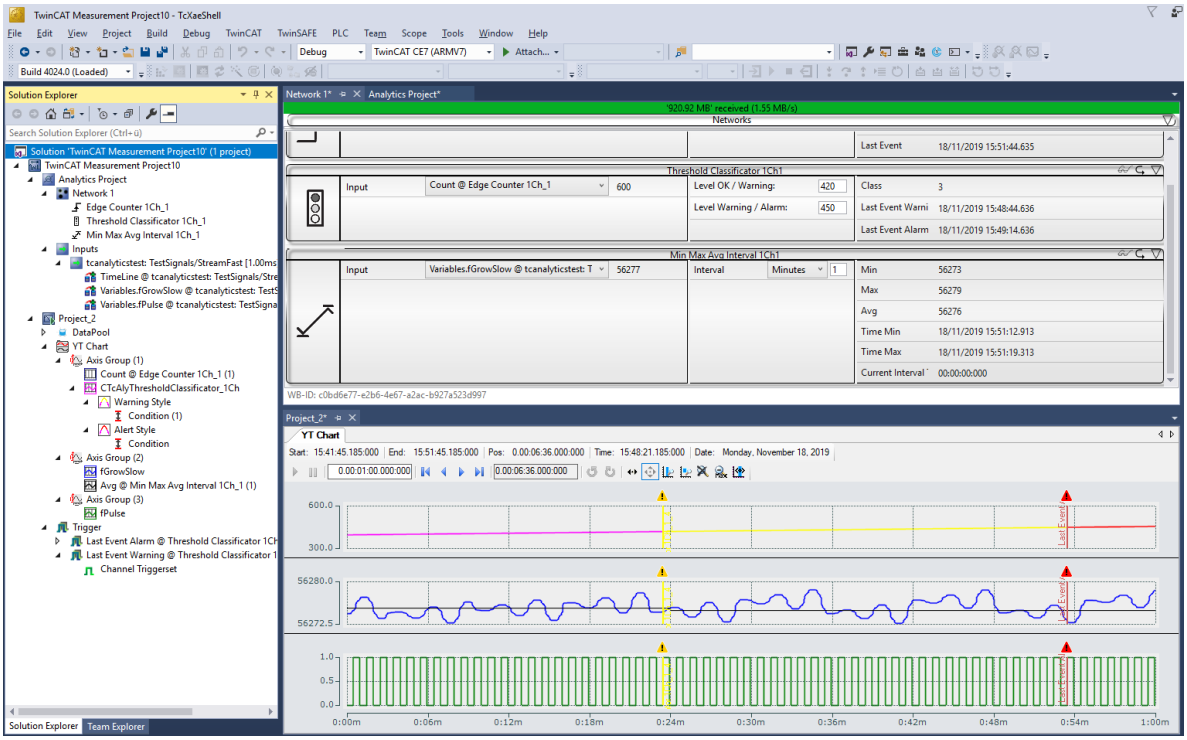
11. Sie können die Variablen per Drag-and-Drop in die Eingänge des spezifischen Algorithmus ziehen. Bei den meisten Algorithmen können Bedingungen wie Schwellen, Zeitintervalle, logische Operatoren usw. festgelegt werden. Diese Einstellungen werden in der Mitte jedes Moduls vorgenommen.



⇨ Schließlich ist Ihr erstes Analytics Project fertiggestellt. Um die Analyse zu starten, klicken Sie auf **Start Analytics**. Um die Analyse zu stoppen, klicken Sie auf **Stop Analytics**.



⇨ Vor dem Start der Analyse oder auch während der Laufzeit können Sie auf die Schaltfläche **Add Reference Scope** klicken. Dadurch wird automatisch eine zu Ihrem Analytics-Projekt passende Scope-Konfiguration erstellt.

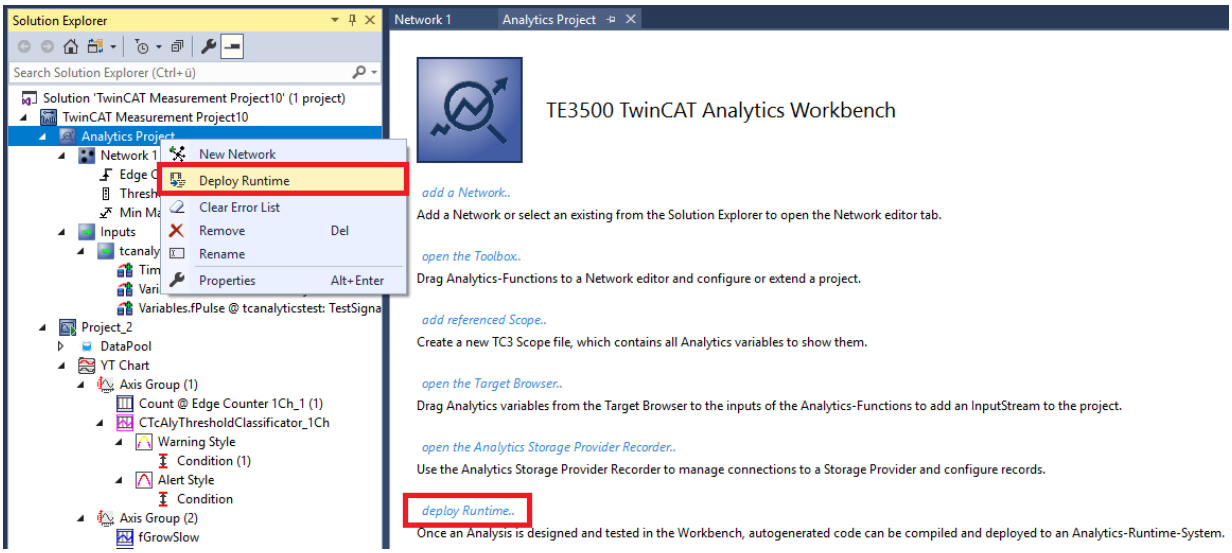


⇨ Die Ergebnisse der Analyse können in den Scope View-Grafiken per Drag-and-Drop angezeigt werden. Beispielsweise kann ein Mittelwert als neuer Kanal in der Ansicht angezeigt werden. Zeitstempel als Marker auf den X-Achsen zeigen signifikante Werte.

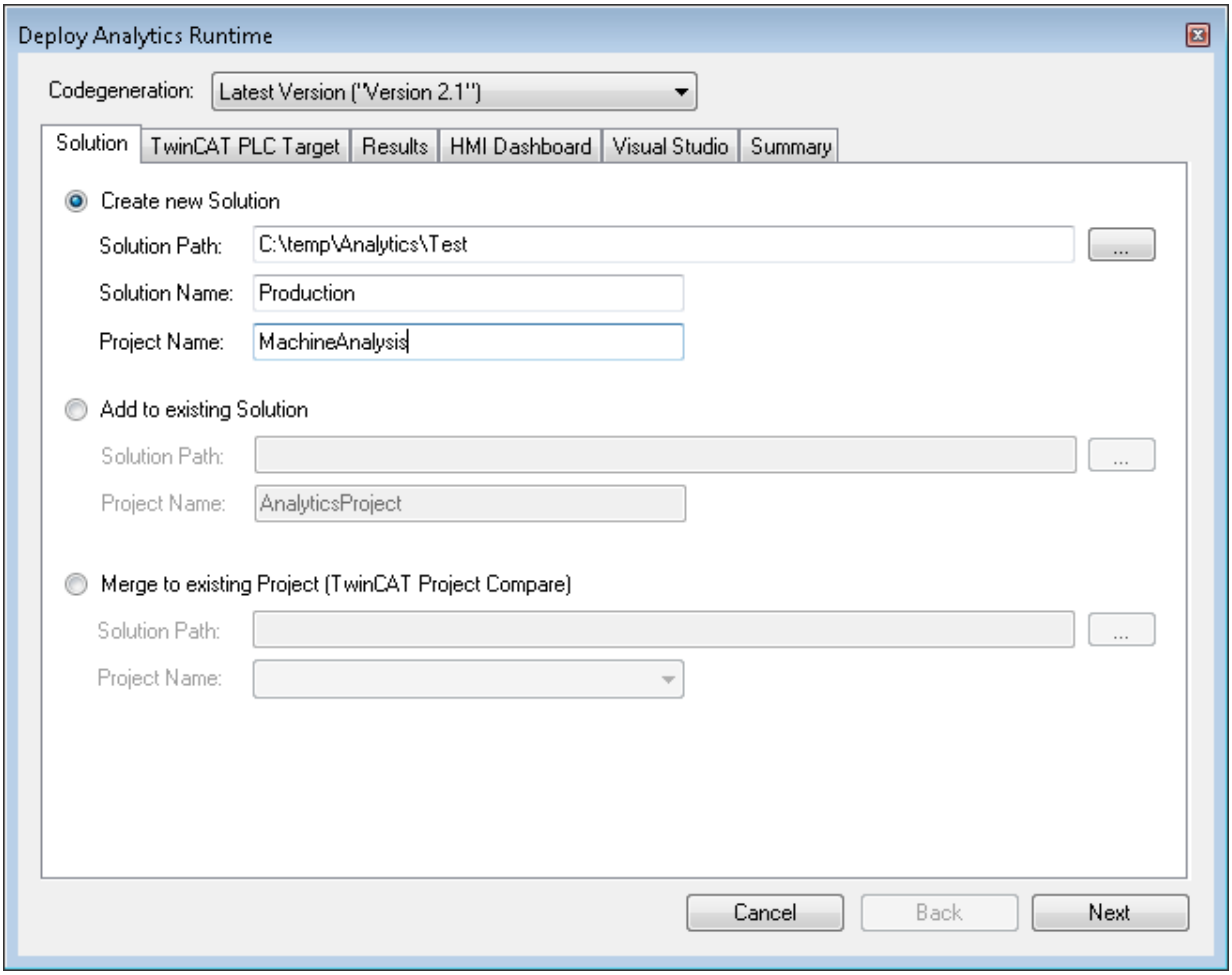
4.5 24-stündige Anwendung von Analytics

Der letzte große Schritt im Workflow von TwinCAT Analytics ist die kontinuierliche 24-stündige Maschinenanalyse. Sie läuft parallel zu den Maschinenanwendungen im Feld. Um dies ganz einfach zu gestalten, kann die TwinCAT Analytics Workbench automatisch einen SPS-Code und ein HTML5-basiertes Dashboard Ihrer Analytics-Konfiguration generieren. Beide können in eine TwinCAT Analytics Runtime (TC3 PLC und HMI Server) heruntergeladen werden und bieten dieselben Analyseergebnisse wie das Konfigurator-Tool in der Engineering-Umgebung.

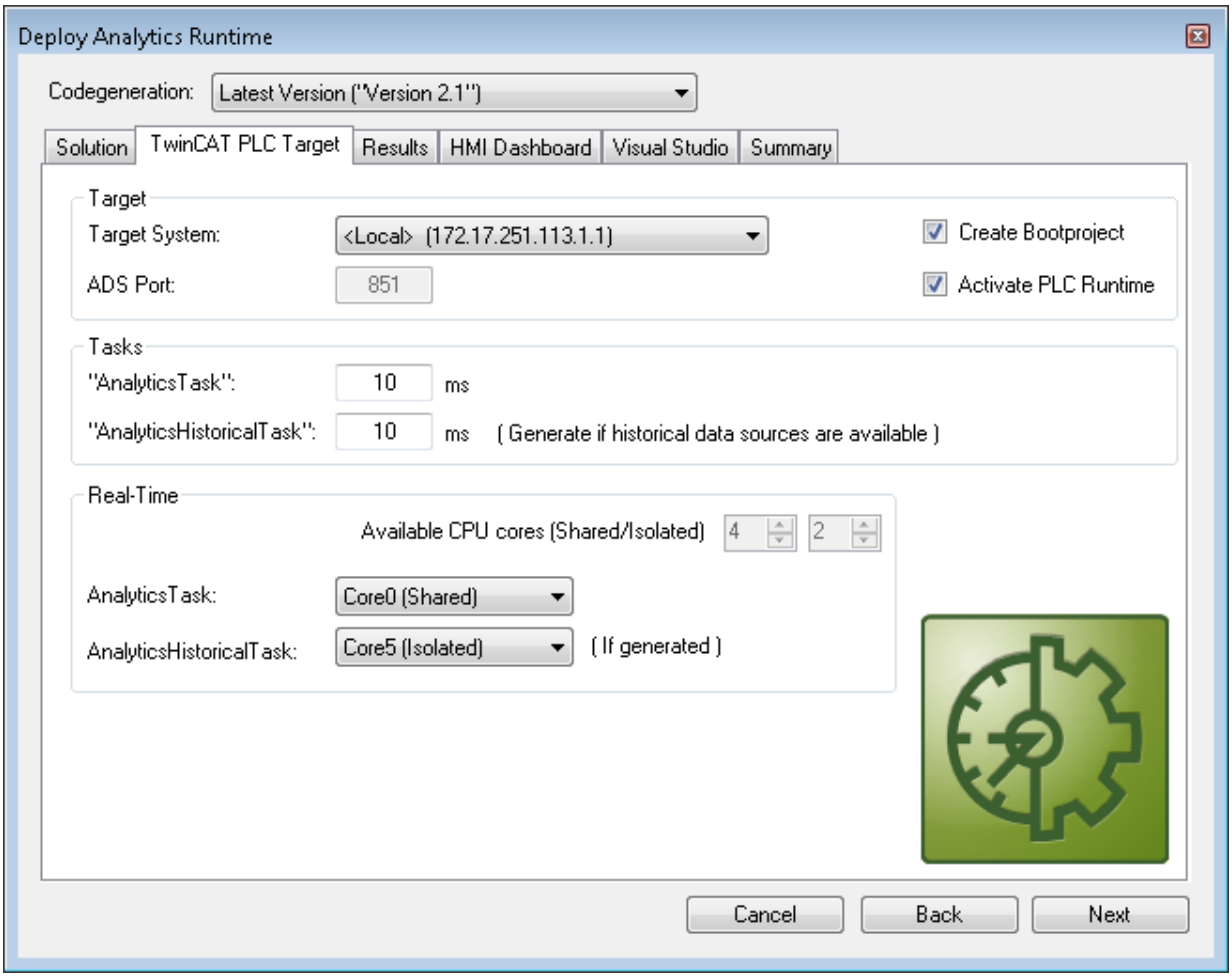
✓ Speichern Sie zunächst Ihre Konfiguration und öffnen Sie den Analytics Deploy Runtime-Assistenten. Dies kann über das Kontextmenü im Analytics Project-Tree Item oder über die Startseite erfolgen.



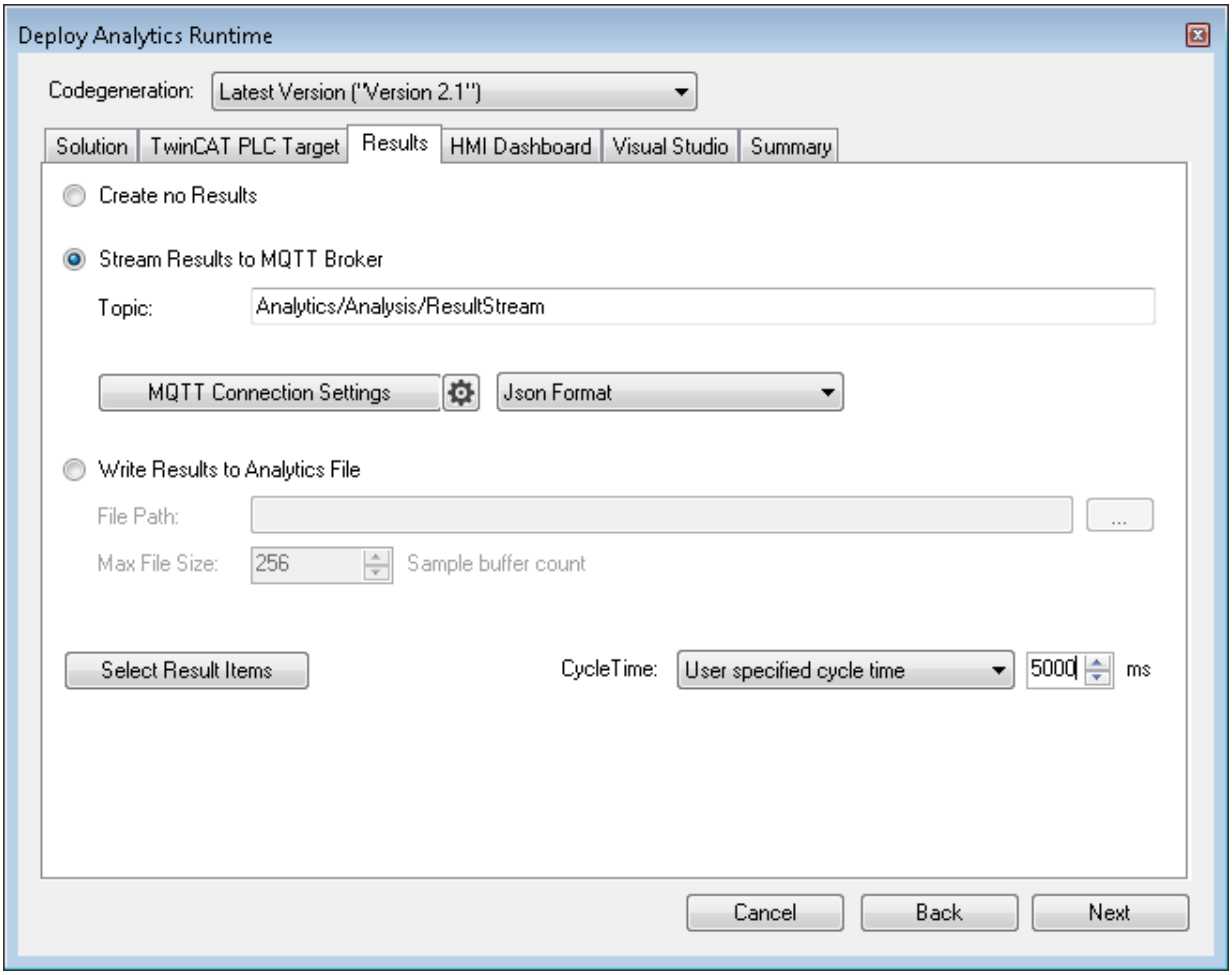
1. Wenn der Assistent geöffnet ist, können Sie sich durch einige Karteireiter klicken. Der erste heißt Solution. Hier können Sie entscheiden, wie Ihr Analytics-Projekt im SPS-Code verwendet werden soll:
   Als…
   vollständig neue Lösung.
   Teil einer vorhandenen Lösung.
   Aktualisierung einer vorhandenen Analytics-Lösung.



2. Im Karteireiter **TwinCAT PLC Target** können Sie das ADS-Zielsystem wählen, das die TwinCAT Analytics Runtime (TF3550) ausführt. Das erzeugte Projekt ist sofort ausführbar. Hierzu können Sie die Option Activate PLC Runtime setzen. Zudem kann ausgewählt werden, dass direkt ein Bootprojekt erstellt wird.

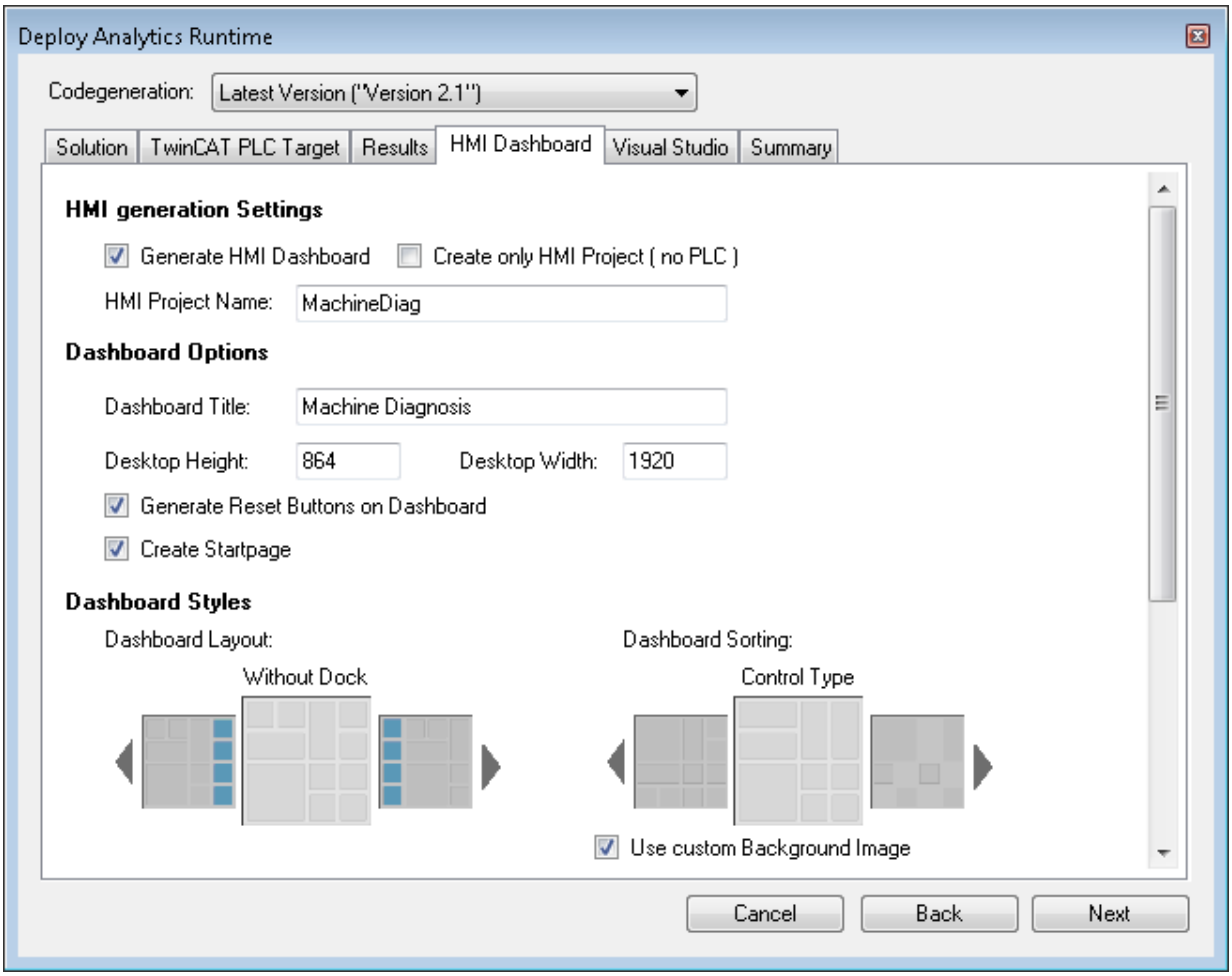


3. Insbesondere bei virtuellen Maschinen ist es wichtig, das Projekt auf isolierten Kernen auszuführen, was ebenfalls eine Option in diesem Karteireiter ist. Der nächste Karteireiter **Results** wird nur benötigt, wenn Sie in den Algorithmuseigenschaften die Option **Stream Results** ausgewählt haben. Wenn Sie Ergebnisse senden möchten, können Sie hier entscheiden, in welcher Weise (lokal in einer Datei/durch MQTT) und welchem Format (binär/JSON) dies geschehen soll. Auch dies wird automatisch generiert und umgehend nach Aktivierung ausgeführt.

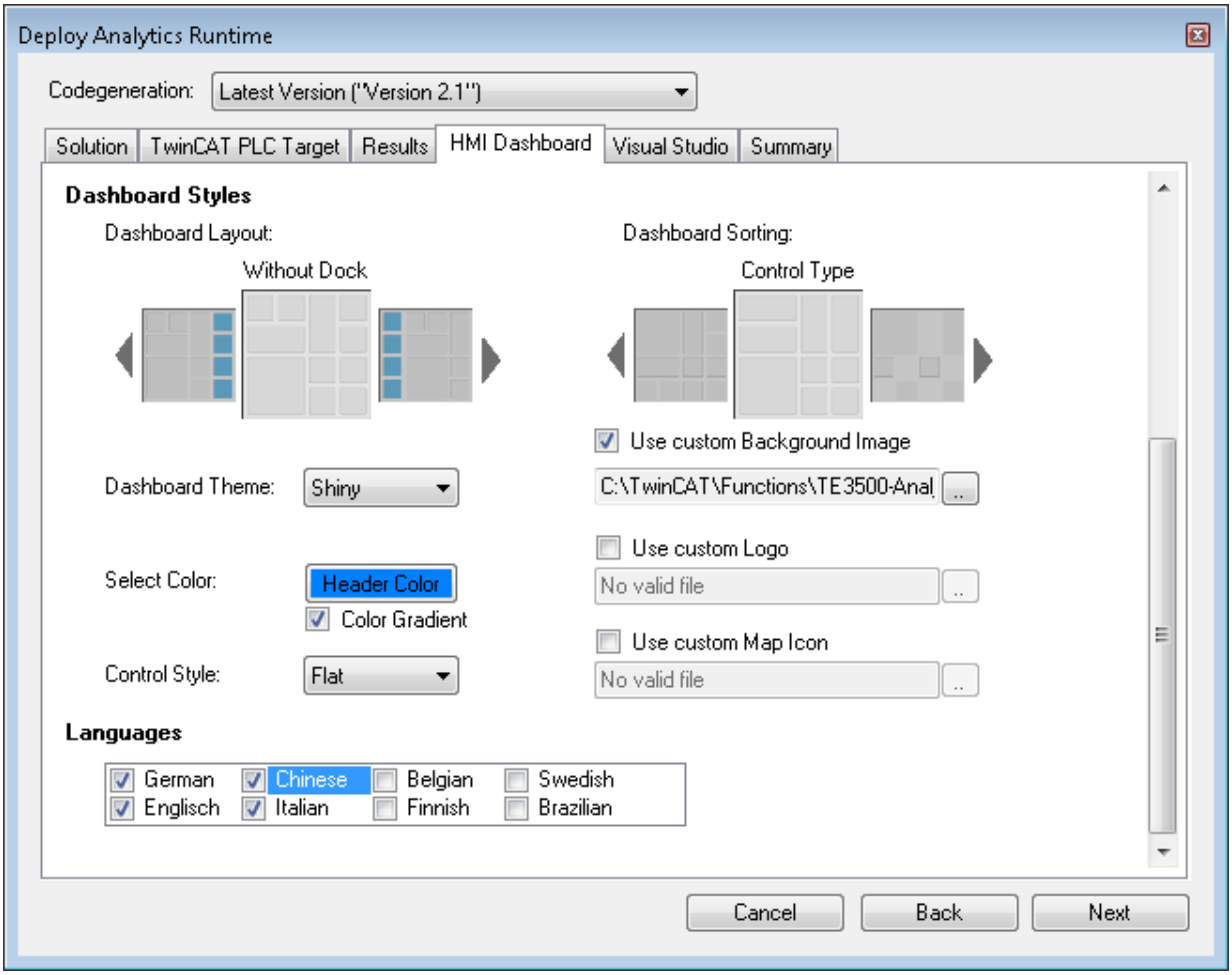


Ein Downsampling der Ergebnisse ist durch die Festlegung einer Zykluszeit möglich. Der nächste Karteireiter ist für das **HMI-Dashboard** vorgesehen. Voraussetzung für die automatische Generierung

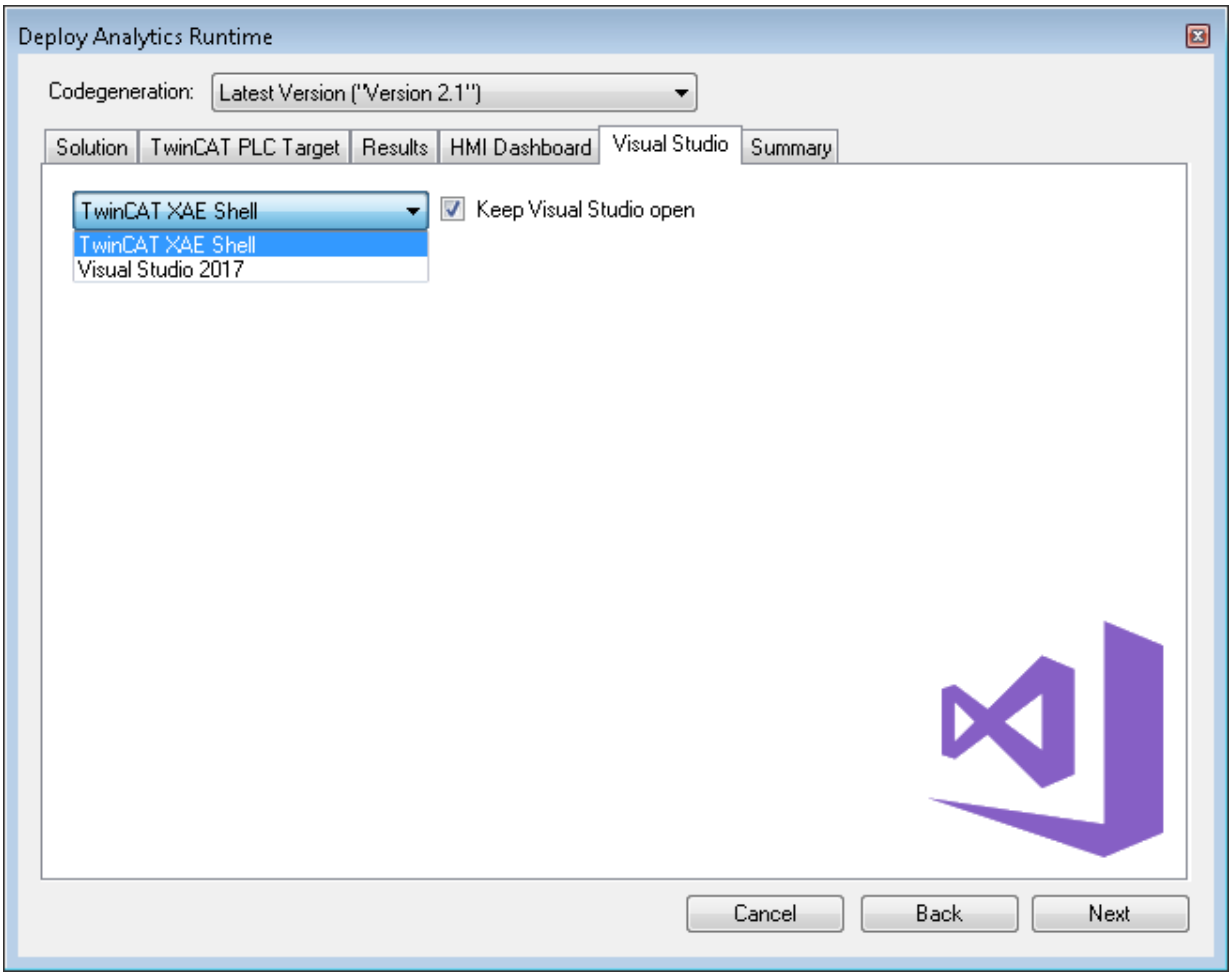
des Dashboards ist die Auswahl von HMI Controls für die entsprechenden Algorithmen, deren Ergebnisse im Dashboard angezeigt werden sollen.



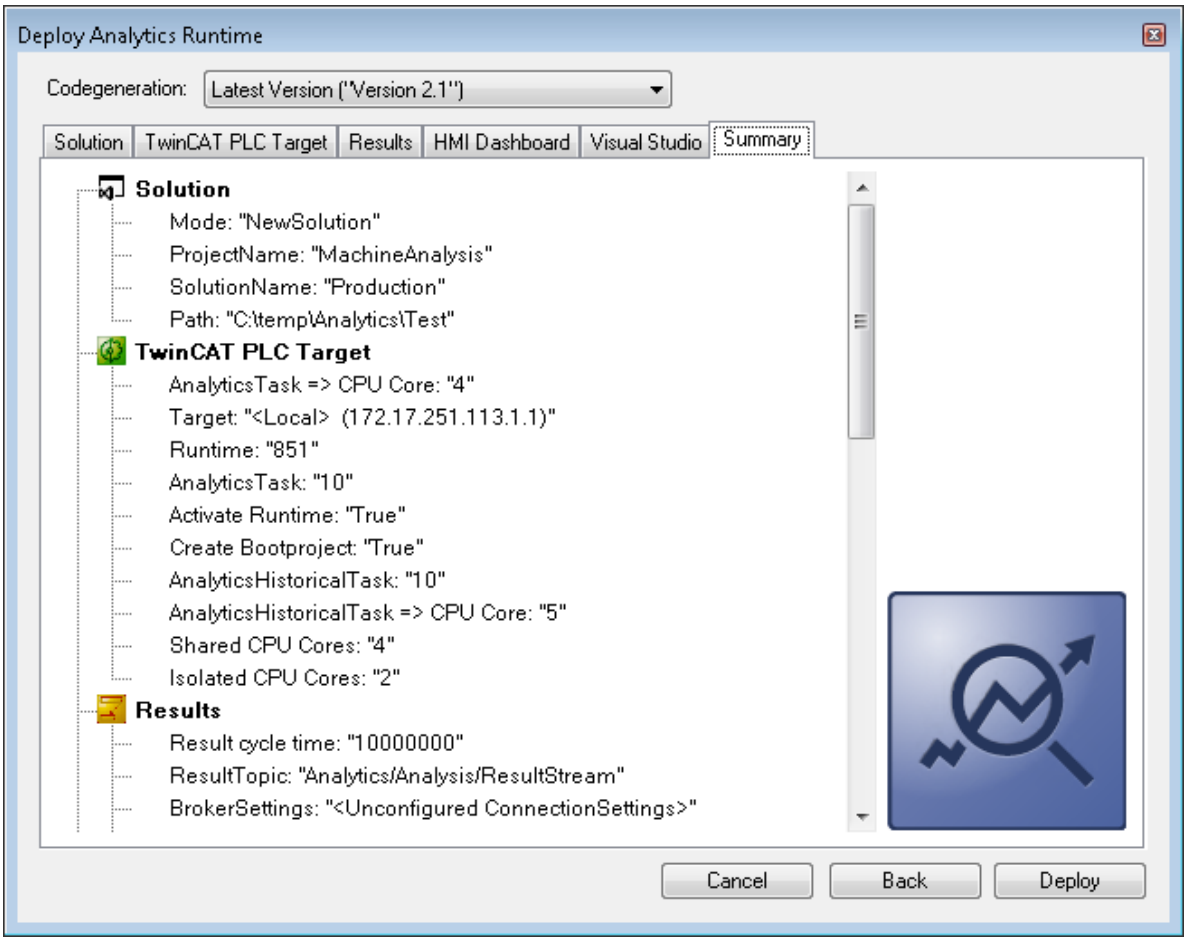
4. Sie können verschiedene Optionen für Ihr Analytics Dashboard auswählen, wie eine Startseite mit einer Karte, Layouts, Sortieralgorithmen, eigene Farben und Logos. Wenn Sie mehrere Sprachen für die Analytics Controls auswählen, wird auch ein Menü zur Sprachumschaltung generiert.



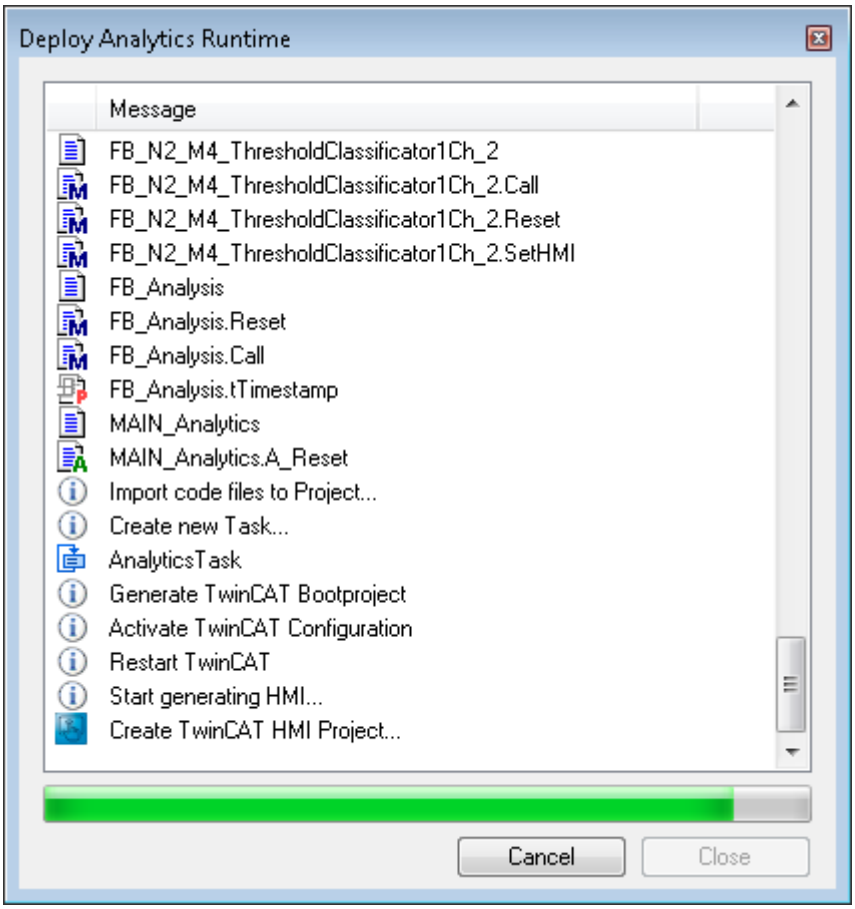
5. Wählen Sie eine der installierten Versionen von Visual Studio® aus, und, ob die Instanz sichtbar starten soll oder ob sie nur eingerichtet und im Hintergrund aktiviert werden soll.

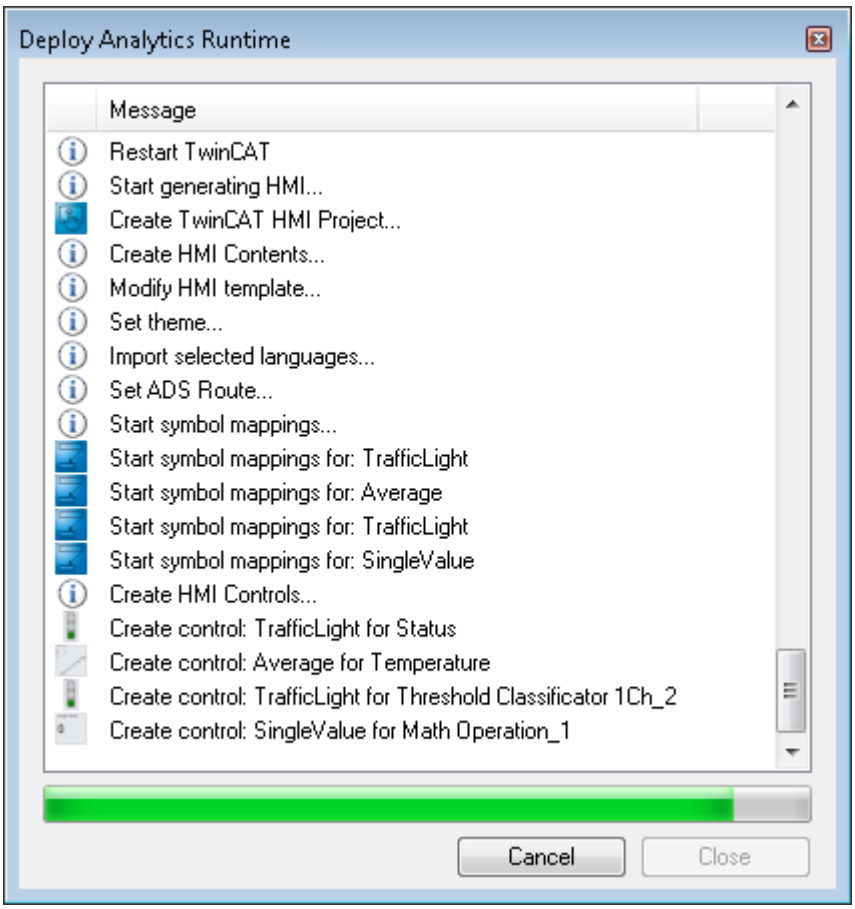


⇨ Zuletzt finden Sie eine Übersicht.

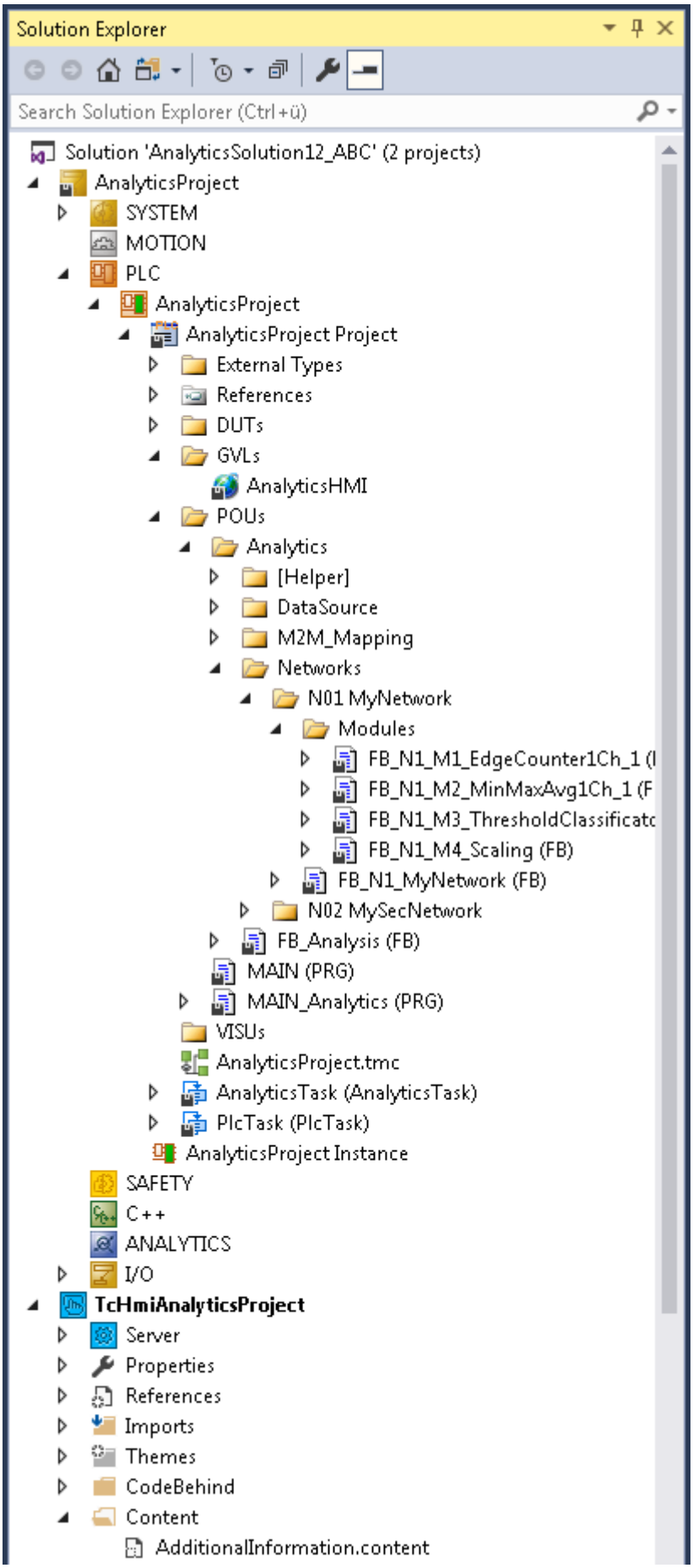


6. Nun können Sie auf die Schaltfläche **Deploy** klicken, um den Generierungsprozess zu starten. Das SPS-Projekt und das HMI-Dashboard werden nun generiert.





⇨ Nach der Meldung „Deploy Runtime succeeded“ finden Sie eine neue Visual Studio®/XAE Shell-Instanz auf Ihrem Desktop. Die neue Projektmappe und beide Projekte werden erstellt.



5 Technische Einführung

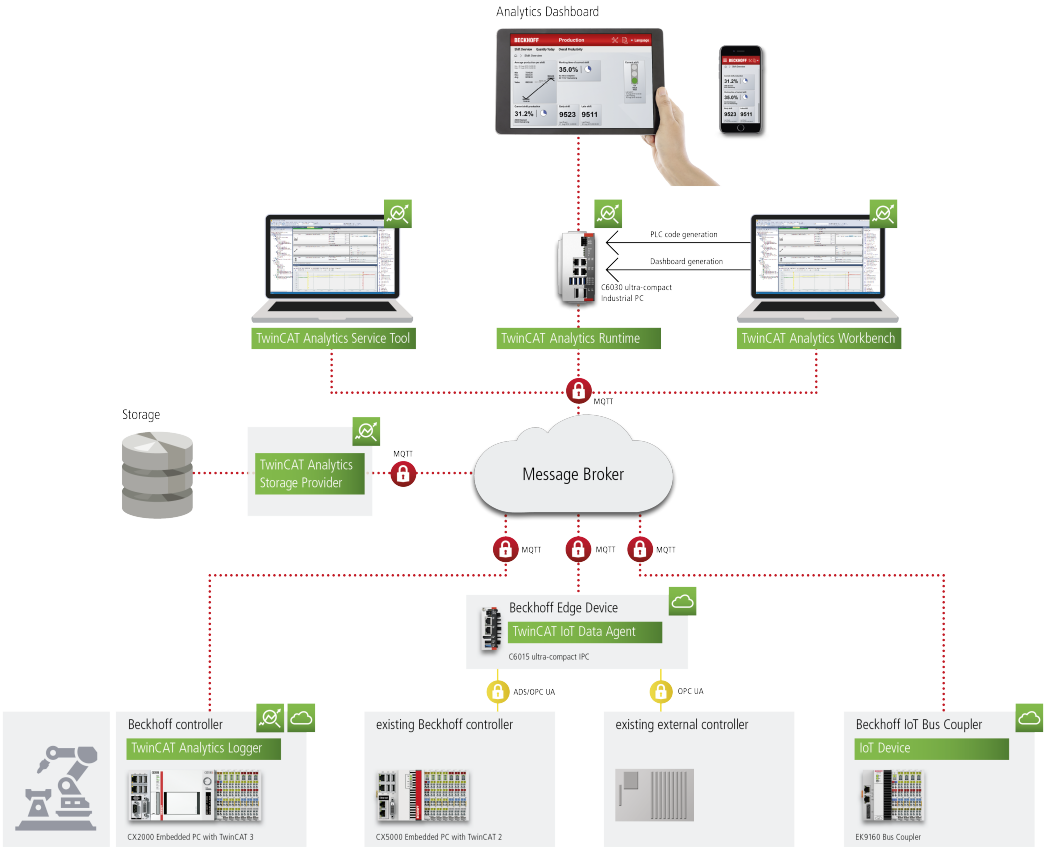
5.1 Grundkonzept

Das folgende Bild zeigt das Grundkonzept von TwinCAT Analytics von der Datenquelle bis hin zum Analytics Dashboard auf Basis von TwinCAT 3 HMI. Die Kommunikation in einem Analytics-Szenario wird durch das IoT-Kommunikationsprotokoll MQTT realisiert.

Datenquellen:

Gegenwärtig gibt es drei verschiedene Datenquellen für TwinCAT Analytics. Alle diese Quellen können mit dem spezifischen Binärdatenformat von TwinCAT Analytics kommunizieren. Dieses Format ist notwendig, um eine hohe Leistung zu erreichen.

- TwinCAT 3 Steuerung mit TF3500 TwinCAT Analytics Logger
- TwinCAT 2, TwinCAT 3 und Fremdsteuerung zusammen mit einem Gateway des TF6720 TwinCAT IoT Data Agent
- Alle EK9160 IoT-Buskoppler



Speicherung:

Mit TwinCAT Analytics können Live- und historische Daten analysiert werden. Der TwinCAT Analytics Storage Provider ist die Schnittstelle zwischen einem nativen MQTT-Message-Broker und verschiedenen Speichern. Zur Speicherung unterstützt TwinCAT Analytics einen Azure Blob-Speicher und eine Microsoft SQL-Datenbank. Die Konfiguration der Speicher wird automatisch vom Storage Provider durchgeführt. Daher ist es nicht notwendig, klassische SQL-Befehle zur Realisierung der Kommunikation zu verwenden. Zudem muss der Benutzer keine spezielle Tabellenstruktur einrichten.

Analyse:

Für Servicetechniker und Maschineninbetriebnahme

Das TE3520 TwinCAT Analytics Service Tool ist das perfekte Hilfsmittel für Experten, die TwinCAT Analytics-Datenquellen analysieren möchten. Es ist in Microsoft Visual Studio® integriert. Der Benutzer kann seine Analysekonfiguration in einem grafischen Konfigurator vornehmen und dabei aus einer Vielzahl verschiedener Algorithmen wählen. Auch eine parallele Interaktion mit dem Scope View ist möglich. Signifikante Werte können einfach per Drag-and-Drop aus dem Konfigurator in den Datenstrom unseres Scope Views gezogen werden.

Für kontinuierliche 24/7 Maschinenanalyse

Die TE3500 TwinCAT Analytics Workbench hat den gleichen Funktionsumfang wie das Service Tool. Darüber hinaus ist es möglich, auf der Grundlage der im Konfigurator realisierten Analysekonfiguration automatisch einen SPS-Code mit zugehörigem HMI-Dashboard zu generieren. Der SPS-Code ist einsatzbereit, sodass Sie die Datenanalyse sofort wie im Konfigurator starten können. Falls nötig, nun jedoch für 24 Stunden an 7 Tagen pro Woche. Der automatisch generierte Code kann zusammen mit dem Dashboard in die TF3550 TwinCAT Analytics Runtime heruntergeladen werden. Alternativ ist ein Download des reinen SPS-Projektes, bei Erstellung ohne HMI-Dashboard, in die Analytics Runtime Base TF3551 (ohne HMI Server) möglich. Beide Laufzeit Produkte sind im Wesentlichen Lizenz-bundle und können auf einem klassischen IPC oder Embedded-PC, aber auch in einer virtuellen Maschine ablaufen.

Produkte:

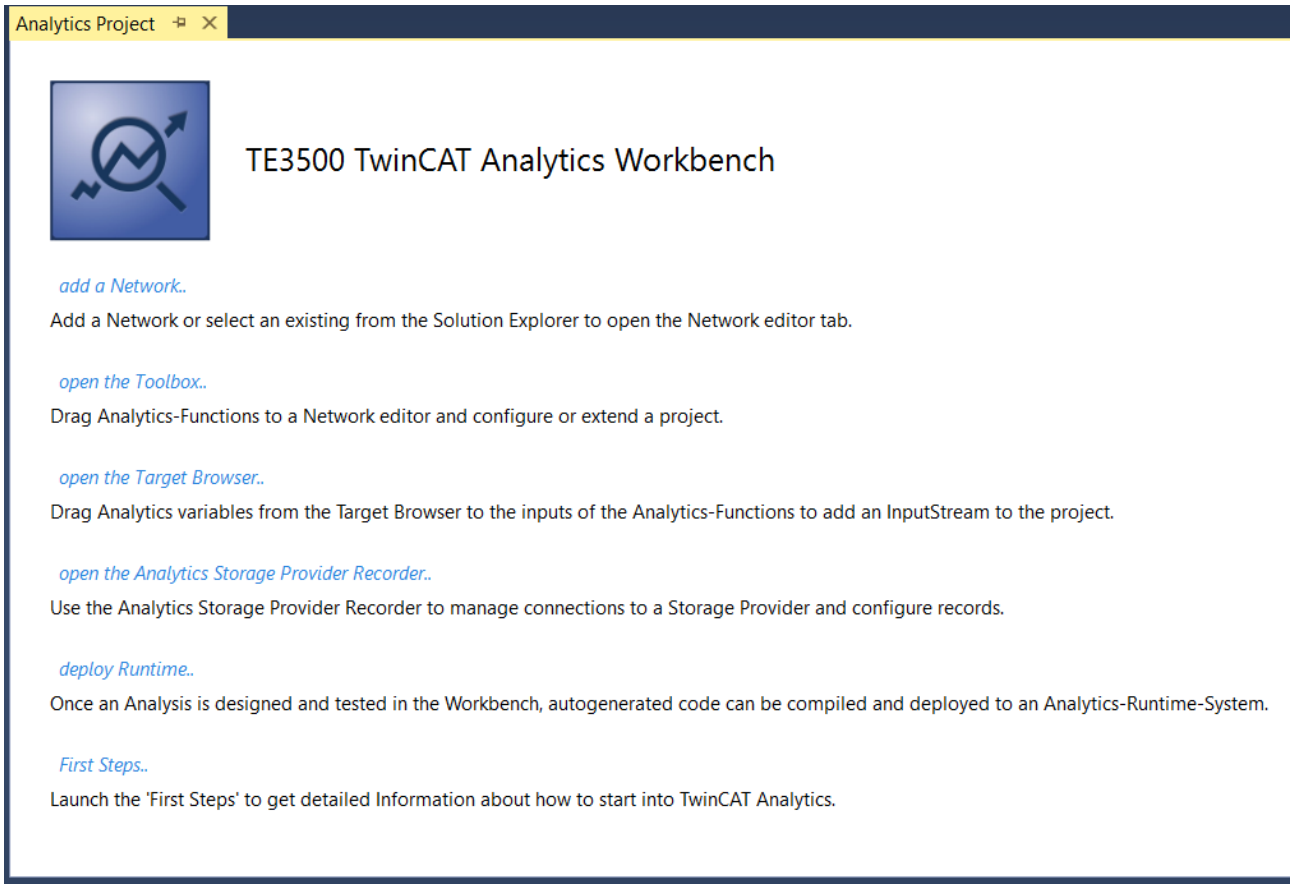
Der Workflow von TwinCAT Analytics umfasst verschiedene Einzelprodukte. Siehe hierzu die folgende Liste mit allen Produkten.

Produktnummer	Produktname
TE3500	Analytics Workbench
TE3520	Analytics Service Tool
TF3500	Analytics Logger
TF3510	Analytics Library
TF3520	Analytics Storage Provider
TF3550	Analytics Runtime – inclusive HMI Server und Client Pack
TF3551	Analytics Runtime Base – ohne HMI
TF3560	Analytics Controller Pack 4
TF3561	Analytics Controller Pack 8
TF3562	Analytics Controller Pack 16
TF3563	Analytics Controller Pack 32
TF3564	Analytics Controller Pack 64
TF3565	Analytics Controller Pack 128
TF6720	IoT Data Agent
EK9160	IoT Coupler

Das TwinCAT Analytics Service Tool kann über einen ADS-Kanal als eine Art Scope ++ verwendet werden. Dies ist automatisch die sinnvollste Minimalkonfiguration aus dem gezeigten Grundkonzept. Für eine Entkopplung der Datenaufnahme und Analyse kann zusätzlich der Analytics Logger eingesetzt werden. Er kann Daten lokal auf dem Maschinenrechner abspeichern. Die Daten können über das Service Tool ausgewertet werden. Möchte man die Datenspeicherung nicht dezentral, sondern zentral organisieren, kann über MQTT der Analytics Storage Provider eingesetzt werden. Die hier zur Verfügung stehenden Datenquellen sind TF3500/TF6720/EK9160. Zudem ist es möglich, nur die TF3510 Analytics Library in einem TwinCAT-System zu verwenden.

6 Konfiguration

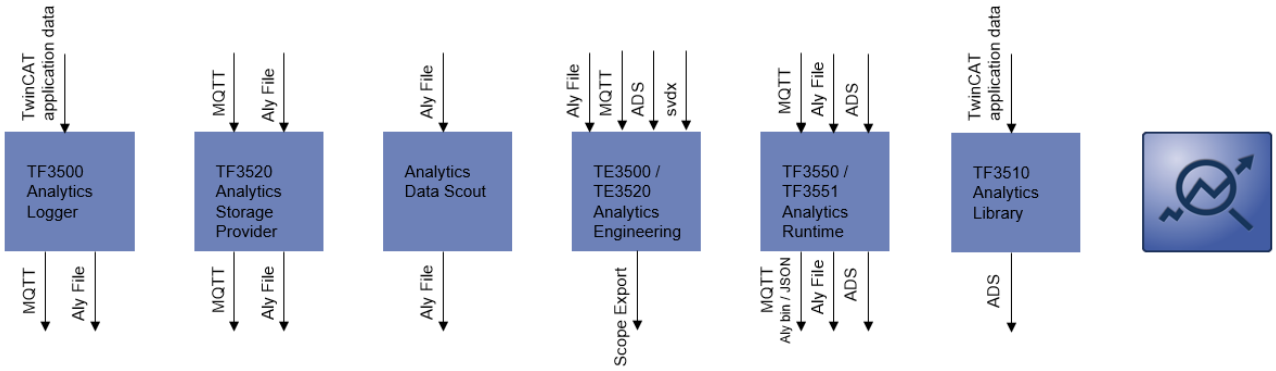
Wenn Sie eine Analytics-Konfiguration erstellen möchten, beginnen Sie auf der Analytics Project-Startseite, die im folgenden Screenshot zu sehen ist.



Auf dieser Startseite haben Sie verschiedene Möglichkeiten, fortzufahren. Diese Möglichkeiten werden nachstehend unter der jeweiligen Option beschrieben.

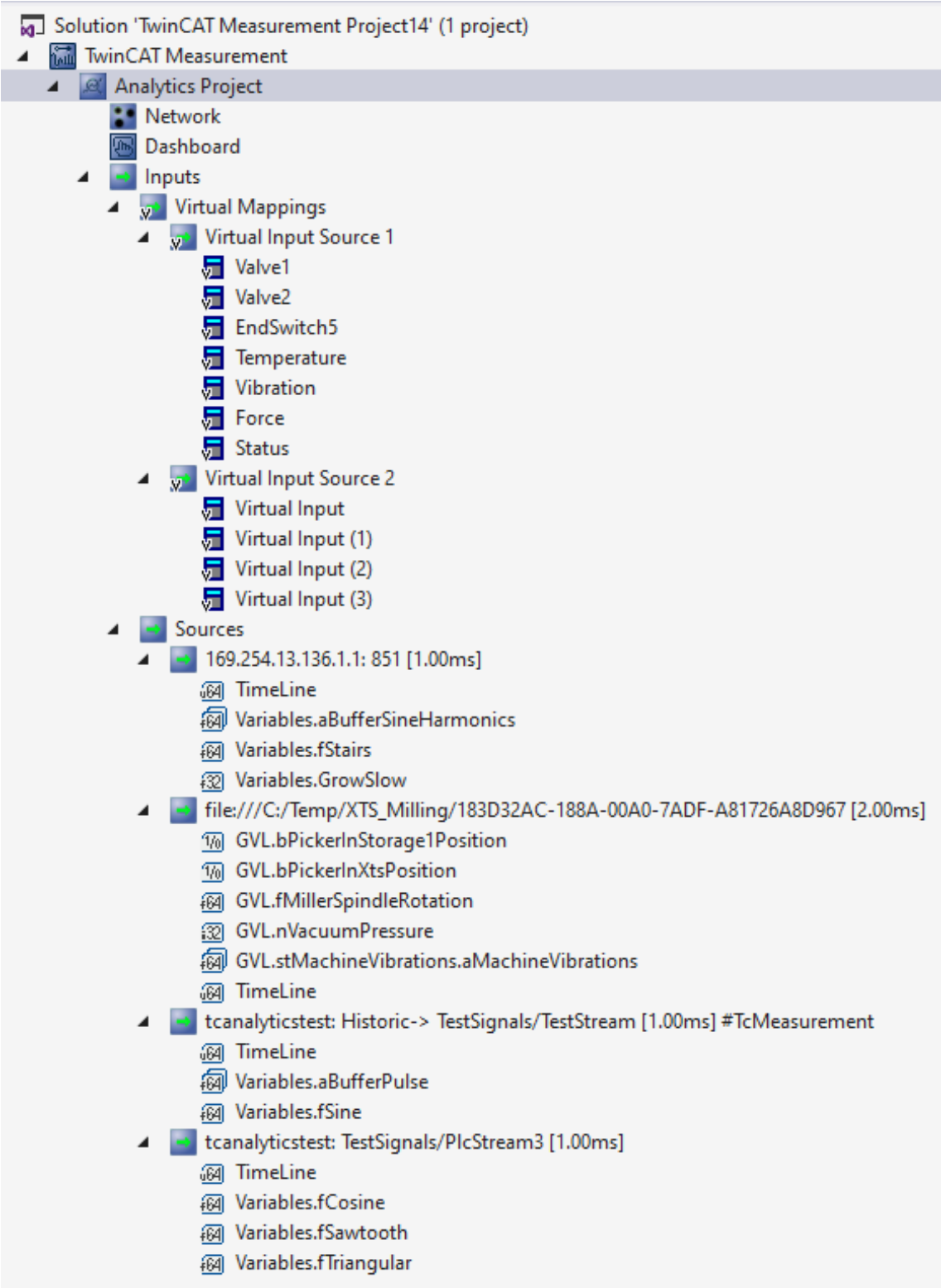
6.1 Datenquellen

Es gibt viele Datenquellen für die TwinCAT Analytics Produkte. Ebenso gibt es viele Ausgangsformate, um so einen nahtlosen Übergang von Produkt zu Produkt und von Tool zu Tool zu gewährleisten. Die folgende Grafik soll einen generellen Überblick liefern.



Am durchgängigsten sind die Analytics binär Formate in Form des Analytics Files und eines MQTT Streams.

Im Engineering selbst sehen Sie die Datenquellen im Solution Explorer im Bereich Inputs unterteilt in Virtual Input Source und Source. Die Source verkörpert dabei die tatsächliche Datenquelle, wie sie oben im Schaubild dargestellt sind. Für die Engineering Produkte TE3500 und TE3520 sind dies MQTT als Live- und Historical- Stream, Analytics File und ADS.



Die Virtual Input Source ist ein abstraktes Mapping der eigentlichen Source. Die Virtual Input Source wird für die Verknüpfung an den Eingängen der Algorithmen genutzt. Dabei ist es sehr praktisch, dass durch dieses Mapping die eigentliche Datenquelle sehr schnell und komfortabel ausgetauscht werden kann, ohne die Algorithmen neu zu verknüpfen.

An den Virtual Input Sourcen können Konfigurationen für verschiedenen Sourcen angelegt werden. Eine Konfiguration beinhaltet das Mapping der einzelnen Symbole einer Source mit einem Virtual Input. Es können auch verschiedene Konfigurationen zu einer Source angelegt werden, wo sich das Mapping der Symbole mit einem Virtual Input unterscheidet.

Bei dem Hinzufügen von Quellen und dem Ersetzen dieser, ist Ihnen optional der Source Wizard behilflich, welcher auf den nachfolgenden Seiten erläutert wird.

6.1.1 Wizard

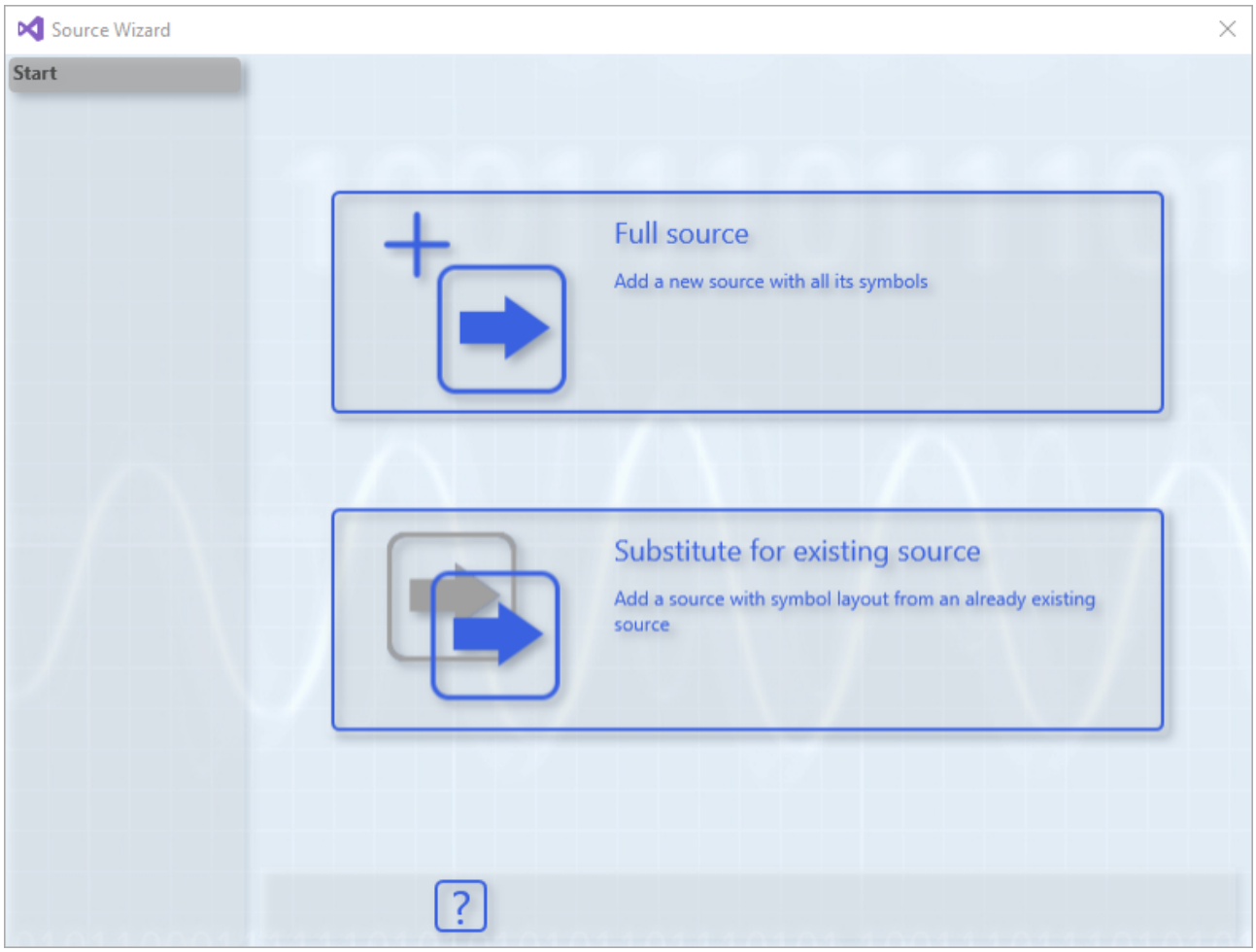
Zum Hinzufügen von neuen Datenquellen kann der Source Wizard verwendet werden. Dieses Fenster bietet die Möglichkeiten an gesamte Datenquellen hinzuzufügen oder für eine bestehende Datenquelle einen Ersatz zu finden.

Um den Source Wizard nutzten zu können muss der TwinCAT Target Browser installiert sein.

Seiten:

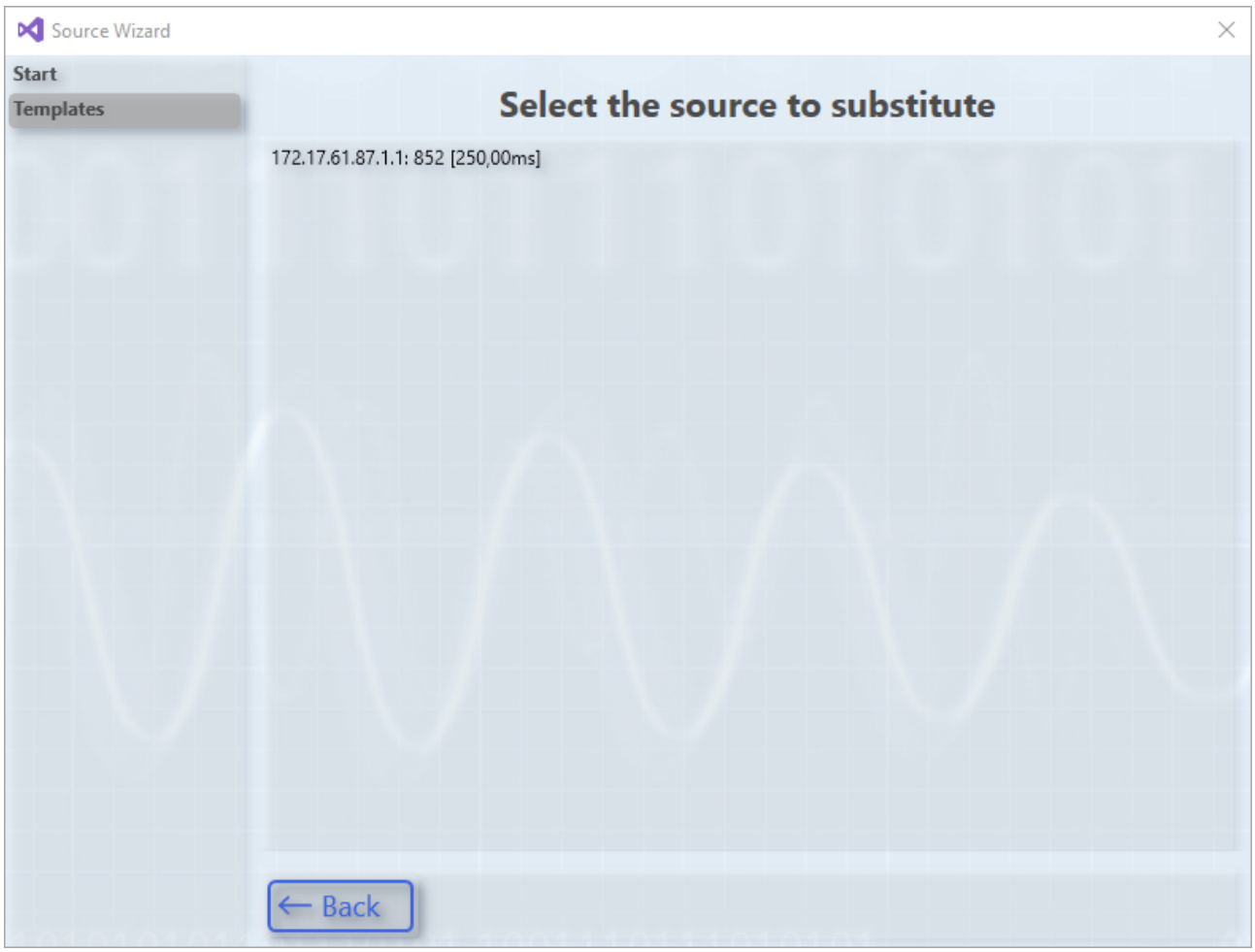
Der Source Wizard ist in mehrere Seiten aufgeteilt. Auf der linken Seite werden die bereits gezeigten Seiten aufgelistet. Mithilfe dieser Liste, sowie der Buttons **Back** und **Next** können bereits geöffnete Seiten erneut gezeigt werden.

Start



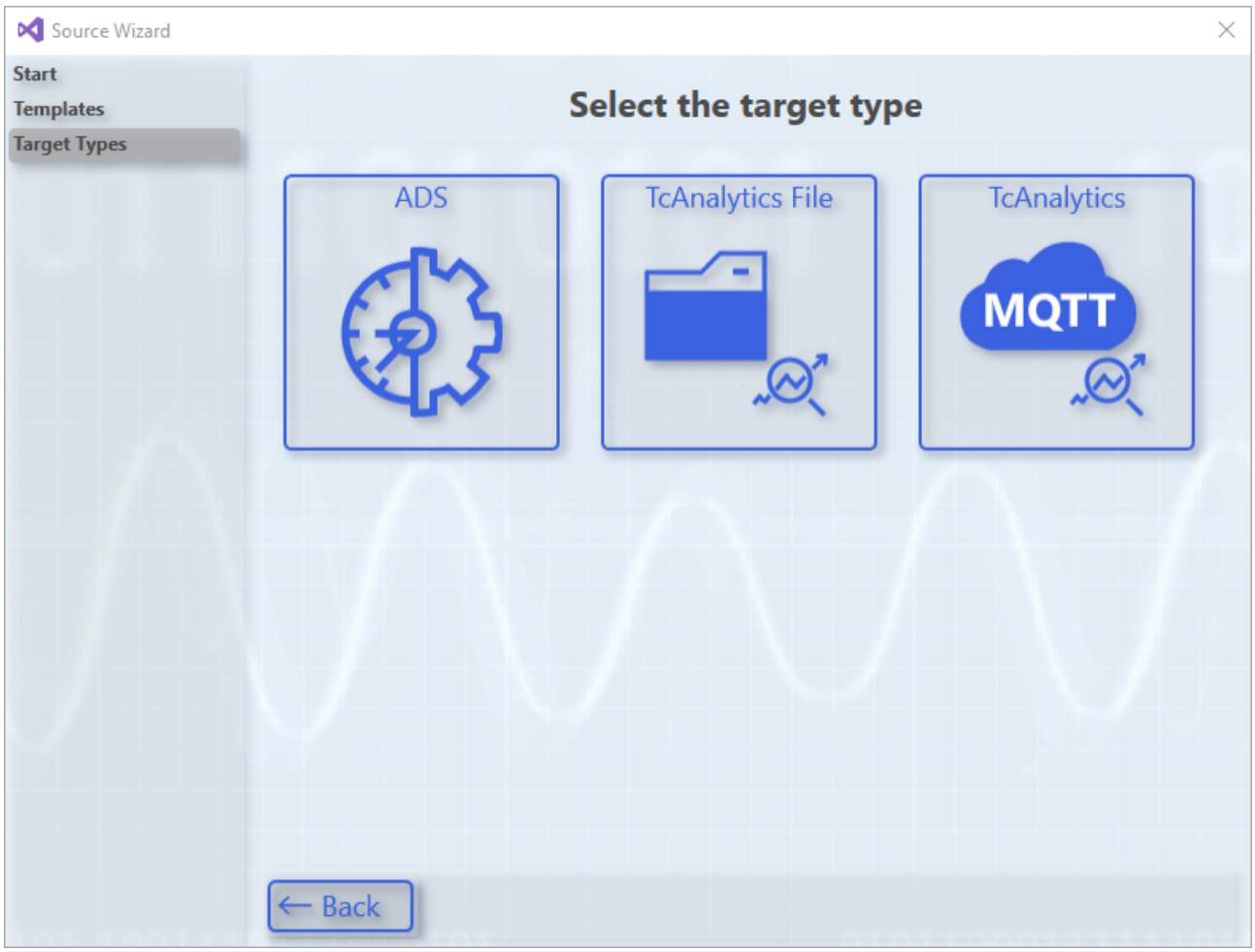
Auf dieser Seite kann ausgewählt werden ob eine gesamten Symbole einer Quelle hinzugefügt oder nur Symbole, die anhand einer schon vorhandenen Quelle gesucht werden, hinzugefügt werden sollen. Die Option zum Ersetzten einer Quelle kann nur gewählt werden, wenn bereits eine Quelle im Projekt vorhanden ist.

Vorlagen



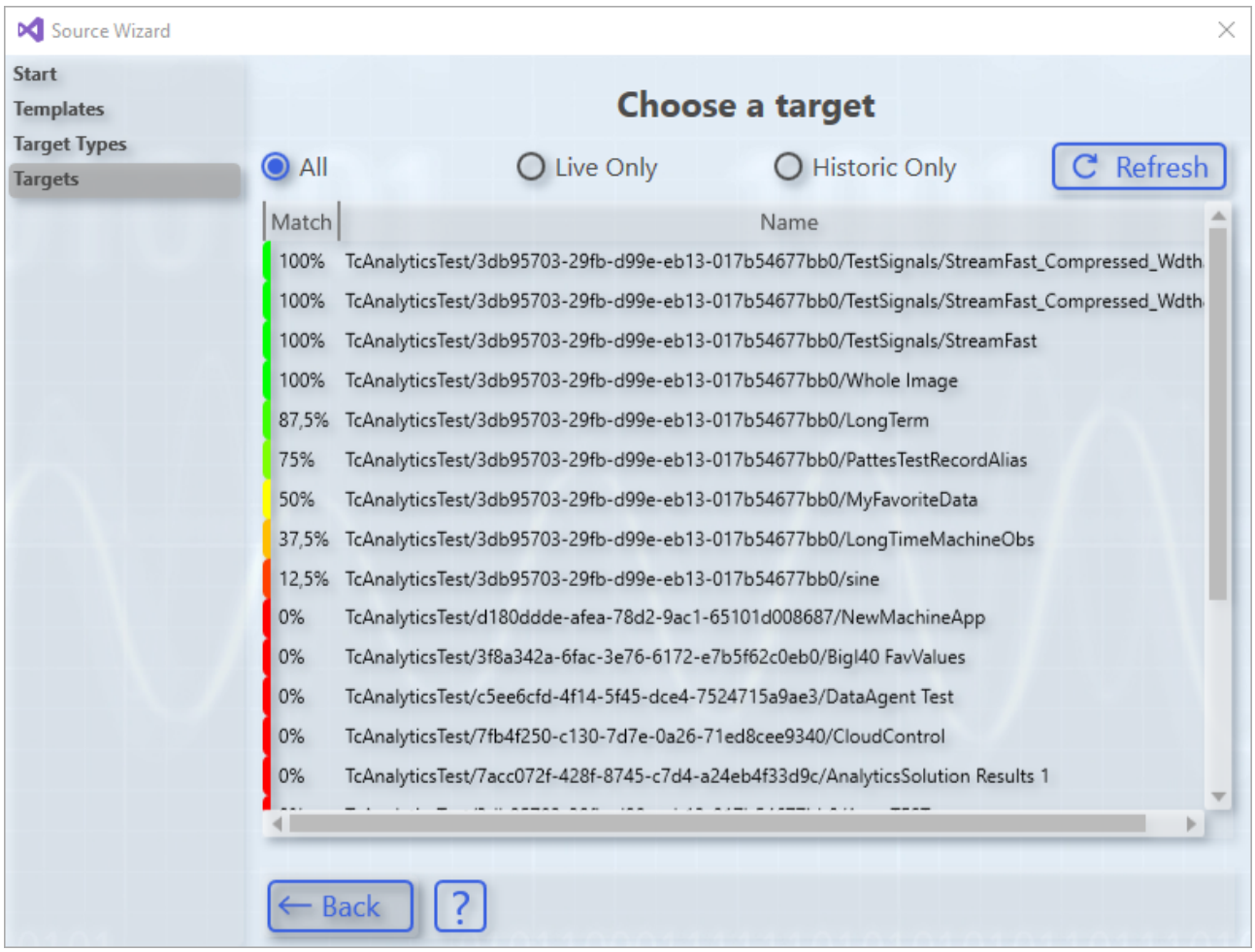
Diese Seite wird nur angezeigt, wenn auf der Start Seite die Option Ersetzten ausgewählt worden ist. Hier werden bereits vorhandenen Quellen des Projektes angezeigt. Zum Fortfahren muss eine Quelle ausgewählt werden die dann als Vorlage für die neue dient.

Quell Typen



Auf dieser Seite werden verschiedene Quell Typen angezeigt, wie z.B. ADS oder Analytics File. Die Anzahl der Typen liegt von den im Target Browser installierten Extensions ab, die mit dem Source Wizard kompatible sind.

Quellen



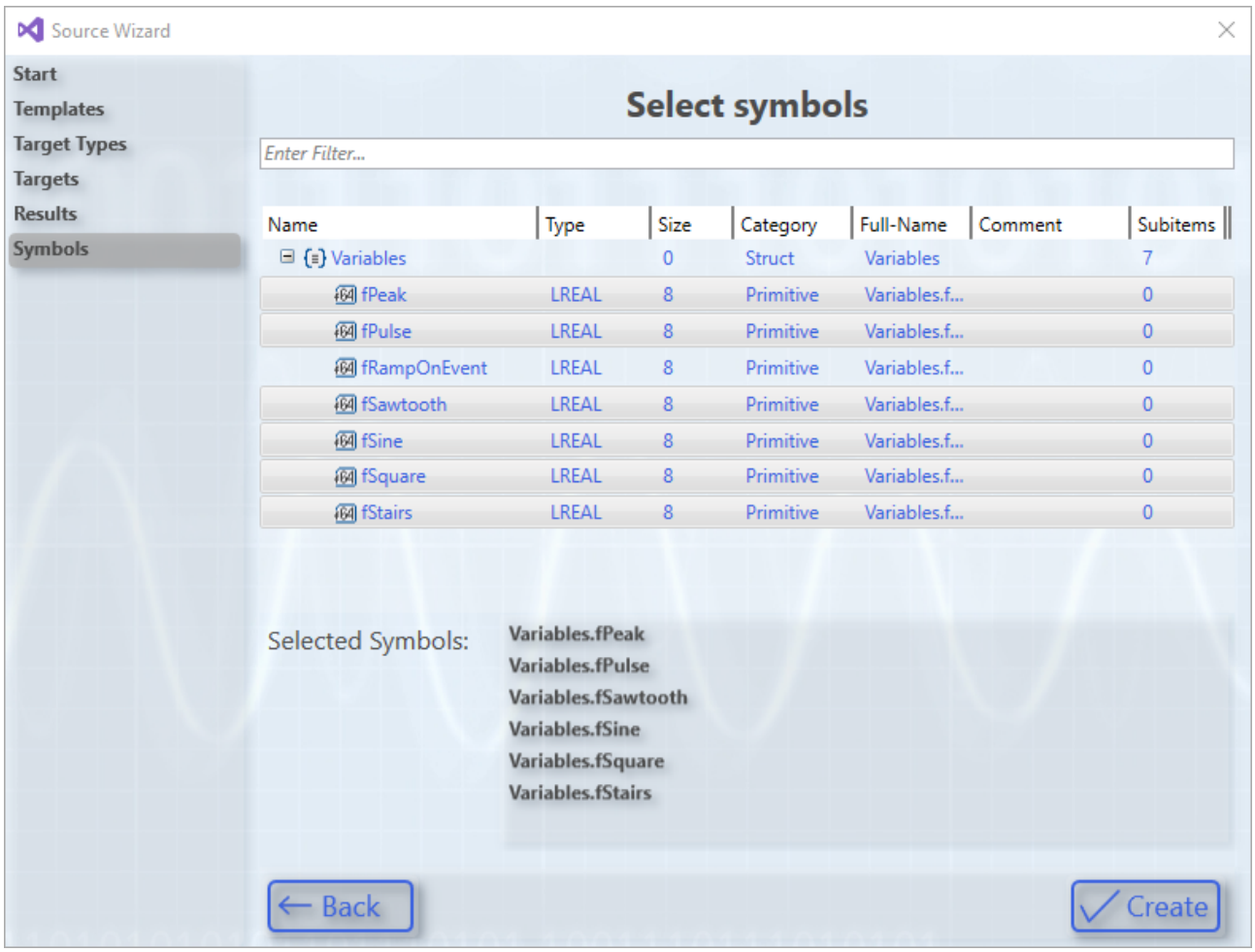
Nach der Auswahl eines Typen werden seine verfügbaren Quellen angezeigt. Um weitere Quellen finden zu können müssen im Target Browser neue Quellen hinterlegt werden. Wenn auf der Start Seite die Ersetzen Option ausgewählt worden ist, werden neben dem Namen der Quellen auch die Übereinstimmung zur Vorlage, die auf der Vorlagen Seite ausgewählt worden ist, angezeigt. Die Übereinstimmung sagt aus wie viele der Symbole aus der Vorlage gefunden worden sind. Außerdem kann die Liste gefiltert werden, sodass nur live oder nur historische Quellen angezeigt werden, außer wenn es nur historische oder nur live gibt.

Ergebnisse

Ergebnis Seite bei der Nutzung von Ersetzen

Die Ergebnis Seite listet alle Symbole auf, die hinzugefügt werden würden. Außerdem werden die Symbole der Vorlagen angezeigt, die nicht gefunden worden sind (diese Liste entfällt, wenn auf der Start Seite die Option zum Hinzufügen ganzer Quellen ausgewählt worden ist). Mit einem Klick auf **Create** wird der Wizard beendet. Alle Symbole in der „Matches“ List werden dem Projekt hinzugefügt. Falls die gewählten Symbole nicht den Anforderungen entsprechen kann „Select Symbols“ geklickt werden, dies öffnet dann die Symbole Seite.

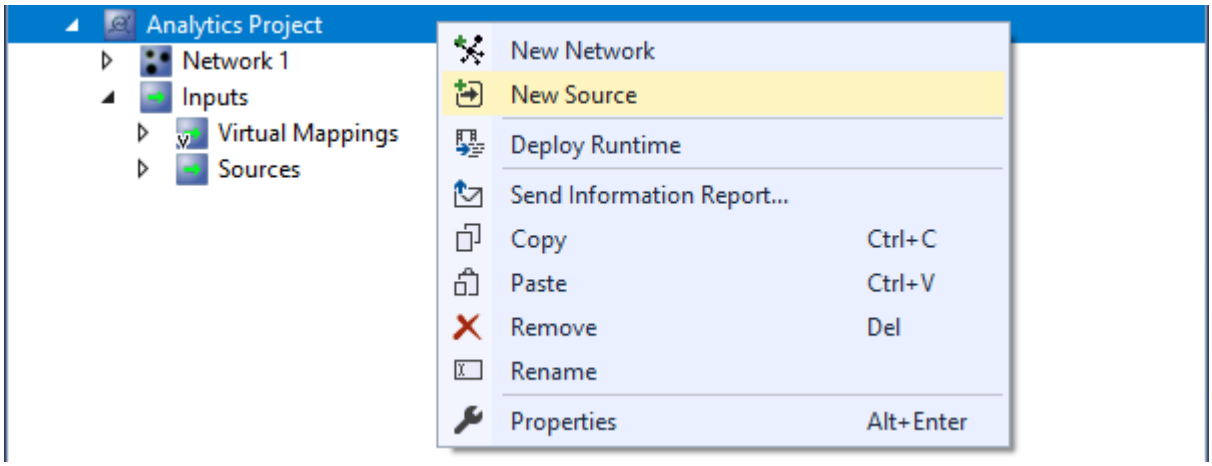
Symbole



Wird diese Seite geöffnet werden alle Symbole der gewählten Quelle aufgelistet, dabei sind schon die Symbole hervorgehoben die auch auf der Ergebnis Seite als gefunden gelistet sind. Zudem sind alle derzeitig hervorgehobenen Symbole unterhalb des Symbol Baums aufgelistet. Hier können nun die gewünschten Symbole ausgewählt oder nicht notwendige Symbole abgewählt werden. Bei Klicken des „Create“ Buttons wird der Wizard beendet und die ausgewählten Symbole werden dem Projekt hinzugefügt.

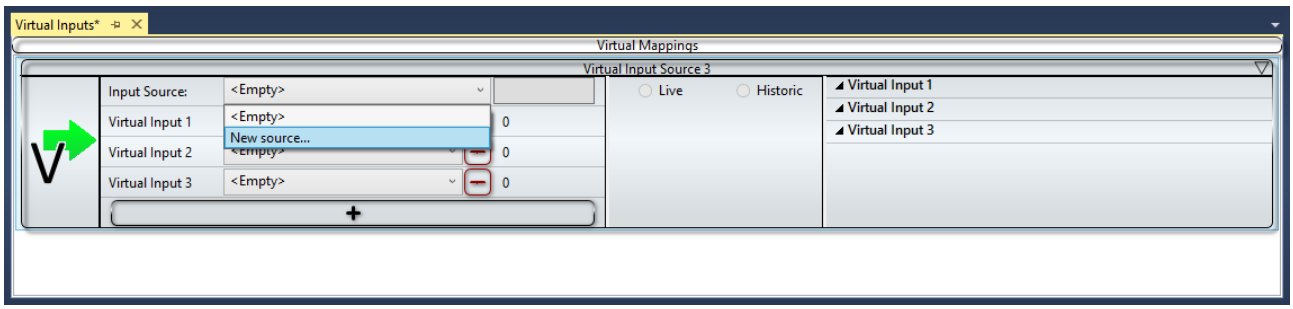
Aufruf:

Kontextmenü



Durch das Rechtsklicken auf das Analytics Projekt oder eines darunter liegenden Knotens im Projektmappen Explorer wird das Kontextmenü geöffnet und dort gibt es den Eintrag **New source**. Wird dieser ausgewählt öffnet sich der Wizard. Nach erfolgreichem Abschluss wird die gewünschte Quelle dem Projekt hinzugefügt.

Virtuelle Quelle



In einem Modul für eine Virtuelle Quelle kann in der Combobox **Input Source** der Eintrag **New source…** gefunden werden. Wird dieser ausgewählt öffnet sich der Wizard. Bei diesem Weg den Wizard zu öffnen wird nicht bei der Start Seite gestartet, sondern auf der Vorlagen Seite, wobei die derzeitig ausgewählte Vorlage die Quelle der virtuellen Quelle ist. Nach erfolgreichem Abschluss wird die neue Quelle in der virtuellen Quelle ausgewählt und die Symbole können im Auswahlmodus zu den virtuellen Inputs zugewiesen werden.

Abfrage

Wenn ein Symbol auf eine Funktion per Drag Drop hinzugefügt wird und dieses Symbol zu keiner im Projekt vorhandenen Quelle gehört, dann kann zwischen drei Möglichkeiten gewählt werden. Es kann eine neue virtuelle Quelle hinzufügt und um die Quelle gleichzeitig mit den anderen Quellen zu verwenden, es kann eine Quelle von einer bereits existierenden virtuellen Quelle geändert werden oder der Source Wizard kann geöffnet werden. Wenn der Wizard geöffnet wird gibt es je nach Auswahl auf der Start Seite eine unterschiedliche Aktion, die nach Abschluss des Wizards durchgeführt wird. Wenn eine ganze Quelle hinzugefügt wurde, wird neben der Quelle auch eine virtuelle Quelle angelegt, sodass sie gleichzeitig mit anderen benutzen werden kann. Wenn Ersetzen ausgewählt wurde wird die neue Quelle hinzugefügt und in einer virtuellen Quelle ausgewählt, dann können die Symbole im Auswahlmodus zu den virtuellen Inputs zugewiesen werden.

6.2 Data Scout

Der TwinCAT Analytics Data Scout ist ein sehr wichtiges Tool im Bearbeitungsablauf von TwinCAT Analytics. Es dient dem Daten-Sichten und kann Daten bzw dessen Aufnahmen manipulieren. Beispielsweise können unterschiedliche Aufnahmen zu einer großen Aufnahme zusammengefügt oder nicht benötigte Datenabschnitte entfernt werden. Die so künstlich entstandene neue Aufnahme kann als Datenquelle für die eigentliche Analyse genutzt werden. Der Data Scout ist ausschließlich für die Bearbeitung bereits aufgenommener Daten zuständig. Das Eingangsformat ist Analytics File, welches gleichzeitig auch das Ausgangsformat ist. Das Engineering Tool ist mit TE3500 Analytics Workbench und TE3520 Analytics Service Tool verfügbar.

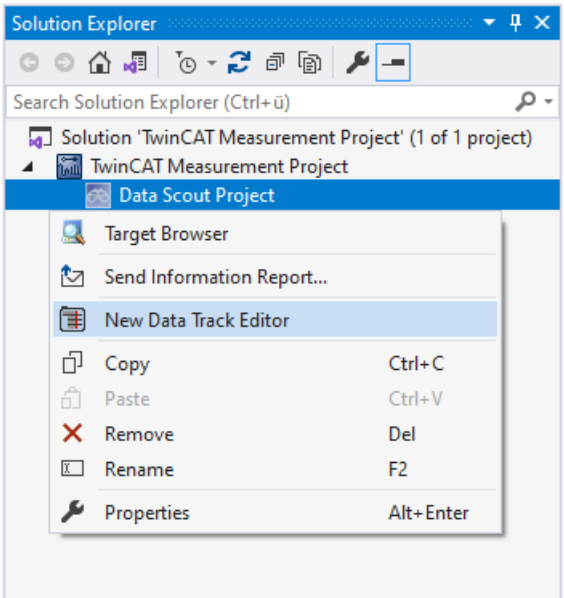
Das Data Scout Projekt kann in der Engineering Umgebung unterhalb des Measurement Projektes angelegt werden und wird auf den folgenden Seiten genauer erläutert.

6.2.1 Data Track Editor

Der Data Track Editor ist Teil des Analytics Data Scouts. Er kann dazu genutzt werden, um Daten zu sichten, aus verschiedenen Quellen Daten zusammenzufügen und zum Ausschneiden von Teilen der Daten oder ganzen Symbolen.

Erstellen

Ein neuer Track Editor kann zu einem Analytics Data Scout hinzugefügt werden indem auf New Data Track Editor im Kontext Menüs eines Data Scout Projekts geklickt wird.



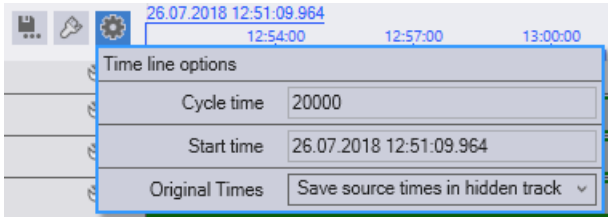
Aufbau

1. Tracks [▶ 56] stellen die Symbolik und Daten im Data Track Editor dar.
2. Das Chart [▶ 63] wird genutzt um die in den Tracks liegenden Daten als Graphen zu visualisieren.
3. Die Toolbar [▶ 64] enthält Funktionen, um z.B. den angezeigten Bereich zu ändern oder zum Ausscheiden von Bereichen.

Zum Exportieren [▶ 65] muss der Button geklickt werden.

Timeline

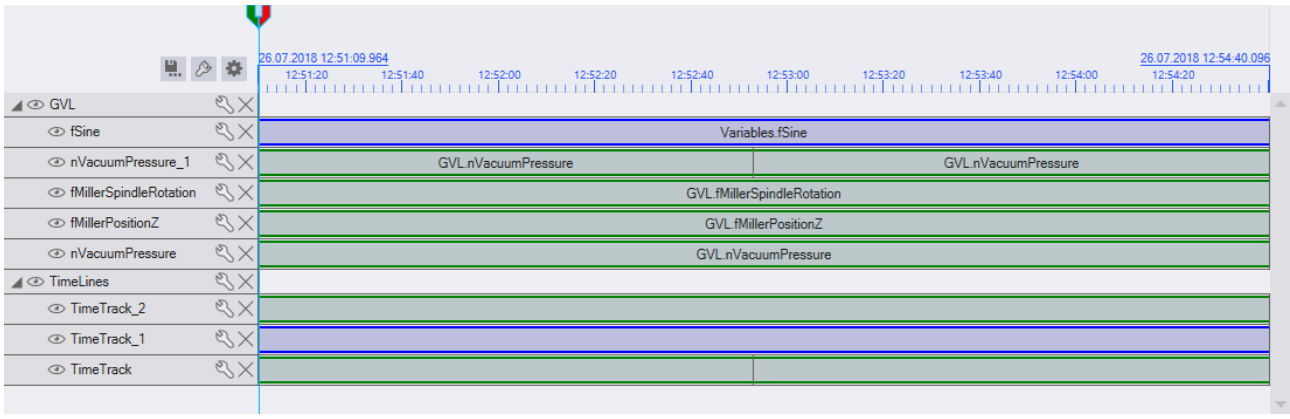
Da im Track Editor verschiedene Quellen hinzugefügt werden können, werden neue Zeiten vergeben. Der Startzeitpunkt und die Zykluszeit können in den Timeline Options eingestellt werden. Diese Zeiten werden im Export und im Chart verwendet. Die Start- und Zykluszeit der ersten hinzugefügten Quelle wird übernommen, falls diese noch nicht eingestellt worden sind.



Die Endzeit wird automatisch berechnet, hierbei wird die Track mit den wenigsten Samples verwendet.

Time Tracks

Um die originalen Zeiten der Quellen nicht zu verlieren können Time Tracks generiert werden. Wie diese generiert werden kann in den Timeline Options eingestellt werden. Es gibt drei Varianten zur Generierung von Time Tracks. Die Time Tracks können als sichtbare Tracks generiert werden, dabei werden alle in einem Container zusammengefast. Die zweite Variante ist die Time Tracks auf die gleiche Weise zu generieren, aber sie und der Container, in dem sie liegen, werden nicht im Editor angezeigt. Die letzte Variante ist es keine Time Tracks zu erstellen und die originalen Zeiten beim Export zu verwerfen.



Die Time Track Generierung funktioniert über die Datenelemente der Tracks. Für jede Kombination aus Datenelemente gibt es eine Time Track. Dabei werden nur der Zeitbereich und die Quelle, aber nicht die Symbole der Datenelemente berücksichtigt.

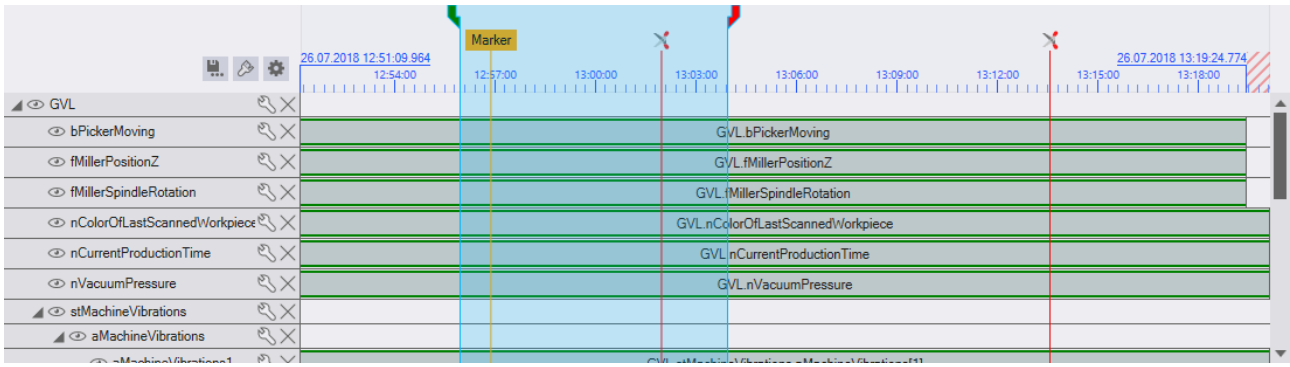
In der Beispiel Darstellung wurden die Time Tracks sichtbar generiert. Es wurden automatisch drei erstellt da in den darüberliegenden Tracks drei Kombinationen gefunden worden sind. Zum einen die Track mit einen blauen Datenelement, zum anderen eine Track mit zwei grünen Datenelementen und noch drei Tracks mit einem grünen Datenelement.

Time Tracks sind wie Single Tracks aufgebaut. Sie können Datenelemente enthalten, aber diese Datenelemente haben keine Symbole, sondern nur einen Zeitbereich und eine Quelle.

6.2.1.1 Tracks

Tracks sind Elemente des Data Tracks Editors, welche Infos zu den enthaltenen Daten und zum Aufbau der Symbolik beinhalten.

Darstellung



Der im Data Track Editor für die Tracks vorgesehene Bereich hat folgende Elemente:

	Dieser Button öffnet ein neues Track Editor Export Fenster.
	Durch Anklicken wird ein Popup geöffnet, wo Einstellungen zur Darstellung der Tracks vorgenommen werden können.
	Durch Anklicken wird ein Popup geöffnet, wo Einstellungen zur Timeline und den Time Tracks vorgenommen werden können.
	Der blaue Bereich liegt über den Tracks und stellt dar welche Daten im Chart dargestellt werden. Außerdem wird der selektierte Bereich für einige Funktionen der Toolbar verwendet. Der rote und der grüne Bereich können verschoben werden, um die Selektierung zu vergrößern oder zu verkleinern. Wenn zwischen dem roten und grünen Bereich geklickt wird, kann der selektierte Bereich verschoben werden. Am linken und rechten Rand kann die Selektierung einrasten. Dies geschieht in der Nähe von Marker und Track Enden. Um die Einrastfunktion zu ignorieren kann die Alt-Taste, während des Verschiebens, gedrückt werden. Die Selektion kann nicht kleiner als 10 Werte sein.
	Marker werden sowohl in den Track als auch im Chart angezeigt. In den Tracks können Marker durch das Anklicken des Textes umbenannt werden und durch ein Rechtsklick können Marker entfernt werden. Die Marker die keinen Text, sondern eine Schere haben, sind spezielle Marker, die zum Ausscheiden verwendet werden können.
	Die Timeline hat am linken Ende die Startzeit markiert. In gleichmäßigen Abständen folgen dann Markierungen, welche die dort anliegende Zeit anzeigen.
	Am rechten Ende der Timeline ist eine Markierung für die Endzeit. Die Endzeit wird entspricht der Startzeit plus die Anzahl von Samples von der Track mit den wenigsten Samples mal der Zykluszeit. Wenn die Anzahl der Samples in den Tracks sich unterscheiden wird, hinter der Markierung für die Endzeit, ein rot gestrichelter Bereich angezeigt. Dieser Bereich wird nicht mit exportiert, da dort nicht für alle Tracks Daten vorhanden wären.
Tracks Container	Im unteren Teil werden die Track dargestellt, wobei die Anzeige der Daten von den Tracks bei der Startzeit startet und davor wird die neue Symbolik angezeigt.

Daten Tracks

Derzeit gibt es zwei Arten von Tracks die Daten enthalten können. Die erste ist die Single Track. Diese stellt ein einzelnes Symbol dar. Die zweite ist die Multi Track. Diese kann mehrere Symbole zusammenfassen, um so die Darstellung einfach zu halten. Allerdings können in Multi Tracks z.B. die Datentypen oder Namen der enthaltenen Symbole nicht angepasst werden.

Tracks die Daten enthalten werden auf ähnlich weise dargestellt.

Auf der linken Seite sind folgende Funktionen:

	Hier kann festgelegt werden, ob die Track im Chart als Graph sichtbar sein soll oder nicht.
Track Name	Der Name der Track wird im Chart für den Graph und im Export als Symbolname verwendet. Der Name muss unterhalb eines Containers eindeutig sein und kann keine Leerzeichen enthalten.
	Durch Anklicken wird ein Popup geöffnet. Dieses zeigt die Einstellungen der Track an (siehe im Punkt Einstellungen für genaueres).
	Dieser Button entfernt die Track und alle darunter liegenden Tracks

Auf der rechten Seite werden die Datenelemente der Track angezeigt. Daten Tracks können mehrere Datenelemente enthalten, welche zur Art der Track passen (z.B. eine Single Track kann kein Array enthalten).

Container Tracks

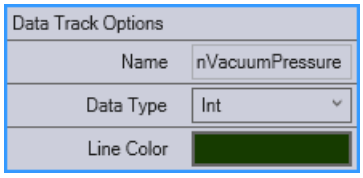
Container Tracks sind ähnlich wie Datentracks aufgebaut. Allerdings können keine Datenelemente in Container Tracks liegen, dafür können sie aber Tracks enthalten. Enthaltene Tracks werden unterhalb der

Container Track angezeigt und können mit den Buttons ein- oder ausgeklappt werden. Wenn eine Container Track sichtbar oder unsichtbar geschaltet wird, werden auch alle Sub Tracks sichtbar oder unsichtbar.

Einstellungen

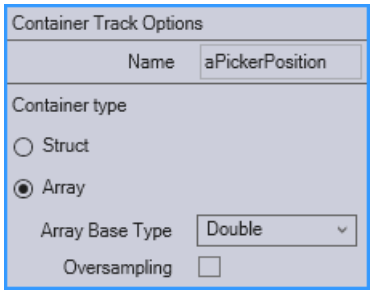
Für alle Arten von Tracks gibt es im Popup des Buttons verschiedene Einstellungen.

Für Single Tracks gibt es diese:



Datentyp	Hier kann der Datentyp der Single Track festgelegt werden. Die angezeigten sowie die exportierten Daten werden zu den hier angegebenen Typen umgewandelt. Für String Datentypen wird noch ein Textfeld eingeblendet, welches die Länge des String bestimmt. Wenn der Datentyp zu Problemen beim Umwandeln führen könnte, da ein oder mehr Datenelemente der Track einen größeren Datentype haben, wird eine Warnung ausgegeben, aber die Werte werden immer noch umgewandelt.
Linien Farbe	Legt die Farbe des Graphs im Chart fest. Beim Anklicken wird ein Color Picker angezeigt in dem die Farbe eingestellt werden kann.

Für Container Track gibt es diese Einstellungen:

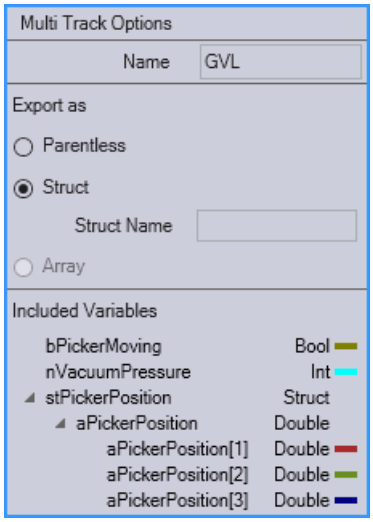


Containertyp	Eine Container Track kann eine Struktur oder ein Array sein. Um als Array erstellt zu werden müssen allerdings alle Sub Tracks zu einem Array vereinbar sein. Wenn nur Single Tracks vorliegen, kann eine Container Track zu einem Array mit einem primitiven Datentyp gemacht werden. Wenn nur Container Tracks vorliegen, welche die gleiche Struktur haben, kann die Container Track zu einem Array mit Primitiven Array oder Struktur als Basis Datentyp gemacht werden.

Jeder Containertyp hat seine eigenen Eigenschaften:

Struktur	Struktur Name	Gibt den gewünschten Namen für den Datentypen im Export vor.
Array	Basis Datentyp	Hier kann der Datentyp für die Array Elemente eingestellt werden. Wird der Datentyp hier eingestellt ändert sich der Array Basis Datentyp von Sub Container Track und der Datentype von Sub Single Tracks. Bei String Datentypen kann die Länge des Strings eingestellt werden.
	Oversampling	Diese Option ist nur verfügbar, wenn die Container Track nur Single Tracks als Sub Tracks hat. Wenn aktiv werden die Sub Tracks nicht als einzelne Graphen im Chart angezeigt, sondern die Container Track wird als ein Graph angezeigt. Dazu werden die Werte der Sub Tracks geladen und alle Werte zum gleichen Zeitpunkt werden hintereinandergelegt.

Für Multi Tracks gibt es diese Einstellungen:



Exportieren als	Es kann, ähnlich zum Containertyp der Container Track, eingestellt werden, wie die Multi Track im Export behandelt werden soll. Bei Struktur wird die Multi Track als Struktur Datentyp mit den enthaltenen Variablen als Elementen exportiert. Der gewünschte Name für die Struktur kann angegeben werden. Die Option zum Export als Array ist nur verfügbar, wenn alle enthaltenen Variablen den gleichen Datentypen haben. Der Datentyp kann dann auch nicht umgestellt werden. Die Multi Track wird beim Export zum Array mit den enthaltenen Variablen als Array Elementen. Bei Parentless wird die Multi Track im Export nicht berücksichtig und es ist so als würden die enthaltenen Variablen auf der Ebene der Multi Track liegen.
Enthaltene Variablen	Hier wird die, in der Multi Track vorliegende, Symbolik angezeigt. Alle Datenelemente der Multi Track müssen die gleiche Symbolik haben.

Datenelemente

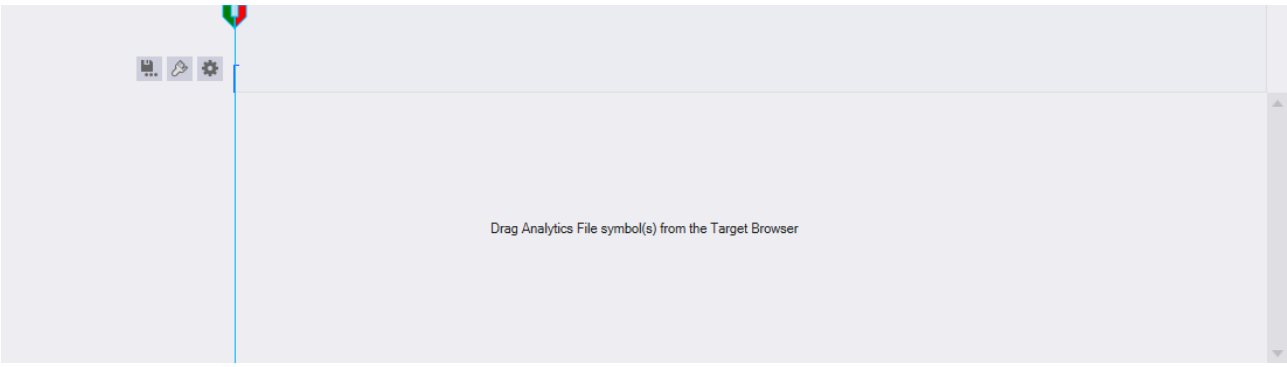
In Tracks können Datenelemente liegen. Diese enthalten eine oder mehr Variablen aus einer Quelle, sowie einen Zeitbereich aus der Quelle. Mithilfe des Zeitbereichs und der Zykluszeit der Quelle wird errechnet wie viele Samples im Dataelement liegen. Die Länge eines Datenelements wird festgelegt indem geschaut wird, welche Track die meisten Samples enthält und wie viel Platz insgesamt verfügbar ist. Die Länge ist dann abhängig von der Anzahl von Samples und wie viele Pixel einem Sample entsprechen.

Ein Datenelemente wird als Block dargestellt. Die Farbe des Datenelements hängt von der Quelle ab, aus der es stammt, so haben alle Datenelemente aus derselben Quelle die gleiche Farbe. Am Ende des Datenelements ist eine Trennlinie. Im Datenelemente wird der Originalname des Symbols aus der Quelle angezeigt. Wenn mehrere Symbole vorliegen, wird nur die Anzahl der Symbole angezeigt.

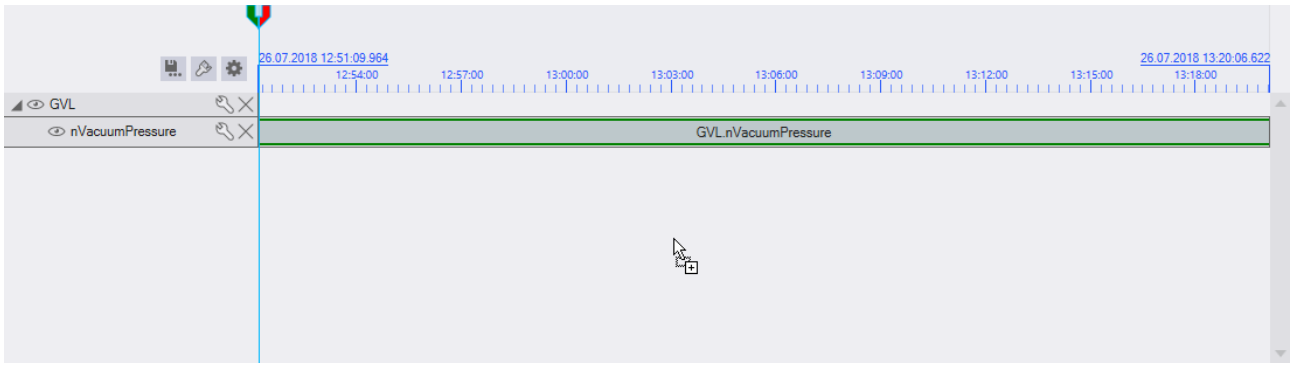
Per Rechtsklick auf ein Datenelement kann das Kontext Menü geöffnet werden. Darüber kann das Datenelement auch komplett entfernt werden, ohne die Funktionen aus der Toolbar zu nutzten.

Hinzufügen per Drag and Drop

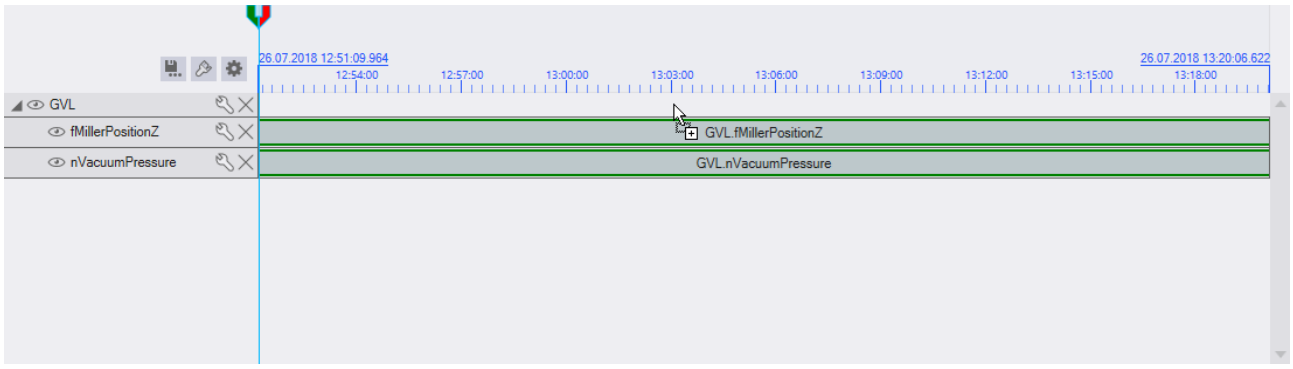
Die Tracks eines Data Track Editors werden in einen gesonderten Bereich angezeigt. Dieser ist zu Beginn leer und kann mithilfe des Target Browsers gefüllt werden.



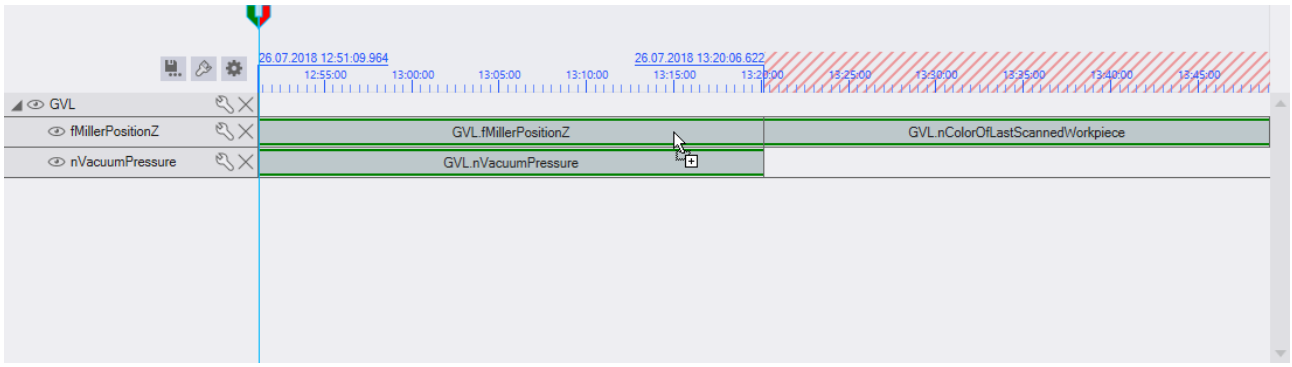
Wenn aus dem Target Browser ein oder mehr Symbole gewählt werden und in den leeren Bereich des Track Editors gezogen werden, dann wird eine neue Container Track erstellt. Zu der neuen Container Track werden alle ausgewählten Symbole als Tracks hinzugefügt.



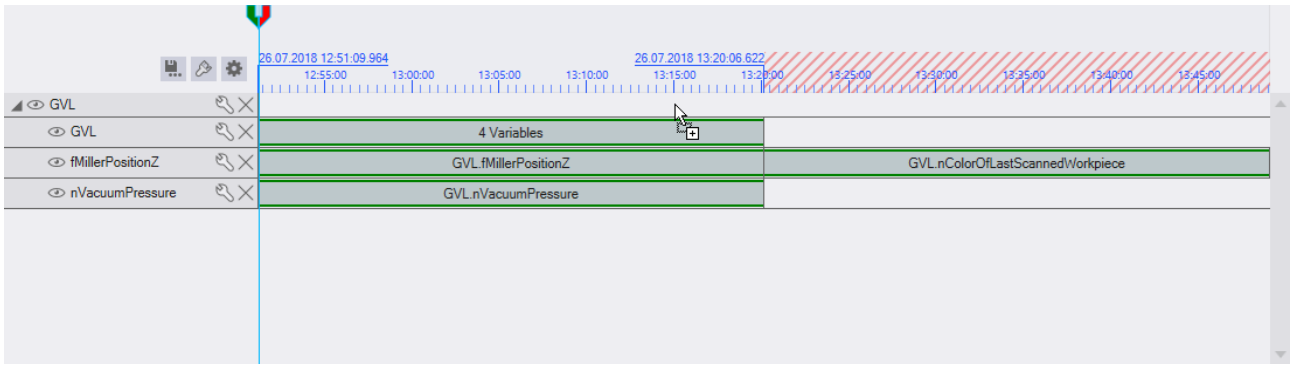
Wird auf den Daten Bereich einer Container Track, per Drag Drop, ein Symbol gezogen, wird eine neue Track unterhalb dieses Containers erstellt.



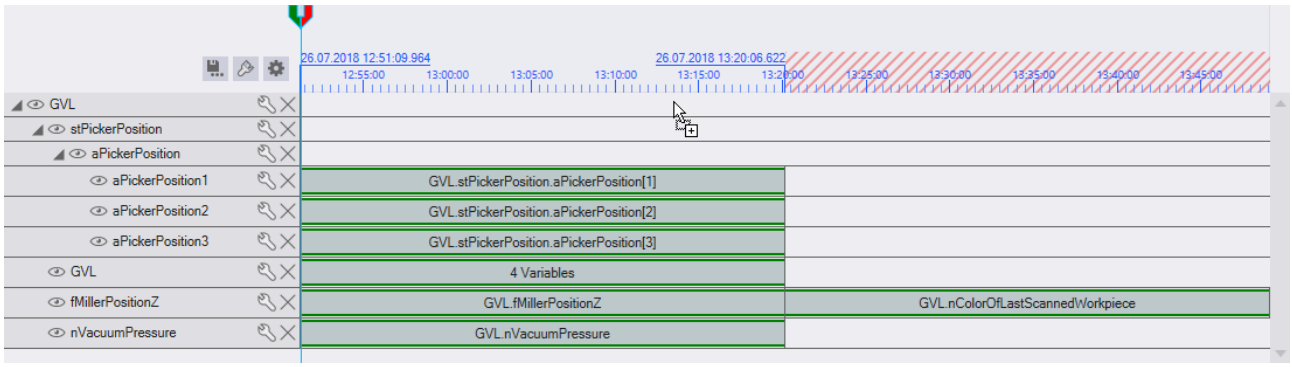
Wenn auf eine Datentrack gezogen wird, kann zu dieser Track ein neues Datenelement hinzugefügt werden. Das ist allerdings nur möglich, wenn das hinzuzufügende Symbol mit der Track kompatibel ist. Es kann keine Struktur auf eine Track mit primitiven Datentypen gezogen werden.



Um neue Multi Tracks zu erstellen kann auf eine Container Track oder in den leeren Bereich gezogen werden, während die [Strg]-Taste gedrückt ist.

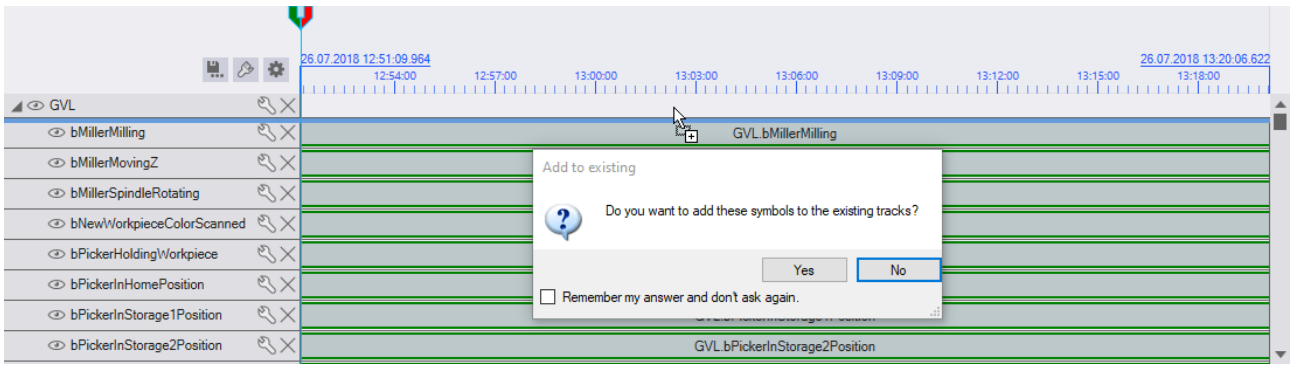


Wenn Strukturen oder Arrays hinzugefügt werden sollen, kann dies wie bei Symbolen mit primitivem Datentyp passieren. Allerdings werden für die Oberelemente von Strukturen und Arrays Container Tracks erstellt.

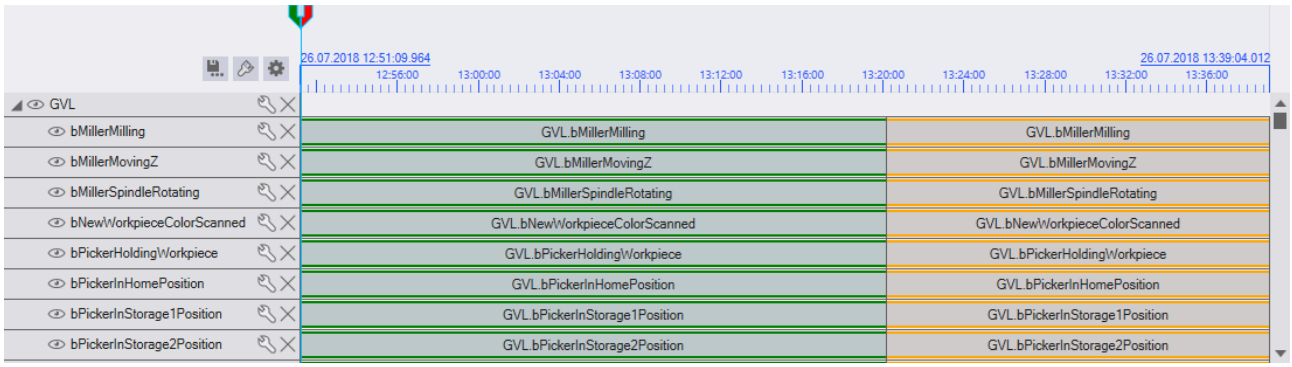


Automatisches Zusammenfügen

Um einfacher Symboliken mit gleichem Aufbau zusammenzufügen, wie zum Beispiel bei einem Analytics File mit mehreren Records, wird beim Drag Drop von mehreren Symbolen auf einen Container oder den leeren Bereich der Tracks geprüft, ob die Symbolik gleich ist. Sobald das erkannt wird, kommt eine Meldung das es möglich ist die Symbole zu den bereits existierenden Tracks hinzuzufügen.



Wenn diese Abfrage mit Yes beantwortet wird, werden die hinzugefügten Symbole als Datenelement an die passende existierende Track angefügt. Wird die Abfrage mit No beantwortet werden neue Tracks hinzugefügt.



Das gleiche Ergebnis kann auch erreicht werden, wenn jedes Symbol einzeln auf die bereits existierende Track gezogen wird.

Reihenfolge verändern

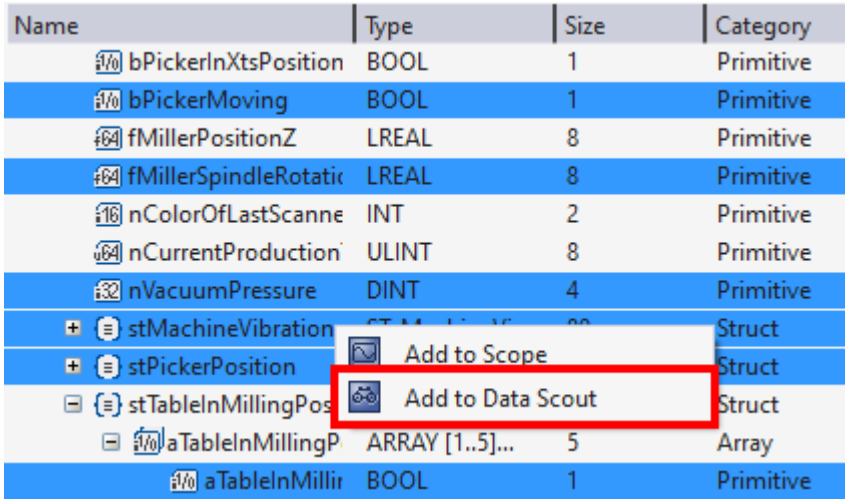
Wenn die Reihenfolge von existierenden Tracks oder Datenelementen verändert werden soll, kann dies mit

Drag and Drop geschehen. Datenelemente können auf der gesamten Fläche und Tracks am linken Rand verschoben werden. Dazu muss der Bereich mit der linken Maustaste angeklickt werden und mit gedrückter Maustaste verschoben werden.

Datenelemente können in der gleichen Track neu angeordnet werden oder in eine andere Track verschoben werden. Wenn Datenelemente auf Container Tracks oder in den Track Editor gezogen werden, wird eine neue Track angelegt. Außerdem muss das Datenelement mit der Track kompatibel sein, in das es verschoben wird z.B. ein Datenelement mit mehreren Symbolen aus einer Multi Track kann nicht in eine Single Track verschoben werden.

Tracks können unterhalb des Containers, in dem sie liegen neu angeordnet oder in einen anderen Container verschoben werden. Es ist nicht möglich Tracks in Container zu verschieben die innerhalb der zu verschieben Track liegen oder Tracks zu verschieben die durch ihre Konfiguration oder die Konfiguration des darüberliegenden Containers nicht verändert werden können z.B. Struktur Arrays.

Hinzufügen per Target Browser Command

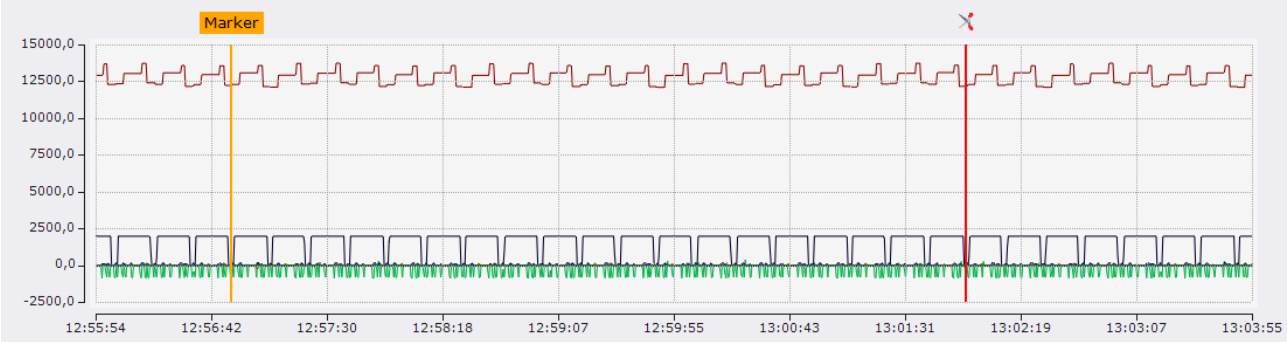


Es gibt die Möglichkeit Symbole aus dem Target Browser zu einem Track Editor hinzuzufügen, indem das „Add to Data Scout“ Kommando im Kontext Menü aufgerufen wird. Das Kommando wird angezeigt, wenn im Target Browser Symbole, welche mit dem Track Editor kompatibel sind, ausgewählt und mit Rechtsklick angeklickt werden. Für jedes Symbol und die Subsymbole von Arrays und Strukturen werden dann Tracks angelegt.

Wenn kein Data Scout Project vorhanden ist, wird versucht ein neues anzulegen. Wenn schon ein Data Scout Project in der derzeitigen Solution liegt, werden die Symbole dort hinzugefügt. Ist ein Track Editor im Solution Explorer ausgewählt, werden die Symbole in diesen eingefügt, ansonsten wird eine neuer angelegt.

6.2.1.2 Chart

Der Data Track Editor nutzt zur Visualisierung der Daten der Tracks ein Chart ähnlich dem YT Chart des Scope Views.



Das Chart besitzt eine Y-Achse auf der linken Seite und unten eine X-Achse, auf der die Zeiten dargestellt werden. Auf diesen Achsen wird jede sichtbare Track die Daten enthält als Graph dargestellt.

Die Zeitwerte werden für das Chart neu vergeben und es werden nicht die Zeitwerte aus den Originalquellen verwendet, um mehrere Quelle in einem Chart darzustellen. Der Startzeitpunkt, sowie die Zykluszeit können in den Track Editor Timeline Optionen eingestellt werden.

Im Track Editor über der Timeline und den Datenelementen der Tracks wird ein blauer Bereich angezeigt. Dieser Bereich zeigt an, welcher Bereich im Chart angezeigt wird. Wenn im Chart über die Funktionen der Toolbar oder mit dem Mausrad gezoomt wird, dann ändert sich auch der selektierte Bereich in den Tracks. Andersherum funktioniert es gleich, also kann in den Tracks der selektierte Bereich angepasst werden und der im Chart angezeigte Bereich ändert sich.

Laden von Daten

Sobald eine Track mit Daten im Data Track Editor liegt werden diese Daten geladen. Das Laden geschieht automatisch im Hintergrund. Beim Laden werden die Tracks, deren Zusammensetzung von Datenelementen gleich ist, also zur gleichen Time Track gehören, gleichzeitig geladen. Jedes Mal, wenn sich der selektierte Bereich Ändert oder wenn eine neue Track eingefügt wird, werden die Daten für den selektierten Bereich aktualisiert.

Wenn bei einer hohen Display Breite im Chart zu viele Daten angezeigt werden würden, werden die Daten nur teilweise angezeigt.

Analytics Files laden

Für die schnelle Darstellung von Analytics Files müssen diese vorbereitet sein. Im Target Browser in der Analytics File Extension wird bei Analytics Files die nicht vorbereitet sind ein grüner Pfeil am Ordner angezeigt. Wird auf diesen Pfeil geklickt öffnet sich ein neues Fenster und das Analytics File wird für die schnelle Darstellung vorbereitet.

Marker

Im Chart können Marker angezeigt werden. Diese bleiben an einer gewissen Position stehen und können durch das Anklicken und Verziehen auf der X-Achse nachträglich verschoben werden.

Es gibt zwei Arten von Markern. Der Standard-Marker hat einen Text über der Marker Linie. Der Text ist mit der gleichen Farbe hinterlegt wie die Marker Linie. Die zweite Art von Marker ist zum Ausschneiden von Daten gedacht. Von diesen Markern kann es in einem Data Track Editor nur zwei geben, wird ein dritter eingefügt wird der Ausscheid-Marker, der dem neuen am nächsten ist, entfernt. Ausschneid-Marker haben keinen Text über ihrer Linie, sondern eine Schere und die Linie ist immer rot.

Wenn ein Marker mit der linken Maustaste angeklickt wird, öffnet sich ein Tooltip welcher die Zeit und die Werte an der Position des Markers anzeigt. Wird ein Marker mit der rechten Maustaste angeklickt, wird ein Kontext Menü geöffnet. In diesem gibt es die Option den Marker aus dem Data Track Editor zu entfernen.

6.2.1.3 Toolbar

Die Toolbar des Data Track Editors bietet verschiedene Funktionen, um die Anzeige im Chart zu manipulieren und die Daten in den Tracks zu bearbeiten.

Diese Funktionen dienen dazu die X und oder Y-Achse zurückzusetzen.

	Setzt die Display Breite des Charts auf die maximal mögliche Breite und setzt die Skalierung der Y-Achse zurück.
	Setzt die Skalierung der Y-Achse zurück.

Von diesen Funktionen ist immer nur eine aktiv. Sie beeinflussen, wie sich ein Linksklick in das Chart verhält.

	Wenn diese Funktion aktiviert ist, kann in das Chart geklickt werden, um einen Bereich auf der X-Achse zu selektieren. Dazu muss ins Chart geklickt, die Maustaste gedrückt gehalten werden und dann muss die Maus nach links oder rechts bewegt werden, was zu einer Vorschau der Selektierung führt. Beim Loslassen der Maustaste wird die Display Breite auf diesen Bereich verkleinert.
	Wenn diese Funktion aktiviert ist, kann in das Chart geklickt werden, um einen Bereich auf der Y-Achse zu selektieren. Beim Drücken der Maustaste wird eine Vorschau angezeigt welcher Bereich selektiert wird. Beim Loslassen der Maustaste wird die Skalierung auf der Y-Achse auf diesen Bereich verkleinert.
	Wenn diese Funktion aktiviert ist, kann in das Chart geklickt werden, um den angezeigten Bereich auf der X- und Y-Achse zu verkleinern. Beim Drücken der Maustaste ins Chart wird eine Vorschau angezeigt welcher Bereich angezeigt werden wird. Beim Loslassen der Maustaste wird die Skalierung der Y-Achse und die Display Breite auf diesen Bereich verkleinert.
	Wenn diese Funktion aktiviert ist, kann in das Chart geklickt werden, um einen neuen Marker einzufügen. Der Marker wird auch in den Tracks angezeigt und kann dort umbenannt werden.
	Wenn diese Funktion aktiviert ist, kann in das Chart geklickt werden, um einen speziellen Marker einzufügen, mit dem ein Bereich ausgeschnitten werden kann.

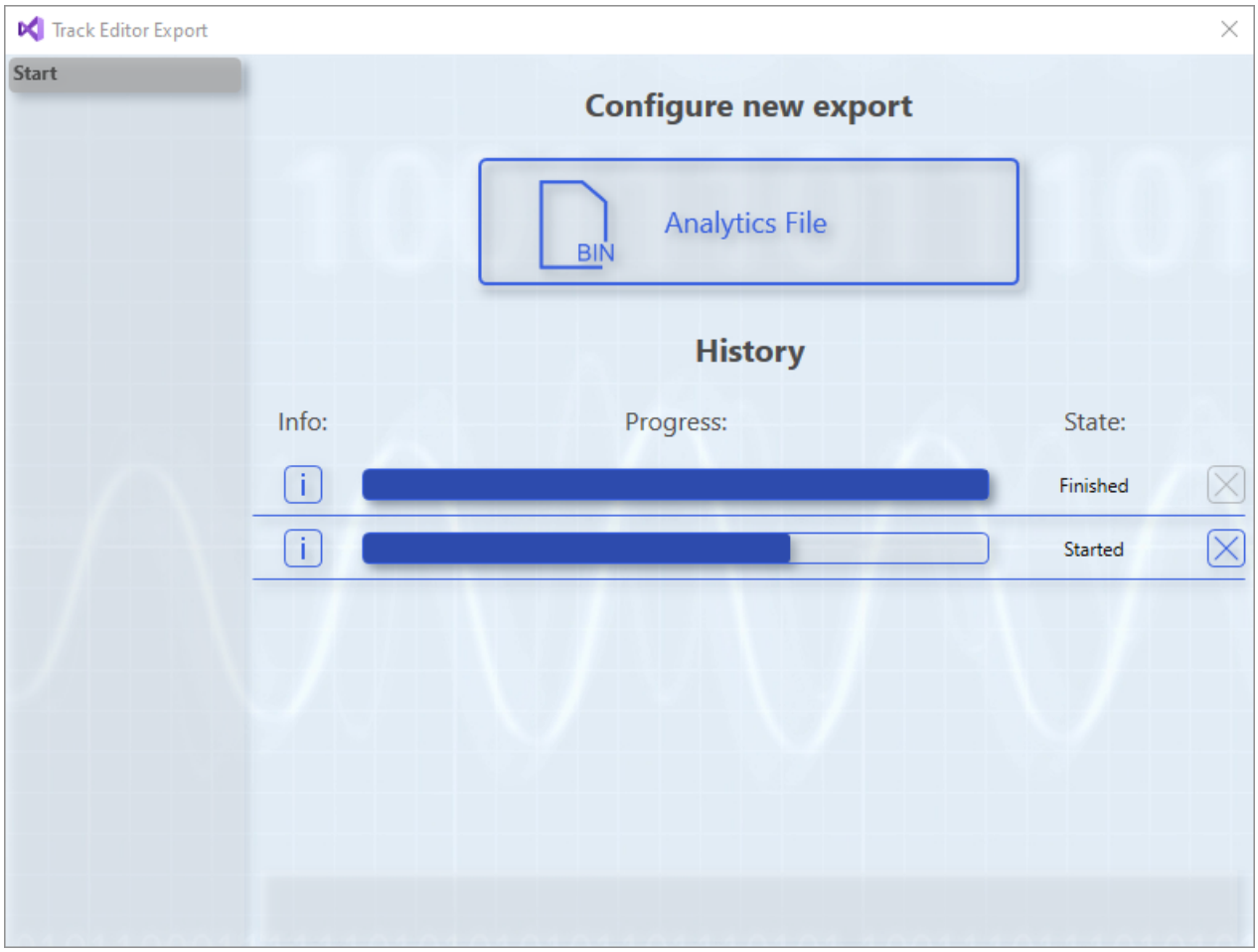
Diese Funktionen können dazu genutzt werden, um die Daten der Tracks zu bearbeiten.

	Legt fest ob nur sichtbare Tracks oder auch unsichtbare Tracks von den folgenden Funktionen, die Daten entfernen können, bearbeitet werden.
	Diese Funktion kann nur verwendet werden, wenn zwei Marker zum Ausschneiden eingefügt worden sind. Diese können mit oder per Rechtsklick auf einen Graphen eingefügt werden. Beim Ausführen wird der Bereich zwischen den zwei Markern entfernt.
	Mit dieser Funktion kann der in den Tracks selektierte Bereich entfernt werden. Der selektierte Bereich wird dann verkleinert, da der zuvor im Chart angezeigte Bereich entfernt worden ist.
	Mit dieser Funktion wird alles außer dem selektierten Bereich entfernt.
	Diese Funktion kann nur verwendet werden, wenn nicht alle Tracks dieselbe Menge an Daten haben. Diese Differenz wird auch rot gestrichelt in der Timeline der Tracks angezeigt. Beim Ausführen werden, bei den Tracks die mehr Daten enthalten, am Ende so viele Daten entfernt, dass die Anzahl mit der Track übereinstimmt, die am wenigsten Daten enthält.
	Hier kann der Curve Creator, in einem neuen Fenster, geöffnet werden. Dieser enthält die Daten des selektierten Bereichs, von den sichtbaren Tracks. Nach dem Schließen des Curve Creators werden die Graphen in einem Analytics File gespeichert und in den Tracks ersetzt. Sollte die Funktion zum Verändern des Start- oder Endpunkts im Curve Creator verwendet worden sein, wird der Bereich ersetzt, aus dem die Daten stammen, ansonsten der vorher selektierte Bereich. Diese Funktion ist nur verfügbar, wenn der selektierte Bereich kleiner als 100.000 Werte ist.

6.2.1.4 Export

Ein Data Track Editor kann mithilfe des Track Editor Export Wizards exportiert werden. Um den Wizard aufzurufen, muss dieser Button , der neben der Timeline der Tracks ist, geklickt werden.

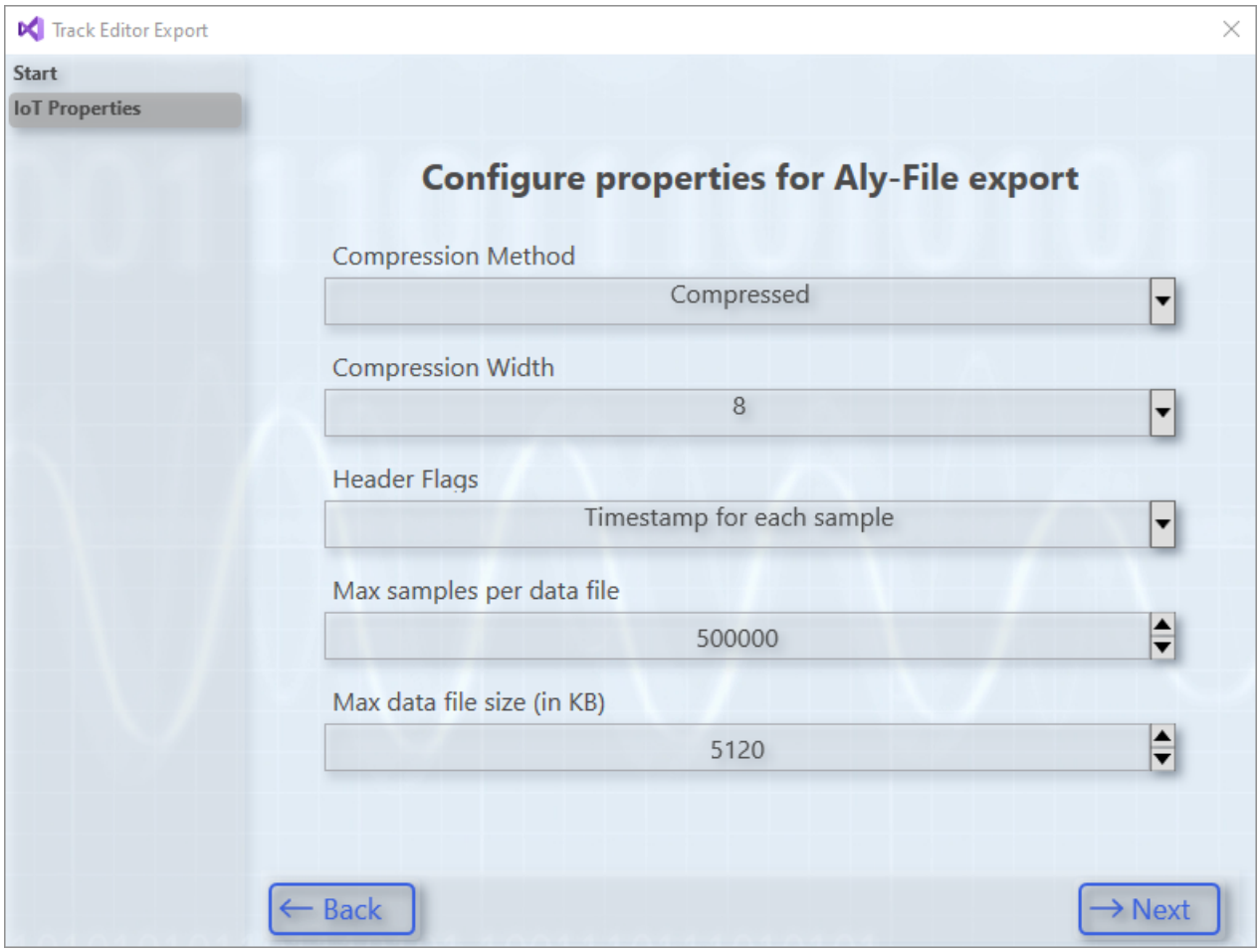
Startseite



Auf der Startseite des Wizards können im oberen Teil neue Exports gestartet werden. Alle verfügbaren Export Formate sind als eigener Button dargestellt. Im unteren Teil ist eine Historie von Exports, die gestartet worden sind, während das Projekt geöffnet war. Jeder Eintrag in der Historie hat auf der linken Seite einen Button mit Infos zum Export. In der Info steht die Startzeit des Exports, wichtige vom Format abhängige Eigenschaften und wenn Fehler auftreten sollten steht hier auch die Fehlermeldung. Rechts daneben ist eine Fortschrittsanzeige, welche anzeige wie viele der zu exportierenden Samples schon exportiert worden sind. Danach wird der Status des Exports ausgegeben. Der Button ganz rechts kann dazu genutzt werden laufende Exports abzubrechen.

Wenn auf einen der Buttons geklickt wird, mit denen ein neuer Export gestartet werden kann, wird eine oder mehr neue Seite im Wizard eingeblendet in der die Einstellungen für das Format vorgenommen werden können.

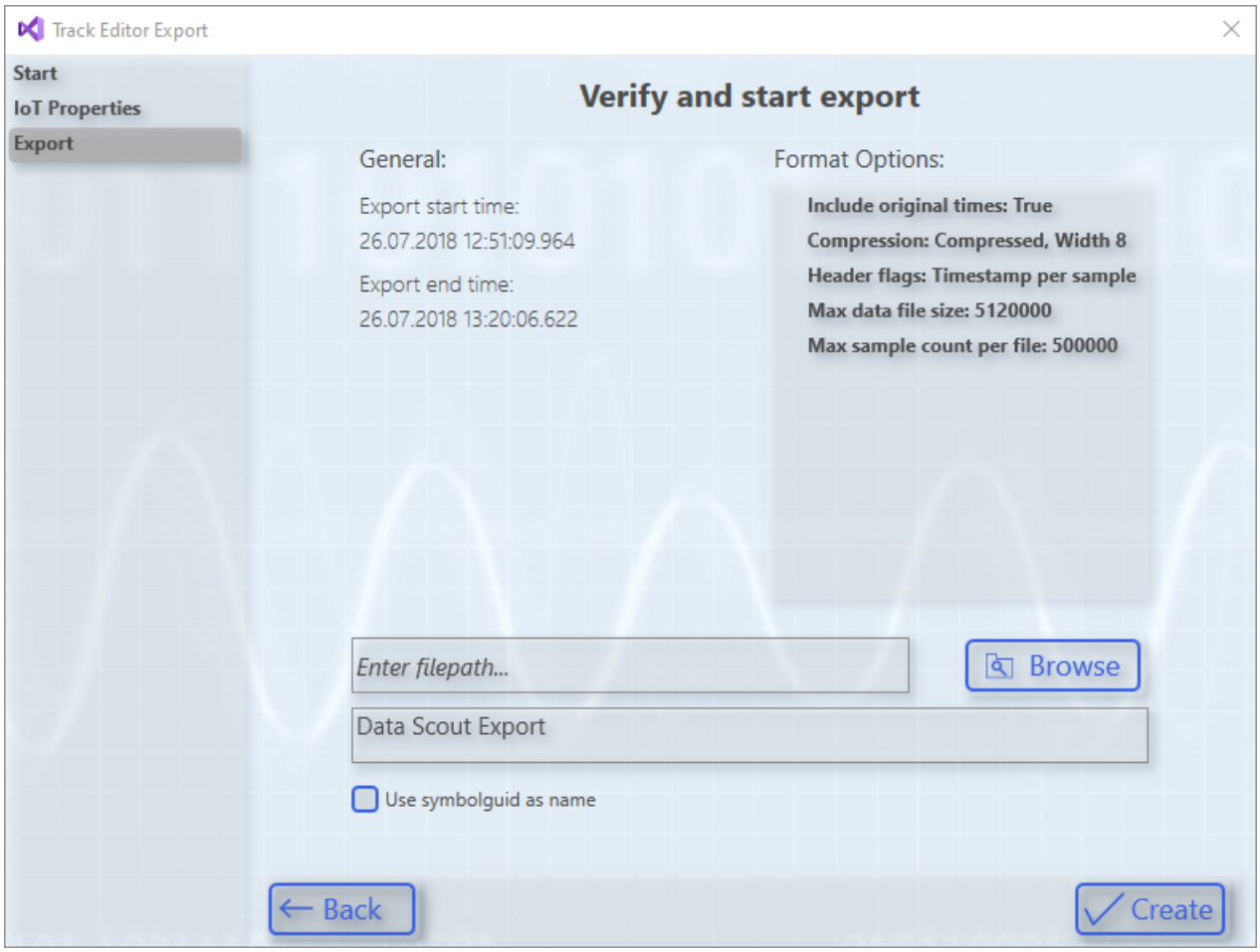
Analytics File Konfigurationsseite



Diese Konfigurationsseite für das Analytics File Format gibt folgende Einstellungen:

Kompressionsmethode	Analytics File können komprimiert oder nicht komprimiert abgespeichert werden. (Datenkompression)
Kompressionsweite	Diese Option ist nur verfügbar, wenn das Analytics File komprimiert abgespeichert werden soll. Sie gibt die maximale Weite zum Vergleichen für die Kompression an.
Header Flags	In den Flags kann angeben werden, ob jede .tay Datei nur einen Zeitwert oder ob jedes Sample in der .tay Datei einen eigenen Zeitwert bekommen soll.
Maximale Samples und Dateigröße	Mit diesen zwei Eigenschaften kann eingestellt werden, wie groß die .tay Dateien des Analytics Files werden können. Zum einen kann eingestellt werden wie viele Samples maximal in eine Datei gespeichert werden sollen und zum anderen kann eingestellt werden wie groß eine Datei in Kilobyte im Dateisystem werden soll. Wenn einer dieser Werte überschritten wird, wird eine neue .tay Datei erstellt.

Export Seite



Die Export Seite wird angezeigt, nachdem die Konfigurationsseiten für das Format durchlaufen wurden. Hier werden noch einmal alle Einstellungen aufgeführt und es kann, wenn für das Format nötig, ein Speicherort angegeben werden. Das obere Textfeld gibt dabei den Ordner an und das untere den Dateinamen an.

Für das Analytics File Format kann auch eine Checkbox angehakt werden anstelle eines Dateinamens anzugeben, dann wird als Ordnername die ID der Symbolik genutzt.

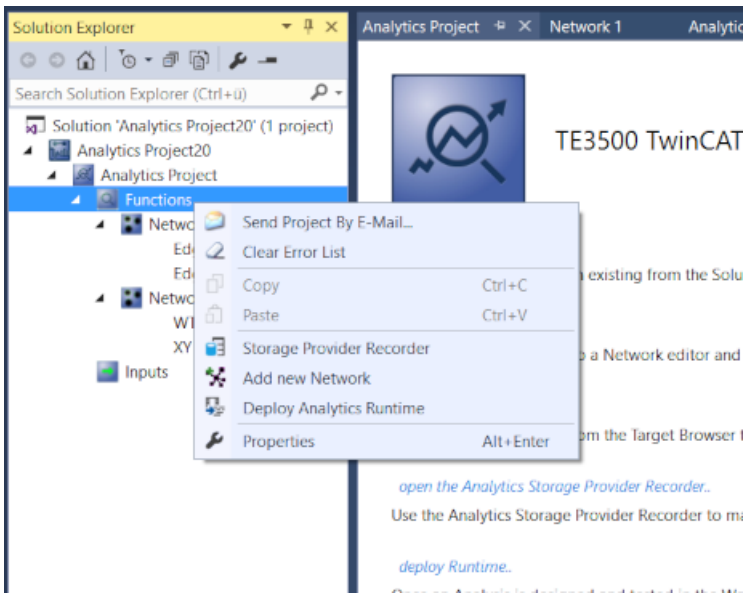
Wenn auf den Create Button geklickt wird, startet der Export im Hintergrund. Dann schließt sich diese Seite und der Wizard springt zurück auf die Startseite. Hier steht dann auch der neue Export in der Historie und der Fortschritt kann verfolgt werden.

6.3 Editor Basics

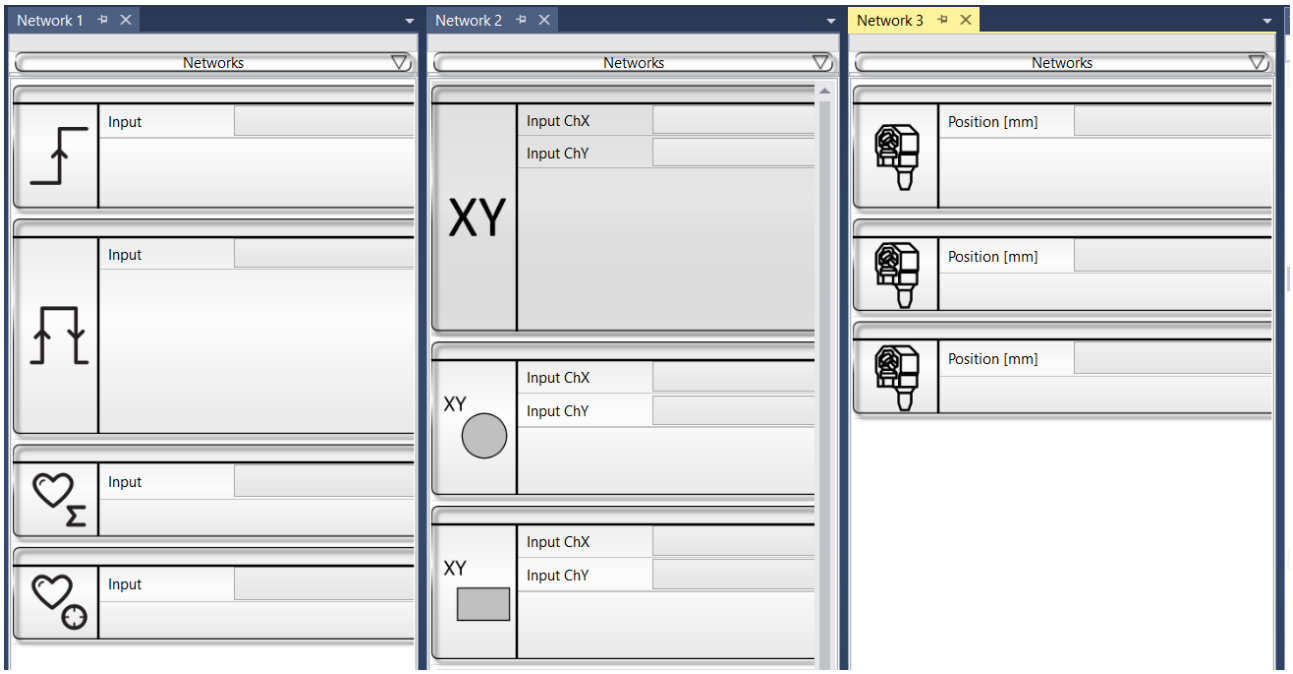
6.3.1 Netzwerke

Die Netzwerke eignen sich zur Organisation und Strukturierung einer Analyse. Dadurch wird die Übersichtlichkeit deutlich gesteigert. Darüber hinaus können sie auch als Container für Algorithmen dienen, welche Sie als Template abspeichern können.

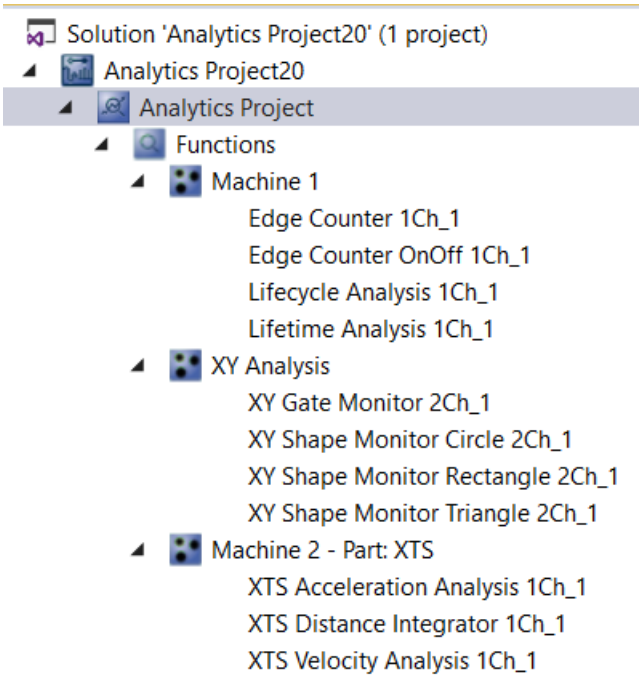
Ein Netzwerk kann direkt von der Analytics Project-Startseite oder über das Kontextmenü hinzugefügt werden.



Jedes Netzwerk wird in einem einzelnen Karteireiter dargestellt. Auf diese Weise können die Netzwerke separat, d. h. nebeneinander oder übereinander angeordnet, angezeigt werden.



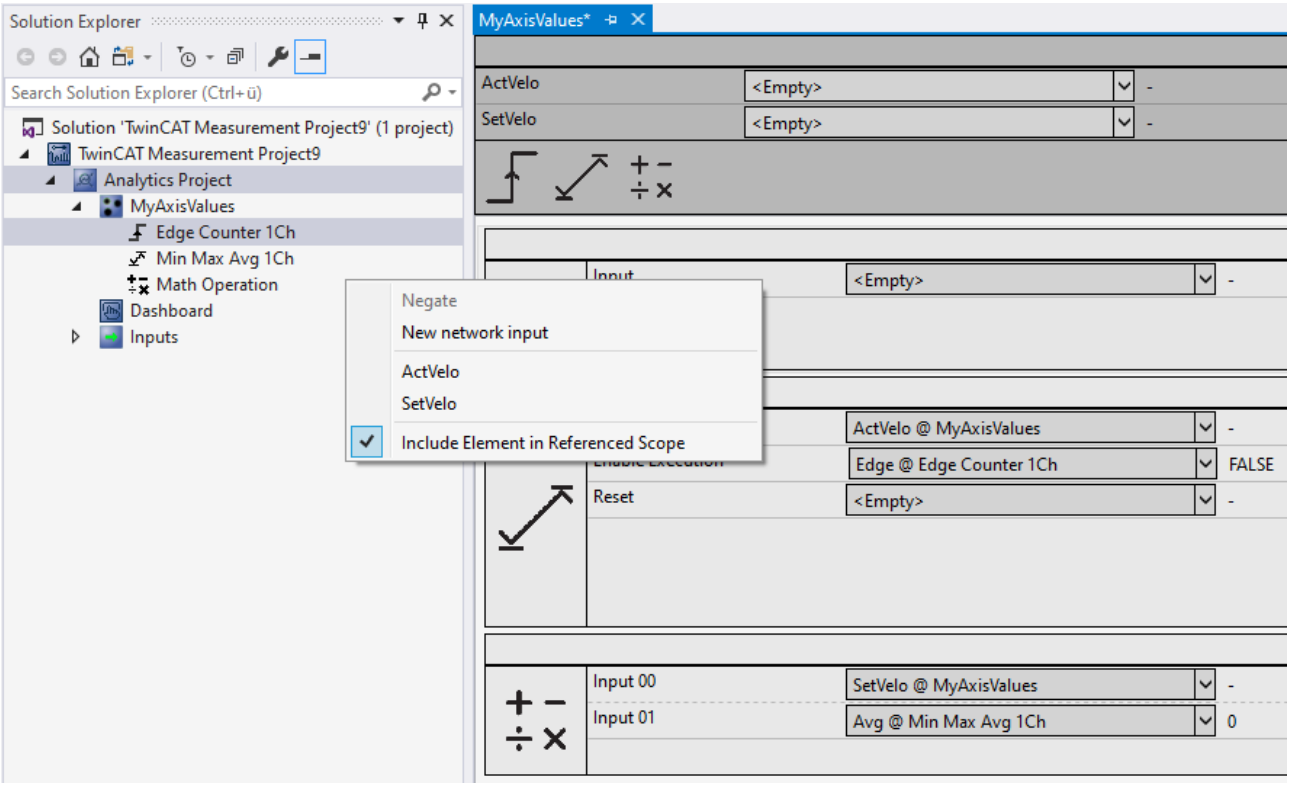
Des Weiteren haben Sie die Möglichkeit, die Netzwerke umzubenennen (F2 auf ausgewähltem Netzwerkelement im Projektmappen-Explorer), zum Beispiel, um Netzwerke für verschiedene Maschinen, Maschinenteile oder andere inhaltliche Verbindungen zu erstellen.



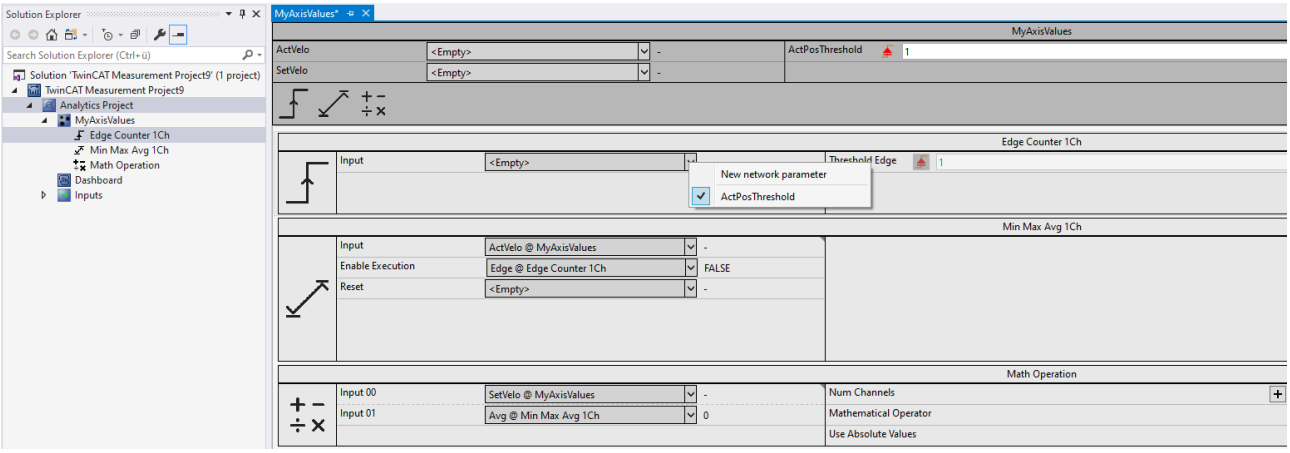
6.3.1.1 Netzwerke als Template

Sie haben die Möglichkeit, Eingänge, Parameter oder auch Ausgänge von Algorithmen innerhalb von einem Netzwerk direkt nach außen an das Netzwerk zu heften. So hat das Netzwerk selbst Eingänge, Parameter und Ausgänge. Dadurch ist es möglich, immer wiederkehrende Analysen als Template abzuspeichern und mehrfach zu instanziieren.

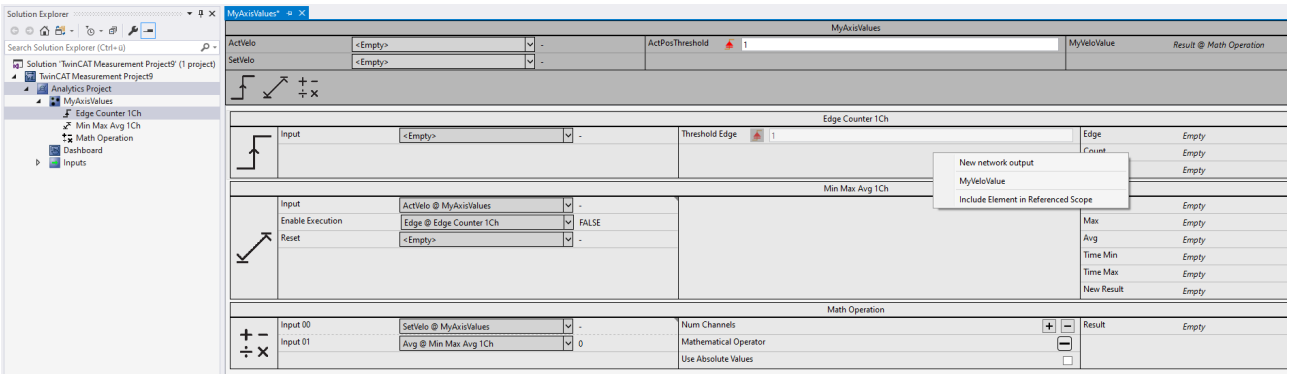
Wenn Sie eine Analyse erstellt haben, müssen Sie zunächst auswählen, welche der verfügbaren Eingänge nach außen am Netzwerk sichtbar sein sollen. Wählen Sie dazu den gewünschten Eingang aus und wechseln per Rechtsklick in das Kontextmenu. Dort können Sie den Eingang mit einem bereits vorhandenen Netzwerkeingang verknüpfen oder alternativ einen neuen Netzwerkeingang definieren.



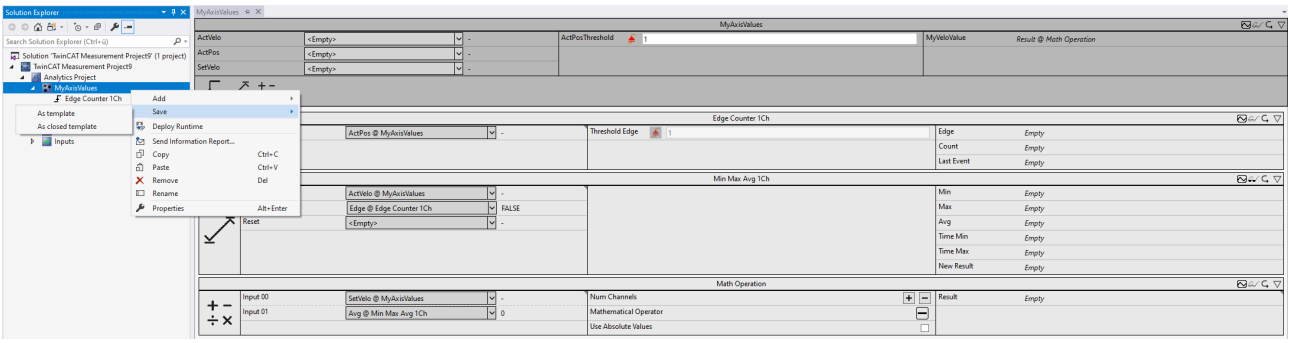
Die gleiche Vorgehensweise gibt es für Parameter und für die Ausgänge.



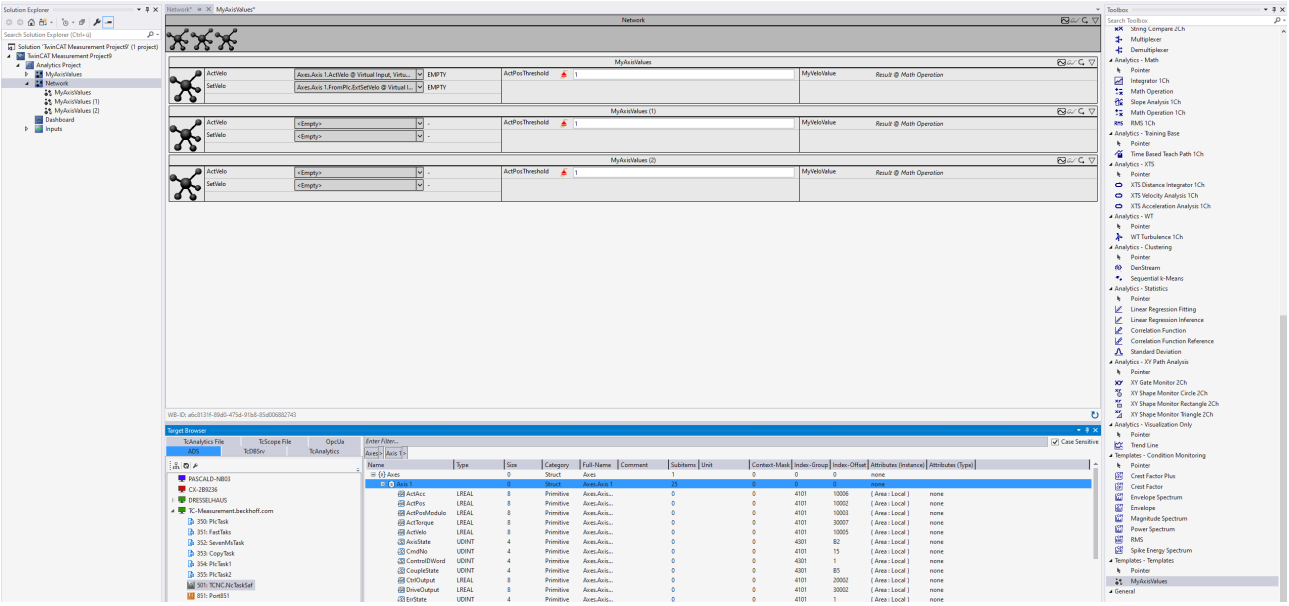
Neben diesen Ein- und Ausgängen gibt es auch die Möglichkeit, sogenannte dynamische Ein- und Ausgänge zu spezifizieren. Diese werden Ihnen automatisch angeboten, sobald einer der gewählten Algorithmen diese Funktion unterstützt. Hintergrund ist dabei die Möglichkeit, die Anzahl der Eingänge oder Ausgänge an einem Algorithmus über einen Parameter zu erhöhen. Beispielsweise bei den Eingängen des Math Operation Algorithmus.



Ist die Definition des Netzwerks, welches als Template dienen soll, abgeschlossen, können Sie es entsprechend abspeichern. Dafür gehen Sie in den Solution Explorer und führen einen Rechtsklick auf dem Netzwerk aus. Im Kontextmenü gibt es die Option Save. Dabei können Sie zwischen Template und Closed Template wählen. In einem einfachen Template können Sie nach der Instanziierung hineinschauen und die Verschaltung der Basisalgorithmen sehen und auch verändern. Bei einem geschlossenen Netzwerk entfällt diese Möglichkeit. Dies ist allerdings kein Knowhow-Schutz! Auch in einer möglichen SPS-Code-Generierung durch die Analytics Workbench ist die interne Logik sichtbar.



Nach dem Abspeichern ist das Template in der Toolbox auswählbar und kann für wiederkehrende Analysen verwendet werden.



Eine besondere Funktion bietet der Target Browser an.
Sind für das Netzwerk die Namen so gewählt worden, dass diese exakt mit Variablen einer Struktur übereinstimmen, können Sie eine vollständige Struktur wie z. B. eine Achsen Struktur direkt auf einen der x-Eingänge eines Netzwerks ziehen. Alle übereinstimmenden Namen von sämtlichen Netzwerkeingängen und Strukturvariablen werden automatisch gemappt.

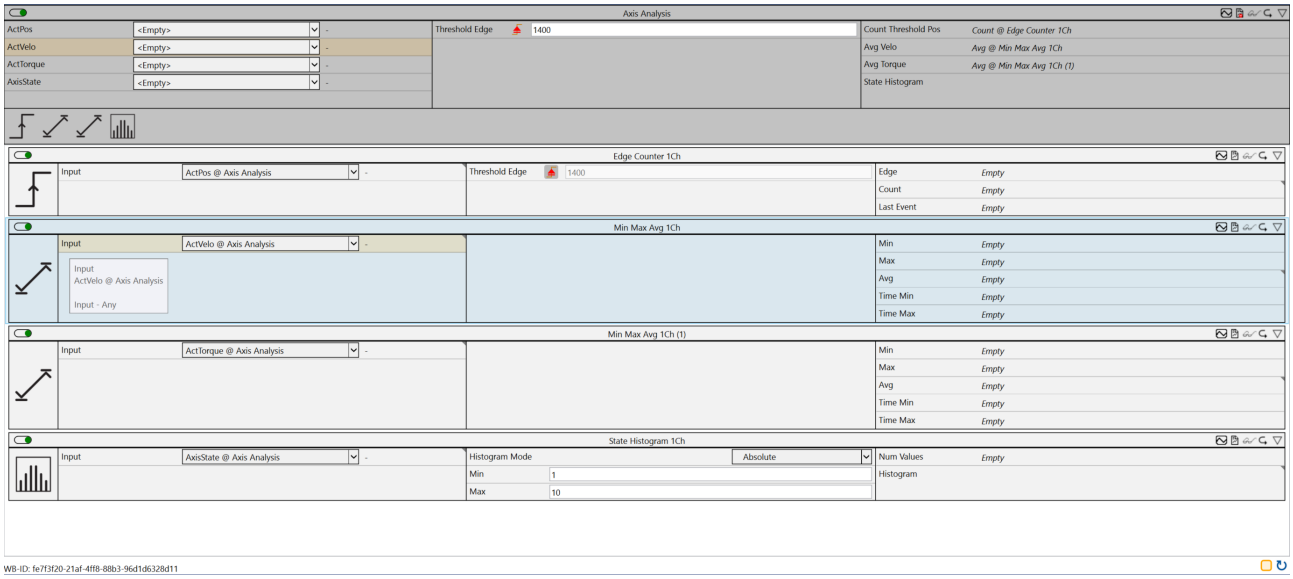
Scope-Konfiguration gespeichert im Netzwerktemplate

Eine erstellte Scope-Konfiguration kann zusammen mit dem dazugehörigen Netzwerk in einem Netzwerktemplate abgespeichert werden [▶ 360], um bei einer erneuten Verwendung eines Netzwerks automatisch die gleiche Scope-Konfiguration zu erhalten.

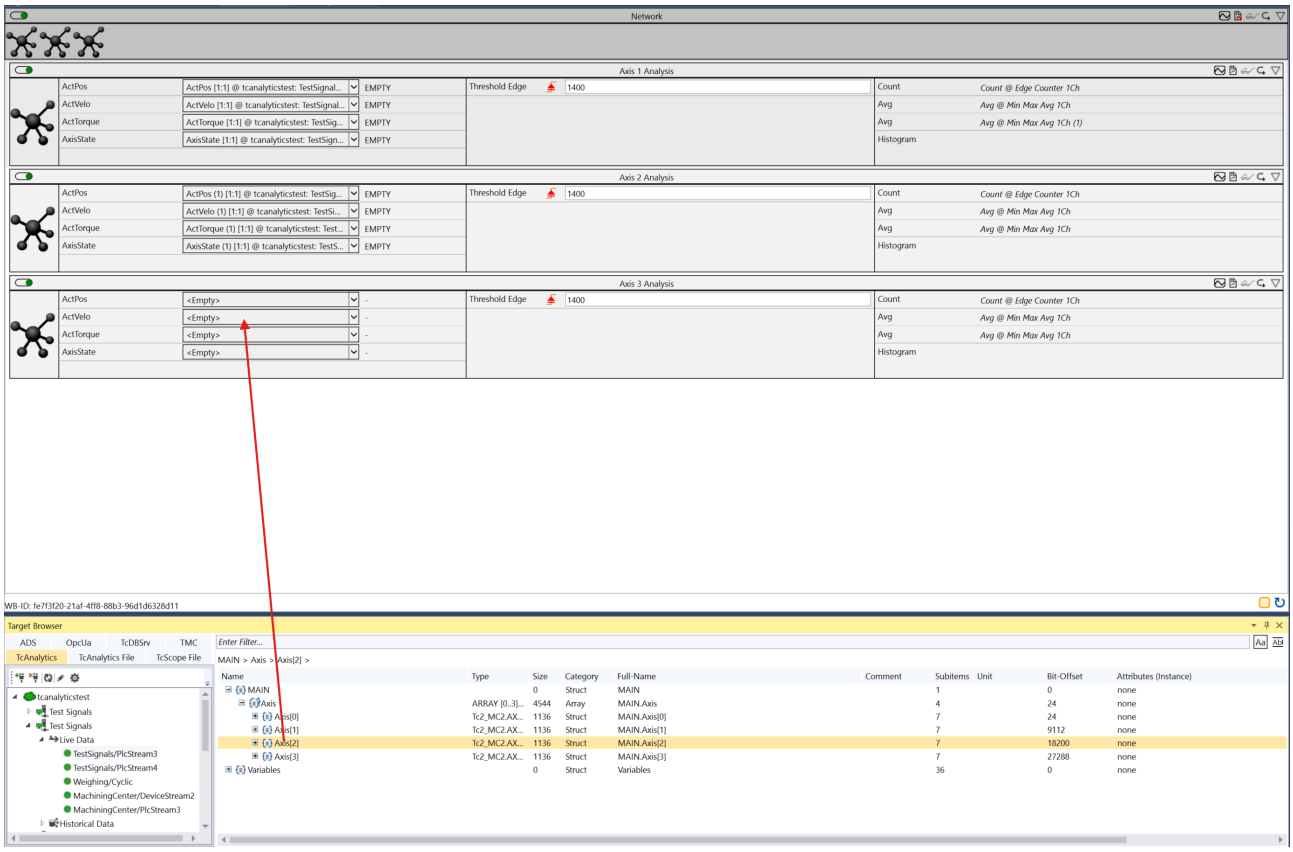
6.3.2 Smart Link

Die Smart-Link-Funktionen in TwinCAT Analytics erleichtern das Engineering. Mit einigen Tasten-/Maus-Kombinationen können sehr schnell gruppenweise Module mit Variablen und anderen Werten verknüpft werden.

Eine erste Variante der Smart-Link-Funktion ist die Verknüpfung von Struktur-Member aus dem Target Browser heraus. Das ist insbesondere nützlich, wenn Sie Analytics-Netzwerk-Templates verwenden. Voraussetzung dafür ist, dass Sie Ihre Netzwerk-Inputs exakt so benennen wie die Member, die Sie aus einer Struktur verknüpfen möchten.

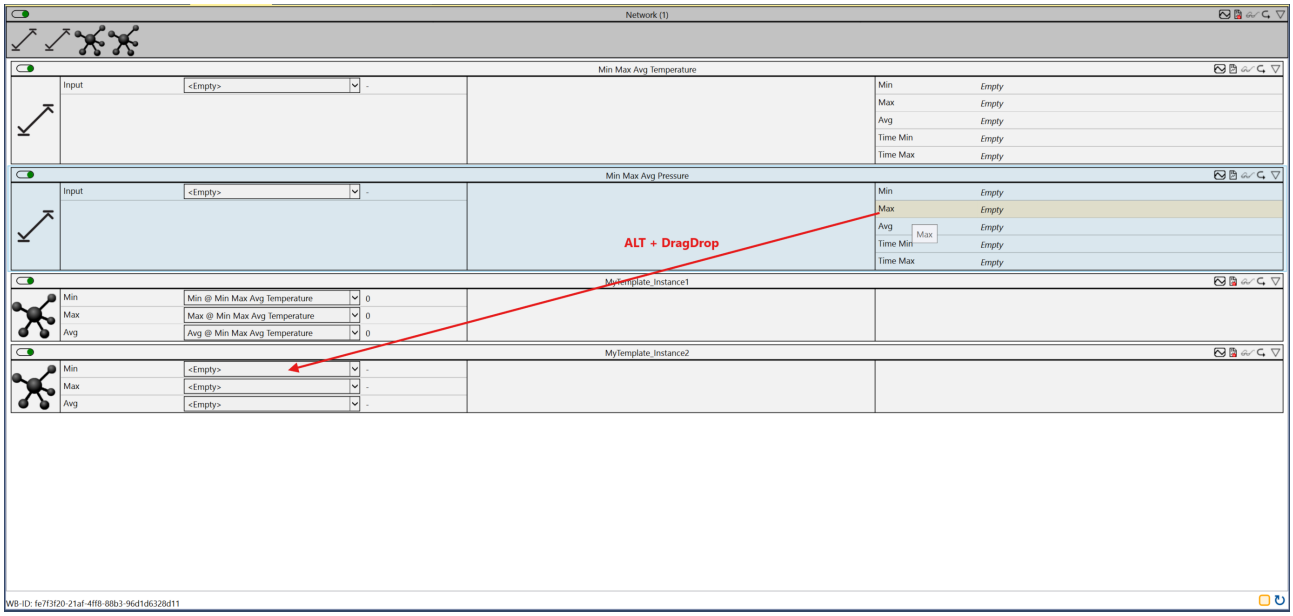


Im gezeigten Bild sind das die Namen der bekannten NC-Struktur NcToPlc. Die Eingänge bekommen die exakt gleichen Namen wie die Member in der Struktur.



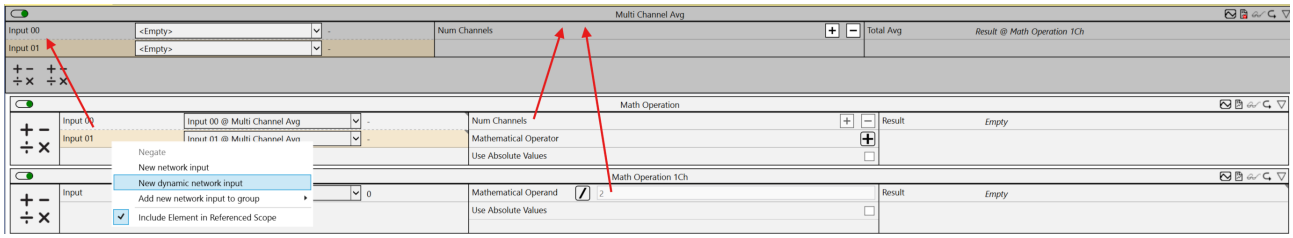
Im nachfolgenden Schritt können somit die Strukturen der Antriebsachsen sehr einfach verknüpft werden. Es reicht die gewünschte Struktur auf einen der Eingänge des Netzwerk Templates per Drag-and-Drop zu verknüpfen. Alle Member werden automatisch mit den Eingängen verknüpft.

Eine weitere Smart-Link-Funktion erlaubt das einfache Verknüpfen von Ausgängen mit Eingängen unter Berücksichtigung der Namen. Bauen Sie beispielsweise ein Netzwerk-Template und benennen die Eingänge so wie die Ausgänge der zu verknüpfende Module.

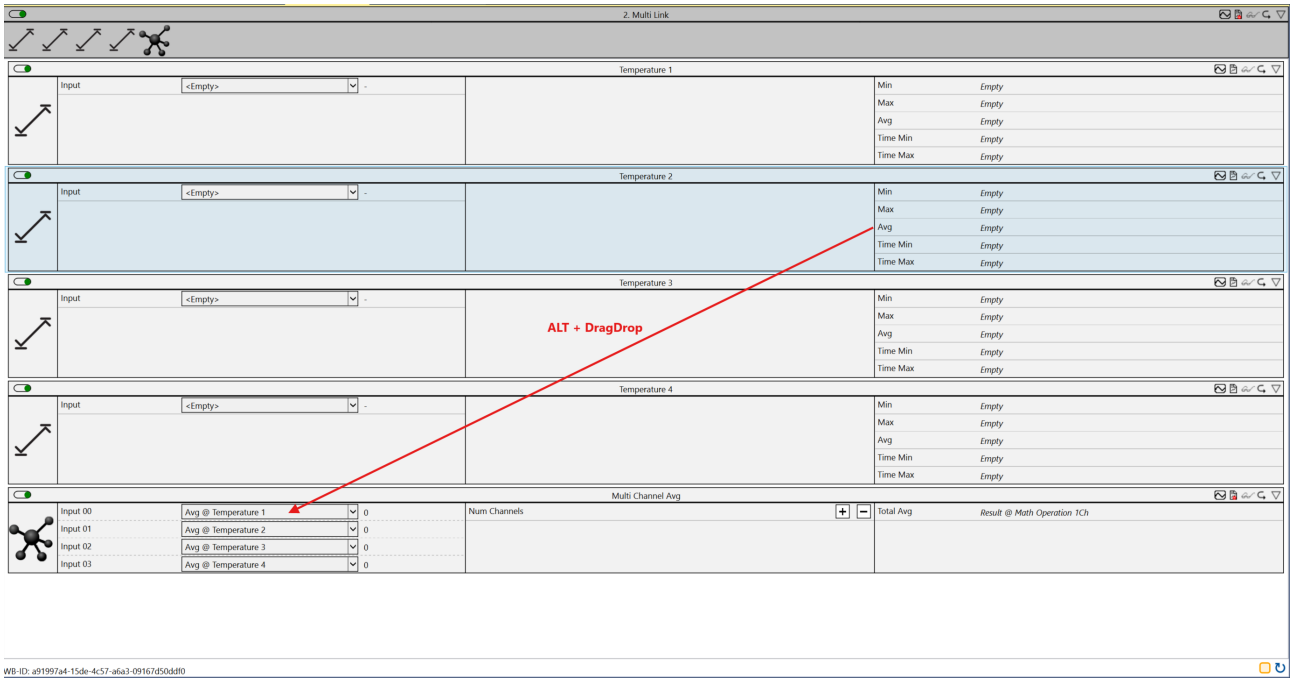


So können später durch die Kombination von gedrückter ALT-Taste und dem Drag-and-Drop eines der Ausgänge auf einen der Eingänge des Templates alle weiteren Ausgänge automatisch mit den zugehörigen Eingängen verknüpft werden.

Die nächste Smart-Link-Funktion ist kombiniert mit einer Suchfunktion in den Modulen des aktuellen Netzwerks. Sie erstellt beispielsweise ein Netzwerk-Template mit dynamischen Eingängen, wodurch sich bei der späteren Verwendung über die Parameter die Anzahl der Eingänge einstellen lässt. Auch der Num-Channel-Parameter, der exemplarisch im Screenshot zu sehen ist, wird als Netzwerkparameter verknüpft. Genauso der Operand, der mit der Anzahl der Kanäle wächst.

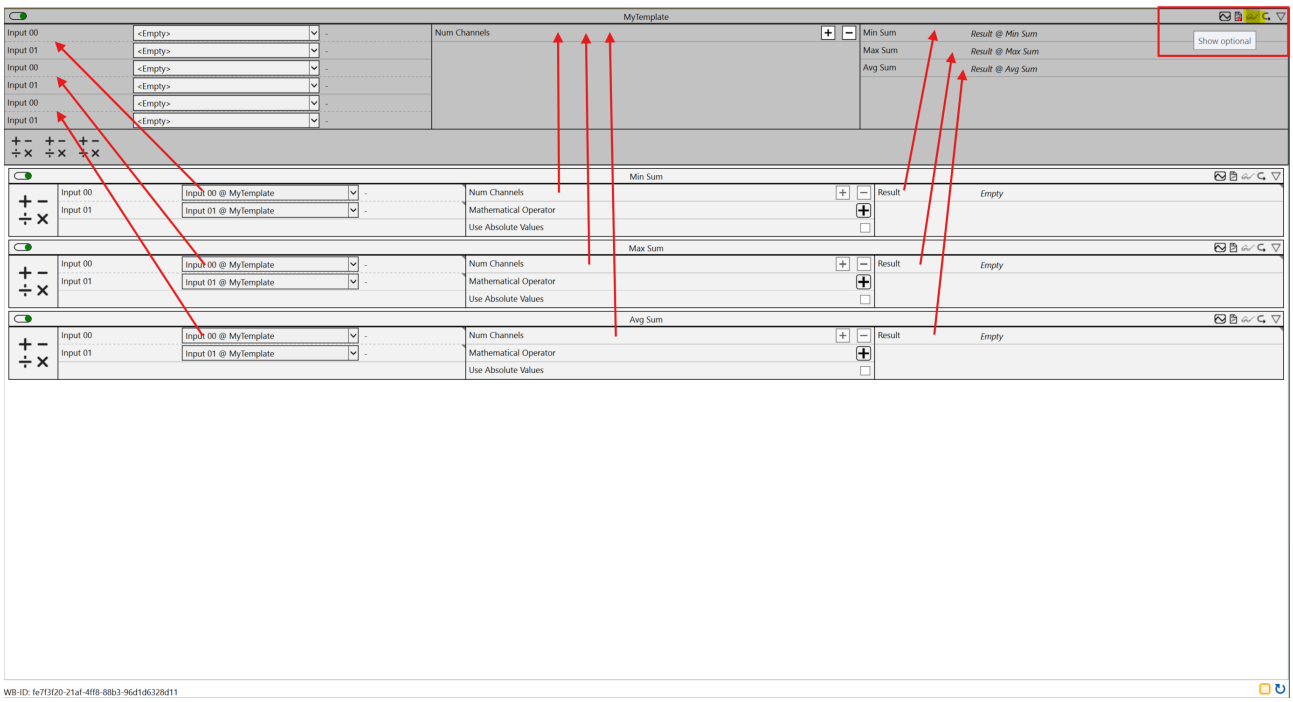


Das Beispiel soll dabei einfach X-Eingänge aufsummieren und den Durchschnitt berechnen.

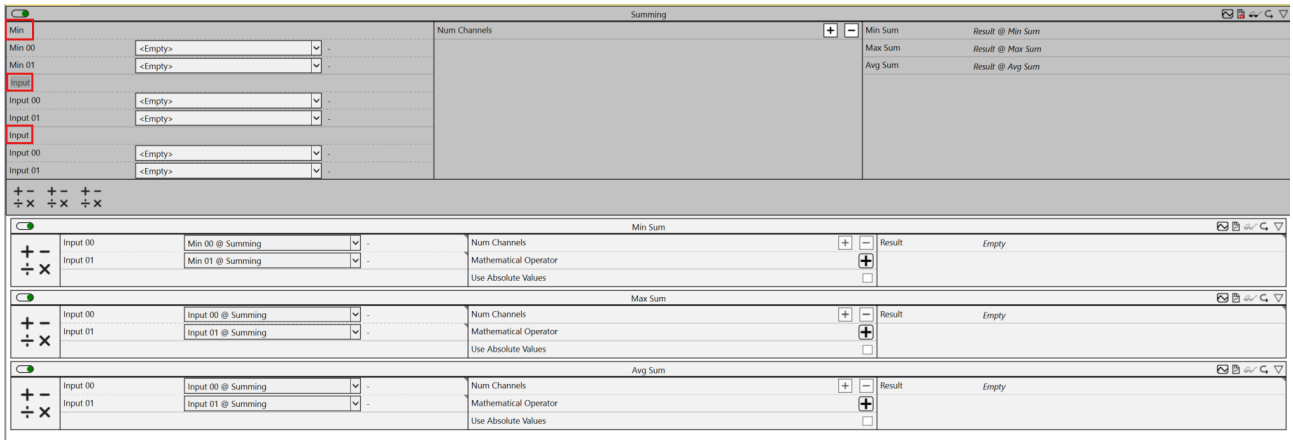


Bei der Nutzung wird am Template die Anzahl der Eingänge eingegeben. Anschließend nehmen Sie einen Ausgang, der entsprechende Namens-Äquivalente im lokalen Netzwerk hat und ziehen ihn bei gedrückter ALT-Taste auf einen der Template-Eingänge. Dadurch wird im lokalen Netzwerk nach Ausgängen mit gleichem Namen gesucht und alle automatisch mit den Eingängen des Templates verknüpft.

Die gezeigte Smart-Link-Funktion lässt sich auch auf eine gruppenweise Verknüpfung ausdehnen. Dafür definieren Sie in Ihrem Template zunächst wieder dynamische Eingänge, Parameter und entsprechende Ausgänge. Im Beispiel sollen alle Min-Werte, alle Max-Werte und alle Durchschnitts-Werte eines Netzwerks aufsummiert werden.

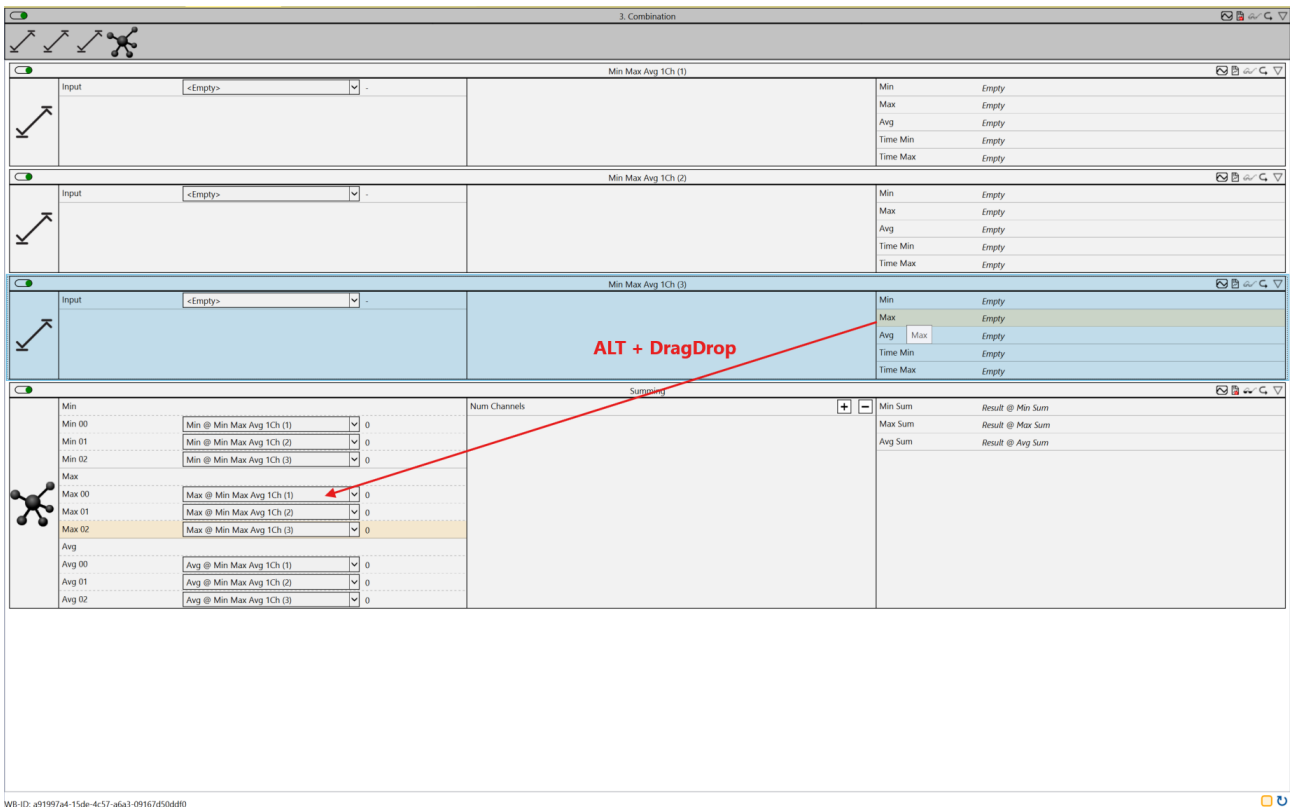


Damit die gruppenweise Verknüpfung funktioniert, klicken Sie auf Netzwerk-Ebene auf die Brille rechts oben, um weitere Details sichtbar zu machen.



Dadurch sehen Sie bei den Netzwerk Eingängen nun Gruppen Namen. Diese vergeben Sie so, wie die späteren Ausgänge exakt heißen. In unserem Beispiel bilden Sie die Gruppen Min, Max und Avg.

Bei der Nutzung des Templates ist es wieder eine einzige Operation mit gedrückter ALT-Taste und einem Drag-and-Drop von einem der Ausgänge auf einen Eingang des Templates. Dadurch werden alle Algorithmen des Netzwerks durchsucht und alle Ausgänge mit gleichem Namen wie die Gruppennamen automatisch verknüpft.



6.3.3 Enable Disable

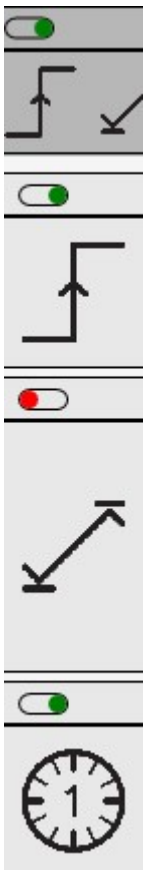
Die TwinCAT Analytics Workbench bietet die Funktionalität einzelne Algorithmen oder auch komplexe Netzwerke in einer Datenanalyse ein- bzw. auszuschalten. In diesem Dokument werden Nutzung und Vorteile dieser Funktion erläutert.

Durch die Funktion können Benutzer gezielt einzelne Teile einer Analyse optimieren oder auch deaktivieren, um das Verhalten der Analyse insgesamt zu beeinflussen.

Vorteile und Anwendungsbeispiele für die Deaktivierung von Modulen in einer Analyse:

- Verbesserte Präzision:
  Durch die Möglichkeit, Algorithmen und Netzwerke in der Datenanalyse ein- oder auszuschalten, können Benutzer die Genauigkeit der Analyse verbessern. Wenn ein Algorithmus nicht optimal funktioniert oder ein bestimmter Netzwerkbereich nicht relevant ist, kann dieser deaktiviert werden, um die Präzision der Analyse insgesamt zu verbessern.
- Optimierung der Ressourcennutzung:
  Die Ein- oder Ausschaltung von Algorithmen oder Netzwerken kann auch dazu beitragen, die Ressourcennutzung zu optimieren. Wenn beispielsweise ein Algorithmus eine hohe Rechenleistung erfordert, kann er bei Bedarf ausgeschaltet werden, um die Ressourcen auf andere Bereiche der Analyse zu konzentrieren.
- Flexibilität:
  Die Möglichkeit, Algorithmen und Netzwerke in der Datenanalyse ein- oder auszuschalten, bietet Benutzern Flexibilität und Kontrolle über die Analyse. Benutzer können die Analyse anpassen, um den spezifischen Anforderungen ihrer Daten gerecht zu werden.
- Debugging:
  Die Funktion ermöglicht es Benutzern auch, Probleme in der Analyse zu debuggen. Wenn ein bestimmter Teil der Analyse fehlerhaft ist, kann der betroffene Algorithmus oder das Netzwerk deaktiviert werden, um das Problem zu isolieren und zu beheben.

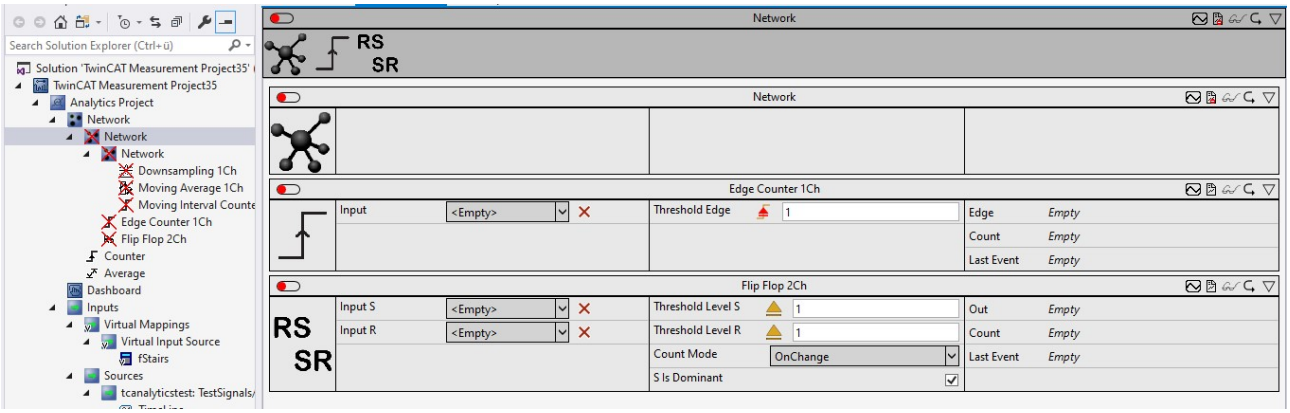
Das Ein- bzw. Ausschalten von Modulen wird über die Steuerungselemente in der linken oberen Ecke des jeweiligen Modules gesteuert.



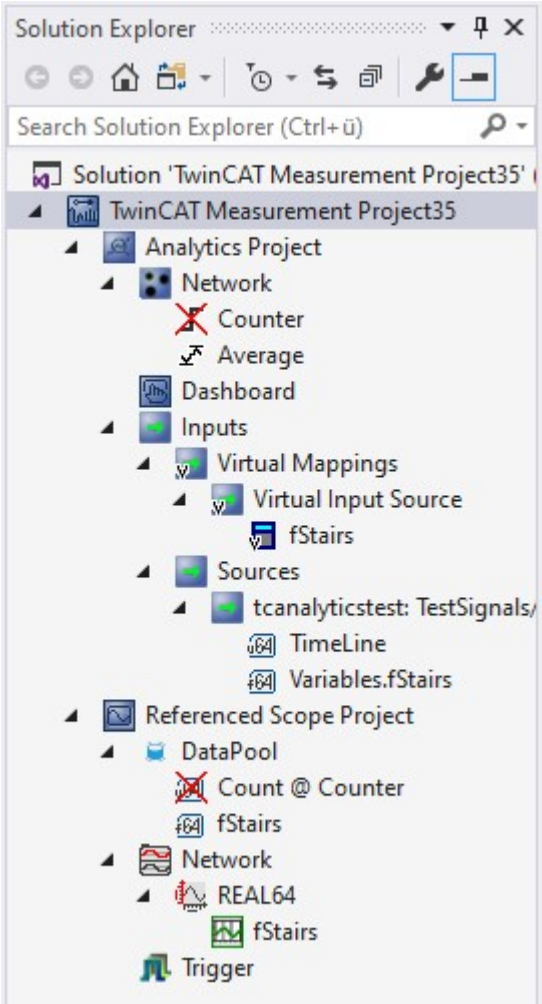
Ausrichtung	Farbe	Status
Rechts	Grün	Enabled
Links	Rot	Disabled

Bei verlinkten Modulen wird die Kette an Links durchgereicht und Eingänge, die mit einem ausgeschalteten Modul verknüpft sind, werden über ein rotes X am Eingang gekennzeichnet.
Disabled Link zwischen zwei Analysealgorithmen:

Wird ein Netzwerk deaktiviert so wird die Abarbeitung aller Algorithmen unterhalb des Netzwerkes sowie dessen Unternetzwerke (Rekursiv) ebenfalls deaktiviert.
Deaktivierte Netzelemente mit ihren Unterelementen:



Sollten Sie ein referenziertes Scope verwenden (siehe Interaktion mit Scope [▶ 356]) so werden verknüpfte Elemente des Scope ebenfalls durch das Deaktivieren beeinflusst.
Durch das Deaktivieren des Algorithmus wird die zugehörige Scope Acquisition ebenfalls deaktiviert:



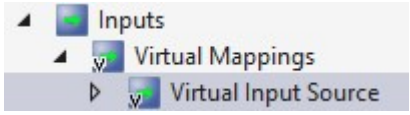
6.3.4 Static Values

Statische Werte werden in Analysealgorithmen häufig verwendet, um Konstanten zu speichern, die während der Ausführung des Algorithmus unverändert bleiben sollen. Diese statischen Werte können beispielsweise als Schwellenwerte, Skalierungsfaktoren oder andere mathematische Konstanten dienen, die im Algorithmus verwendet werden.

Die Verwendung von statischen Werten in Analysealgorithmen bietet mehrere Vorteile. Zum einen kann die Leistung des Algorithmus verbessert werden, indem Berechnungen vermieden werden, die ansonsten, während jeder Ausführung des Algorithmus durchgeführt werden müssten. Stattdessen kann der

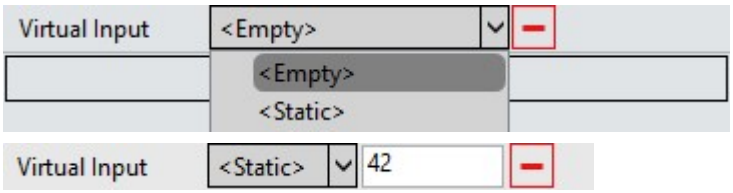
Algorithmus auf den statischen Wert zugreifen, der während der Initialisierung des Algorithmus gesetzt wurde. Zum anderen kann die Lesbarkeit und Verständlichkeit des Algorithmus verbessert werden, indem der Eingangswert klarer strukturiert wird und Konstanten mit sprechenden Bezeichnungen versehen werden.

Statische Werte werden in der TwinCAT Analytics Workbench unter den Virtual Inputs einsortiert, um diese an zentraler Stelle zu erzeugen und später zu verwalten.

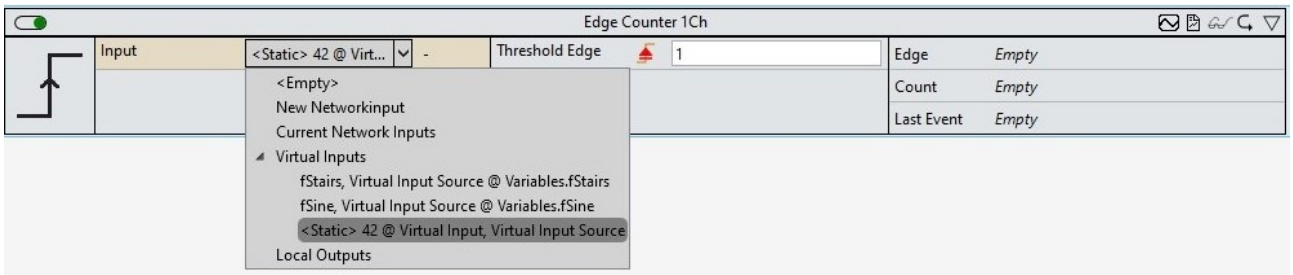


In dem Editor der Virtual Input Source kann über das **+** ein neuer Virtual Input angelegt, oder ein vorhandener bearbeitet werden.

Anschließend kann der virtuelle Input über das Drop-Down-Menü auf einen statischen Wert (hier 42) gesetzt werden.



Dieser statische Wert ist jetzt in der gesamten Konfiguration über den Modul Input auswählbar.



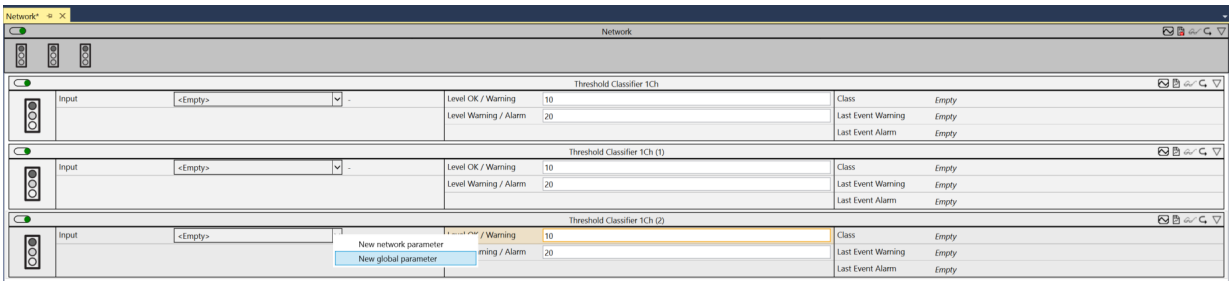
Statische Werte können auch auf Netzwerk Inputs eingesetzt werden. Hier würde die Verknüpfung mit den abhängigen Eingängen der einzelnen Module genau gleich dem Verhalten bei einsprechendem Dateninputs ablaufen.

> **i** Statische Werte in Analysealgorithmen sollten vorsichtig verwendet werden. Wenn sich die Daten oder Parameter ändern, auf die der Algorithmus angewendet wird, kann es erforderlich sein, die statischen Werte anzupassen, um eine optimale Leistung des Algorithmus zu gewährleisten. Daher sollten statische Werte regelmäßig überprüft und gegebenenfalls angepasst werden, um sicherzustellen, dass der Algorithmus korrekt und effektiv arbeitet.

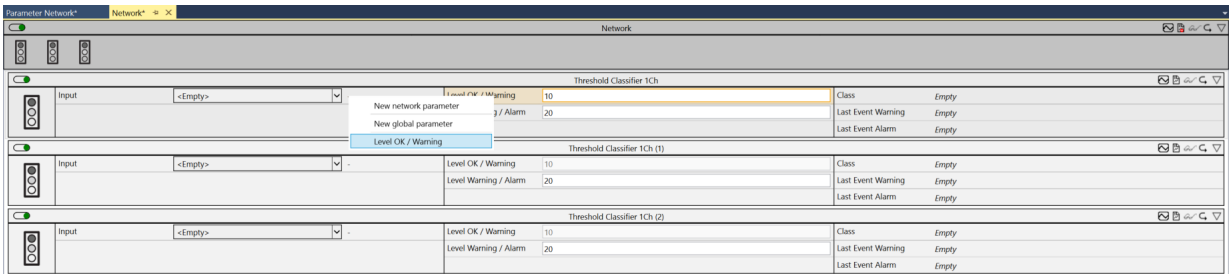
6.3.5 Globale Parameter

Mit der Möglichkeit von globalen Parametern können feste Werte wie beispielsweise ein Kostenfaktor für ein Energiemanagement definiert werden, welche mehrfach in verschiedenen Algorithmen vorkommen können.

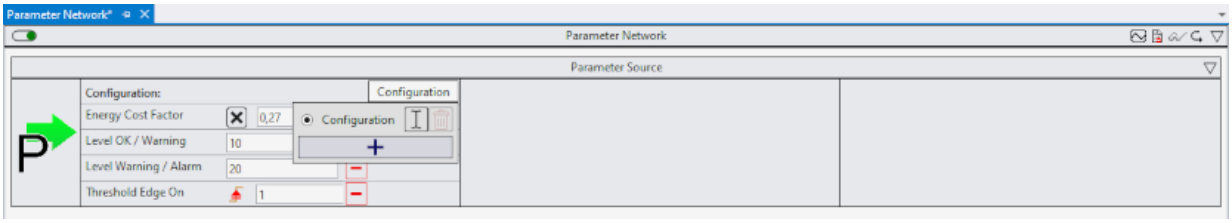
1. Um einen neuen globalen Parameter zu erstellen, klicken Sie mit der rechten Maustaste auf einen bestehenden Parameter und wählen Sie die Option **New global parameter** aus.

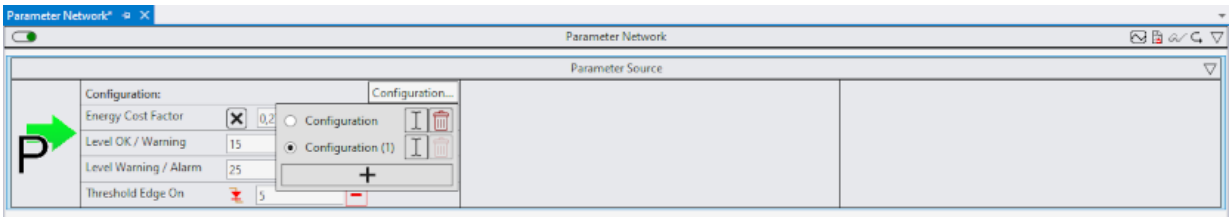


2. Existierende globale Parameter können Sie anschließend über das gleiche Kontextmenü an anderen Algorithmen auswählen.

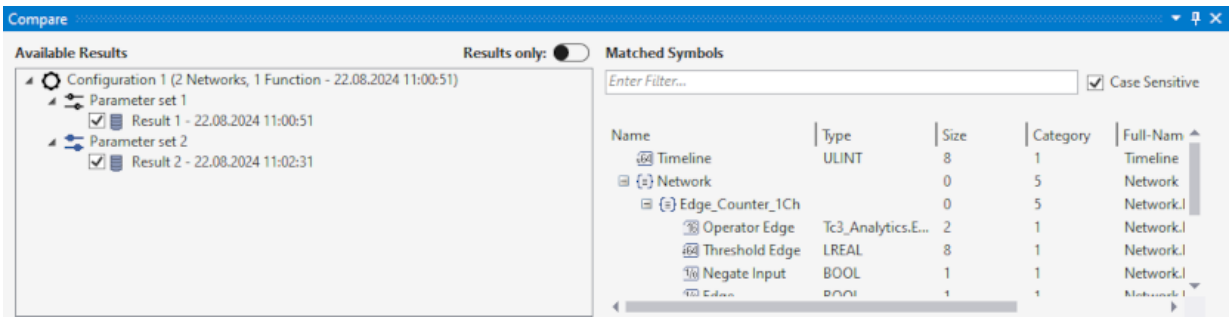


3. Sobald der erste globale Parameter erstellt wurde, wird ein neues **Parameter Network** erzeugt. In diesem können Sie die Parameter an einer zentralen Stelle anpassen. Zusätzlich gibt es die Möglichkeit, mehrere Konfigurationen von Parametern zu definieren, die sich umschalten lassen.





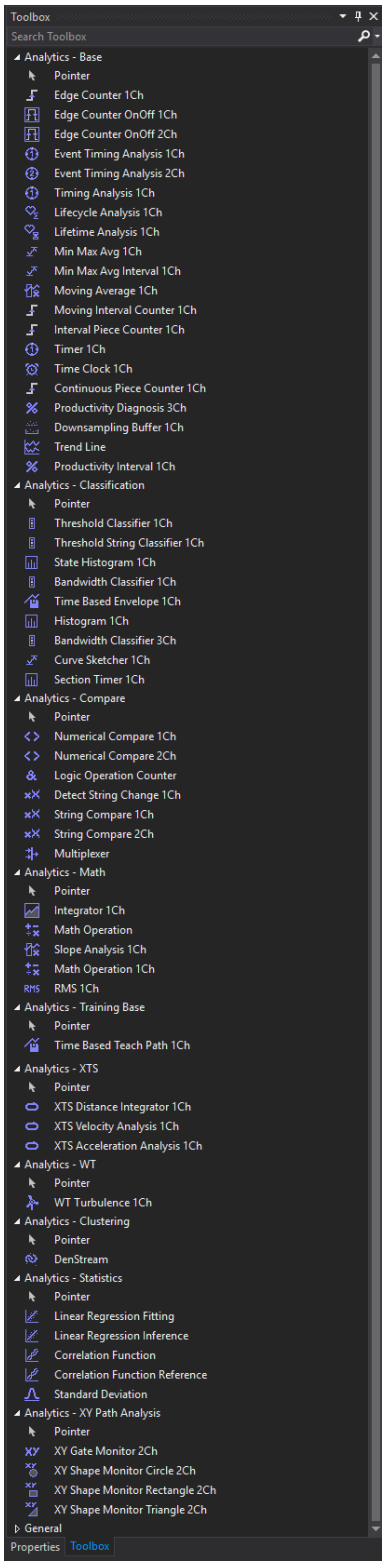
4. Wenn das Projekt nacheinander mit unterschiedlichen Parameterkonfigurationen ausgeführt wird, wird dafür im Analytics Diary [▶ 539] ein neues **Parameter-Set** mit entsprechendem Ergebnisdatensatz unter der bestehenden Projektkonfiguration erstellt.



⇨ Es darf keine weitere Änderung im Projekt vorgenommen werden, andernfalls wird zusätzlich eine neue Projektkonfiguration erstellt.

6.4 Algorithmen

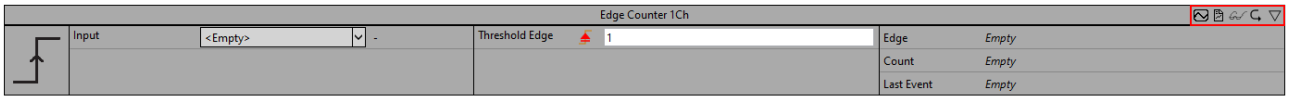
Der TwinCAT Analytics Workbench-Konfigurator beziehungsweise das TwinCAT Analytics Service Tool umfasst verschiedene Analysealgorithmen, die Sie in der Toolbox finden können. Sollten die mehr als 50 Algorithmen nicht den Anforderungen entsprechen, können mithilfe der Analytics Lambda Funktionen anwenderspezifische Algorithmen entwickelt werden. Falls die Toolbox leer ist, müssen Sie das Analytics Projekt auswählen, um die Algorithmen zu sehen.



Gegenwärtig gibt es elf verschiedene Gruppen von Algorithmen: Analytics-Base, Analytics-Classification, Analytics-Compare, Analytics-Math, Analytics-Training Base, Analytics-XTS, Analytics-WT, Analytics-XY Path Analysis, Analytics-Clustering, Analytics-Statistics und Analytics-C++ Lambda Functions. Über die C++

Lambda Funktionen können anwenderspezifische Algorithmen entwickeln werden (siehe C++ Lambda Funktionen [▶ 279]). Zusätzlich besteht die Möglichkeit, Algorithmen aus anderen Bibliotheken zu verwenden, die in die Analytics Toolbox eingebunden sind (siehe Algorithmen aus weiteren TwinCAT Bibliotheken [▶ 310]). Durch den Analytics Custom Toolbox Cleaner [▶ 309] kann die Toolbox hinsichtlich anwenderspezifischen Elementen, wie zum Bespiel den Templates oder den C++ Lambda Funktionen bereinigt werden.

Jeder Algorithmus hat dieselben fünf Icons in der oberen rechten Ecke:



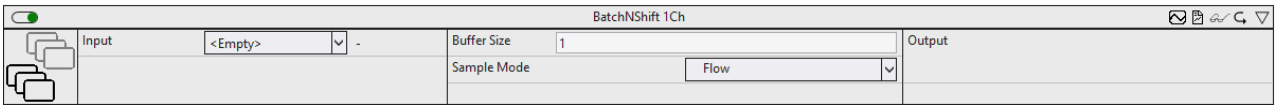
- **Include in Referenced Scope:** Wenn Sie auf das *Included in Referenced Scope* Icon klicken, werden die Algorithmusausgänge zu einem referenzierten Scope-Projekt hinzugefügt.
- **Include in Report:** Wenn Sie auf das *Include in Report* Icon klicken, wird der Algorithmus mit der konfigurierten Visualisierung in einen manuellen Report integriert.
- **Brille:** Wenn Sie auf das Brillen-Icon klicken, können Sie den optionalen Parameter *Enable Execution* sehen. Für diesen Parameter können Sie ein boolsches Signal auswählen, so dass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.
- **Reset-Pfeil:** Wenn Sie auf den Pfeil klicken, werden die Ausgangswerte des spezifischen Algorithmus zurückgesetzt.
- **Minimieren-Pfeil:** Wenn Sie rechts auf den Minimieren-Pfeil klicken, wird der Algorithmus zusammengeklappt.

Im Folgenden werden die verschiedenen Gruppen von Algorithmen beschrieben.

6.4.1 Analytics - Base

Die Algorithmen der Kategorie *Analytics-Base* bieten Basisfunktionen für die Analyse von Prozess- und Anwendungsdaten. Beispiele sind Schwellwerterkennung, Zeitanalyse oder die Berechnung von Durchschnitts-, Minimum- und Maximum-Werten.

6.4.1.1 BatchNShift 1Ch



Der *BatchNShift 1Ch* puffert die Werte des Eingangssignals entsprechend der Puffergröße und des Sample Modus. Die Anzahl der Ausgangskanäle, in denen die gepufferten Eingangswerte gespeichert werden, entspricht der Puffergröße. Mithilfe des Sample Modus kann zwischen zwei verschiedenen Betriebsarten des Algorithmus unterschieden werden. Wird der Sample Modus *Flow* gewählt, so wird ein Ringpuffer beziehungsweise Schieberegister realisiert (Shift). Die Werte werden jeweils nacheinander in den Puffer geschrieben und in jedem Zyklus um eine Stelle des Puffers verschoben. Ist der Puffer voll, fällt der letzte Wert heraus. Im Modus *Wait* wird der Puffer stattdessen immer dann, wenn er vollständig gefüllt ist, komplett geleert und mit neuen Werten befüllt, sodass die Werte in Form von Batches verarbeitet werden (Batch). Zu Beginn einer Analyse wird dabei ebenfalls zunächst gewartet, bis der Puffer vollständig komplett gefüllt ist, erst dann werden die Werte in den Puffer geschrieben. Daher liefert der Baustein erst ab dem Zyklus (*BufferSize + 1)* gültige Werte.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Buffer Size:** Gibt die Größe des Puffers und damit die Anzahl der Werte, die gespeichert werden, an. Die Anzahl der Ausgangskanäle entspricht der Puffergröße.
- **Sample Mode:** Die Werte aus dem Puffer können auf zwei verschiedene Arten an die Ausgangskanäle übergeben werden:

- *Flow*: Der Puffer wird wie ein Ringpuffer gefüllt. Bei Beginn der Analyse werden alle Ausgangswerte auf null gesetzt. Jede Änderung des Ringpuffers wird sofort an die Ausgangskanäle übertragen. Das Flag New Result wird auf TRUE gesetzt, sobald allen Ausgangskanälen ein Wert zugewiesen wurde, und ist immer TRUE, wenn ein neuer Wert im Puffer gespeichert wird.
- *Wait*: Bei Beginn der Analyse oder nach einem Reset werden alle Ausgangskanäle auf null gesetzt. Erst, wenn der interne Puffer voll ist, werden diese Werte an die Ausgangskanäle übertragen und das Flag New Result wird auf TRUE gesetzt. Diese Werte bleiben als Ausgangswerte gültig, bis alle Werte im internen Puffer erneuert sind. Erst dann werden sie an die Ausgangskanäle übertragen.

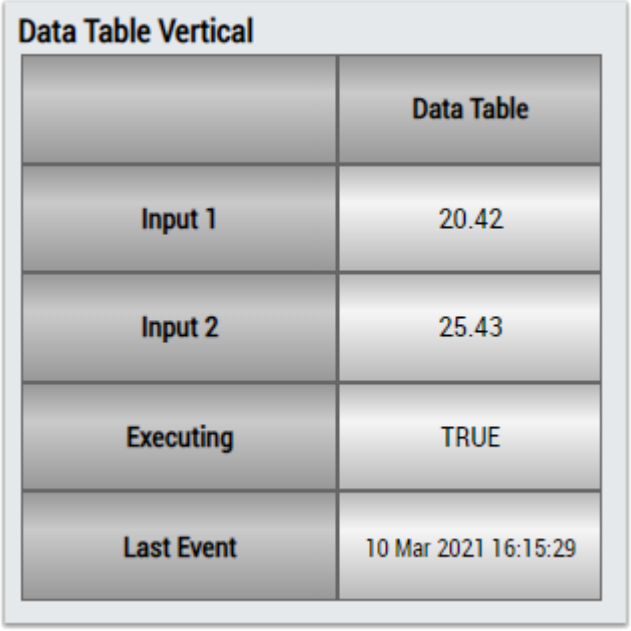
Ausgangswerte

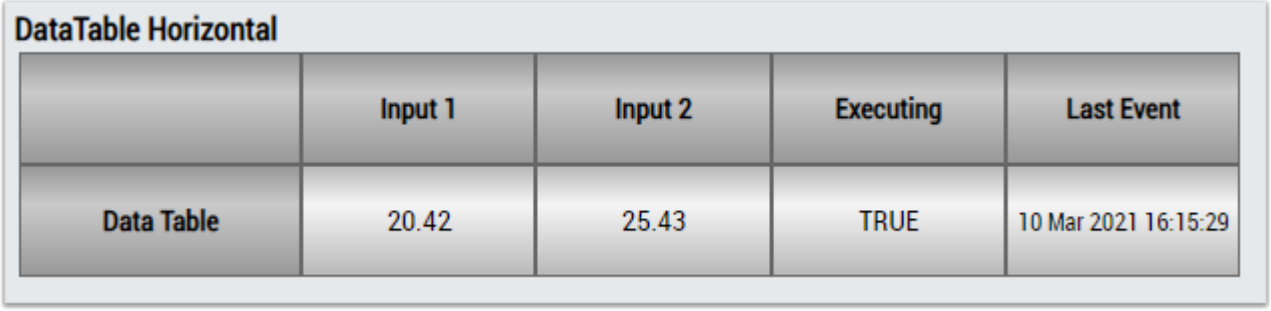
- **Output Value 00..n**: Ergebnisse des BatchNShift-Puffers gemäß dem Sample Mode. Jeder Ausgangskanal repräsentiert dabei einen Puffer-Speicherplatz.

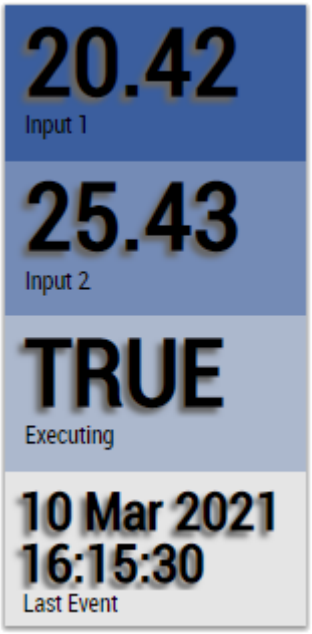
Standard HMI Controls

Für den Algorithmus *BatchNShift 1Ch* sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Table Control oder Multivalue Control visualisiert die Ausgangswerte *Output Value 00..n*

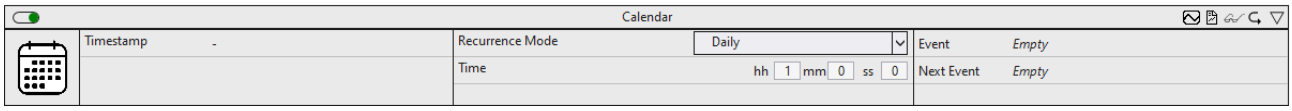






Alternativ können im Algorithmus *BatchNShift 1Ch* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.2 Calendar



Der *Calendar* erzeugt wiederkehrend ein Event zu bestimmten Zeitpunkten. Der Modus zur Wiederholung kann individuell eingestellt werden. Es stehen die Modi der täglichen, wöchentlichen, monatlichen oder jährlichen Wiederholung zur Verfügung.
Als Referenzwert ist ein Zeitstempel erforderlich, da der Algorithmus einen Zeitkontext benötigt, in dem er arbeiten soll. Dieser Referenz-Zeitstempel wird automatisch gesetzt, wenn die Konfiguration einen weiteren Algorithmus umfasst. Daher ist es nicht möglich, den *Calendar* einzeln zu verwenden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Recurrence Mode:** Modus der Wiederholung.
  Daily: Tägliche Wiederholung. Der Zeitpunkt des Events kann eingestellt werden.
  Weekly: Wöchentliche Wiederholung. Der Zeitpunkt des Events, sowie die Wochentage, an denen das Event erzeugt werden soll, können eingestellt werden.
  Monthly: Monatliche Wiederholung. Der Zeitpunkt des Events sowie die Wiederholungsart können eingestellt werden.
  Yearly: Jährliche Wiederholung. Der Zeitpunkt des Events, die Wiederholungsart sowie die Monate, in denen das Event erzeugt werden soll, können eingestellt werden.
- **Time:** Uhrzeit des Events
- **Day Of Week Mask:** Wochentage, an denen ein Event erzeugt werden soll.
- **Recurrence Specification:** Wiederholungsart
  Specific Day: Das Event soll an einem bestimmten Tag erzeugt werden. Dieser kann durch den Parameter *Day* eingestellt werden.
  First: Das Event soll am ersten spezifizierten Tag des Monats erzeugt werden. Die Spezifizierung erfolgt über den Parameter *Day Of Week.*
  Last: Das Event soll am letzten spezifizierten Tag des Monats erzeugt werden. Spezifizierung erfolgt über den Parameter *Day Of Week*
- **Day:** Tag, an dem ein Event erzeugt werden soll.

- **Day Of Week:** Wochentag des Events
  Unspecified: Das Event wird, abhängig vom Parameter *Recurrence Specification*, am ersten oder am letzten Tag des Monats erzeugt.
  Monday..Sunday: Das Event wird, abhängig vom Parameter *Recurrence Specification*, am ersten oder am letzten eingestellten Wochentag des Monats erzeugt.
- **Month Of Year Mask:** Monate, in denen ein Event erzeugt werden soll.

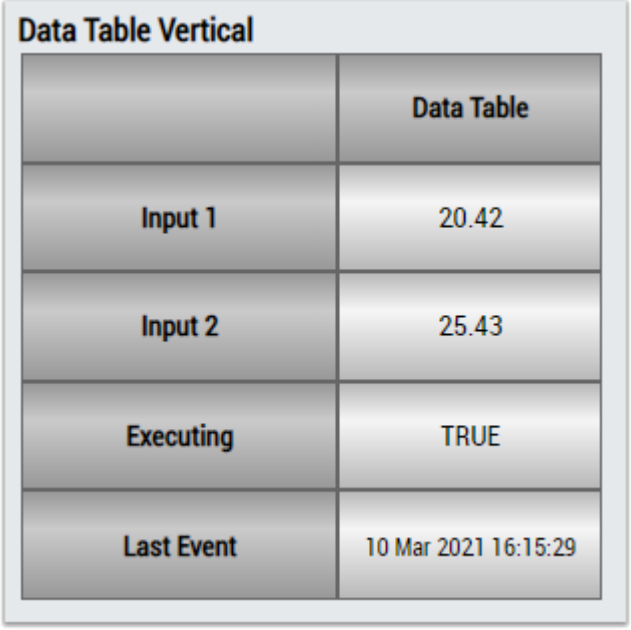
Ausgangswerte

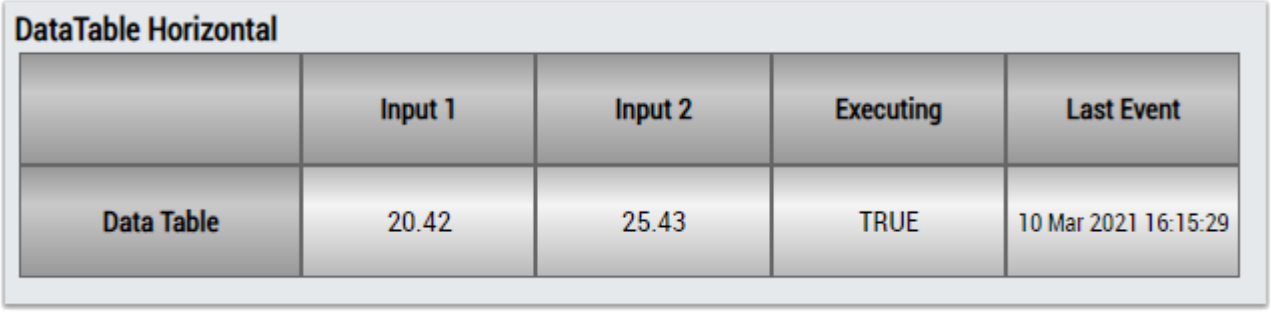
- **Event:** Zeigt *TRUE* zum Zeitpunkt des Events an, ansonsten *FALSE*.
- **Next Event:** Zeigt die verbleibende Zeit bis zur nächsten Event an.

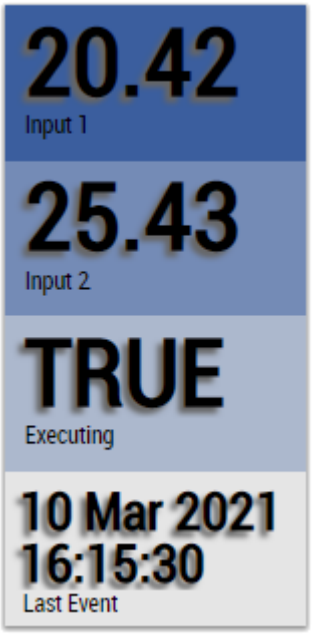
Standard HMI Controls

Für den Algorithmus Calendar sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Event, Next Event.

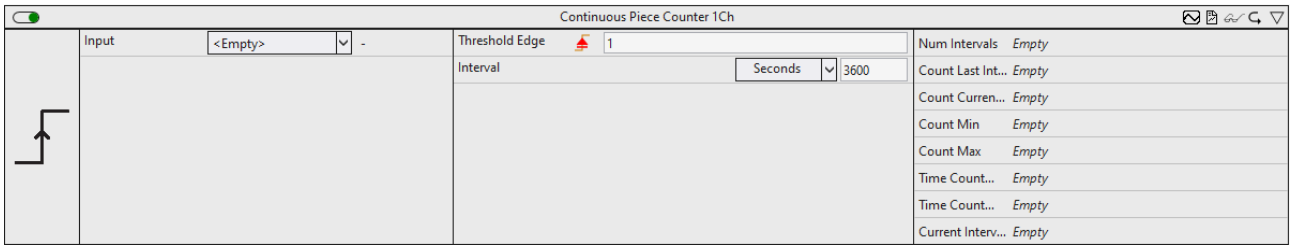






Alternativ können im Algorithmus *Calendar* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.3 Continuous Piece Counter 1Ch



Der *Continuous Piece Counter 1Ch* zählt die Anzahl der Teile innerhalb des konfigurierten Intervalls. Der Zähler zählt hoch, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten Schwelle passiert. Die Berechnung wird neu gestartet, wenn die Zeit des Intervalls abgelaufen ist. Der Algorithmus liefert die Anzahl der Teile, die minimale und maximale Anzahl der Teile sowie die Zeitwerte von Minimum und Maximum.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, zählt der Zähler hoch.
- **Interval:** Zeitintervall, in dem die Werte berechnet werden sollen.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

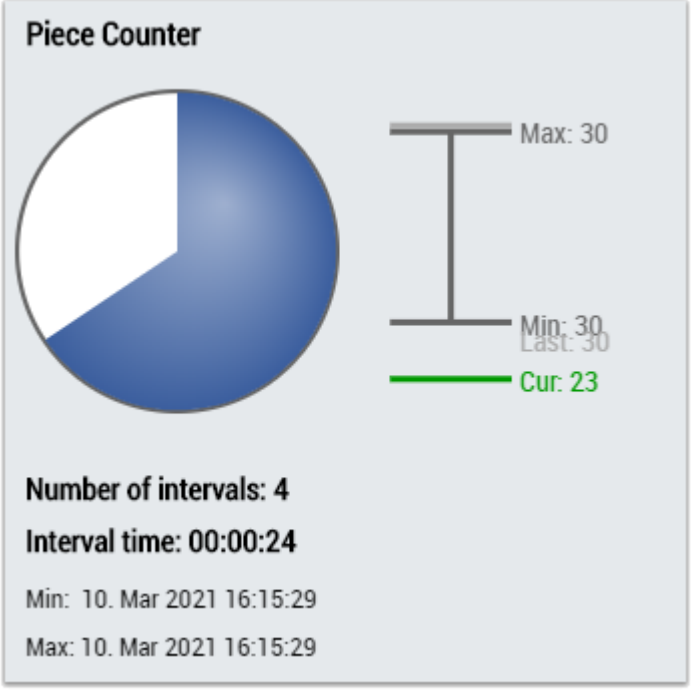
- **Num Interval:** Zeigt die Anzahl der Intervalle an.
- **Count Last Interval:** Zeigt die Anzahl der Teile im letzten Intervall an.
- **Count Current Interval:** Zeigt die Anzahl der Teile im aktuellen Intervall an.
- **Count Min:** Zeigt die Mindestanzahl der Teile in einem Intervall an.
- **Count Max:** Zeigt die Höchstanzahl der Teile in einem Intervall an.
- **Time Count Min:** Zeigt den Zeitwert des Minimums an.

- **Time Count Max:** Zeigt den Zeitwert des Maximums an.
- **Current Interval Time:** Zeigt die Zeit des aktuellen Intervalls an.

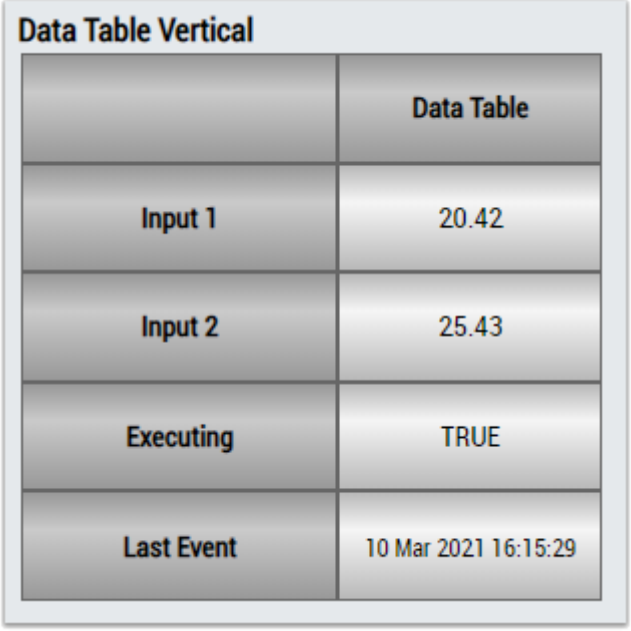
Standard HMI Controls

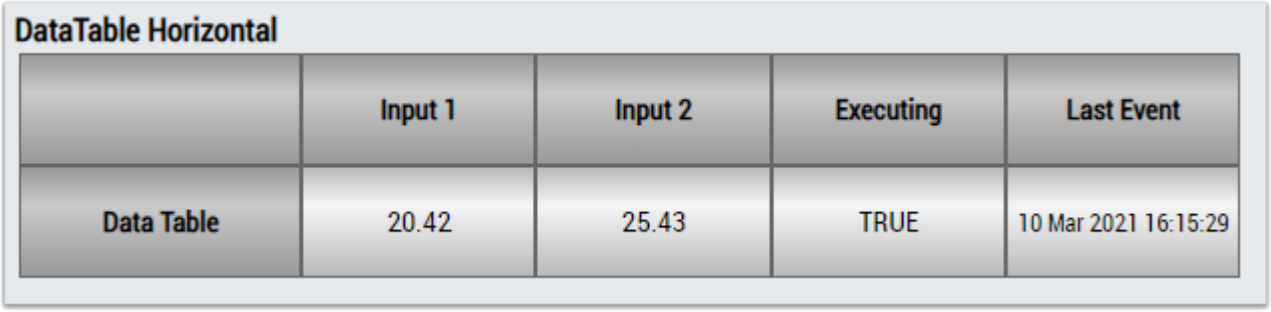
Für den Algorithmus Continuous Piece Counter 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das PieceCounter Control visualisiert die Ausgangswerte Num Intervals, Count Last Interval, Count Current Interval, Count Min, Count Max, Time Count Min, Time Count Max und Time Current Interval.



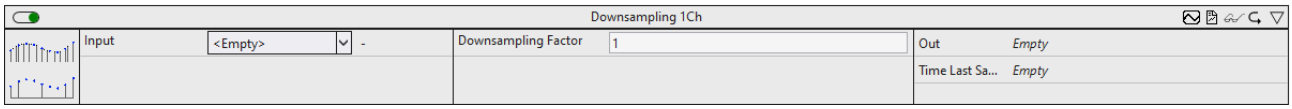
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Num Intervals, Count Last Interval, Count Current Interval, Count Min, Count Max, Time Count Min, Time Count Max, Time Current Interval.





Alternativ können im Algorithmus Continuous Piece Counter 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.4 Downsampling 1Ch



Der *Downsampling 1Ch* verarbeitet die Werte des Eingangskanals mit einem konfigurierbaren Downsampling-Faktor. Dadurch wird ein Downsampling erzielt, sodass das Ausgangssignal einer Repräsentation des Eingangssignals mit einer geringeren Abtastrate entspricht. Dies kann beispielsweise sinnvoll sein, um Trends besser zu erkennen oder aber auch um eine nachträgliche Komprimierung hochabgetasteter Signale vorzunehmen, wenn innerhalb der Analyse lediglich geringere Abtastraten benötigt werden. So lässt sich die Performance der Analyse auf einfache Weise steigern.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Downsampling Factor:** Der Faktor, mit dem das Downsampling vorgenommen wird. Ist der Downsampling Factor beispielsweise 100, so wird lediglich jeder 100. Wert gespeichert. Die Abtastzeit (Sample Time) ist damit 100 mal so groß wie die ursprüngliche Zykluszeit, die zwischen zwei abgetasteten Datenpunkten liegt. Wenn der Downsampling-Faktor auf eins gesetzt wird, werden alle Werte gespeichert.

Ausgangswerte

- **Output Value**: Ausgangssignal mit der um den Downsampling Factor geringeren Abtastrate.

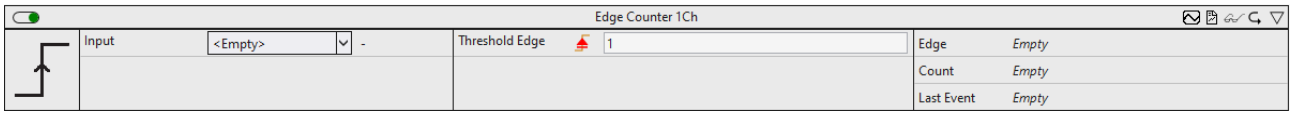
Standard HMI Controls

Für den Algorithmus *Downsampling 1Ch* sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SingleValue Control visualisiert den Ausgangswerte *Output Value*.

Alternativ können im Algorithmus *Downsampling 1Ch* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.5 Edge Counter 1Ch



Der *Edge Counter 1Ch* zählt die Anzahl der ausgelösten Ereignisse. Ein Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten Schwelle passiert.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Ereignis ausgelöst.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

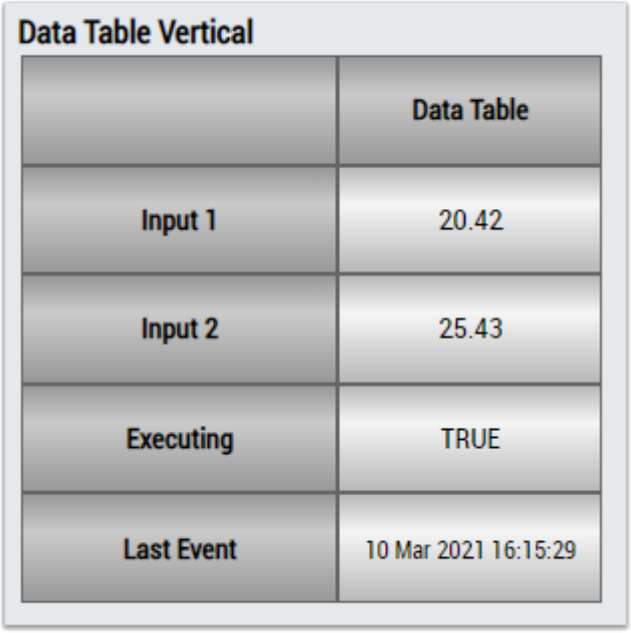
- **Edge:** Zeigt zu dem Zeitpunkt, zu dem das Ereignis ausgelöst wird, *TRUE* an, ansonsten *FALSE*.
- **Count:** Zählt die Anzahl der ausgelösten Ereignisse.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

Standard HMI Controls

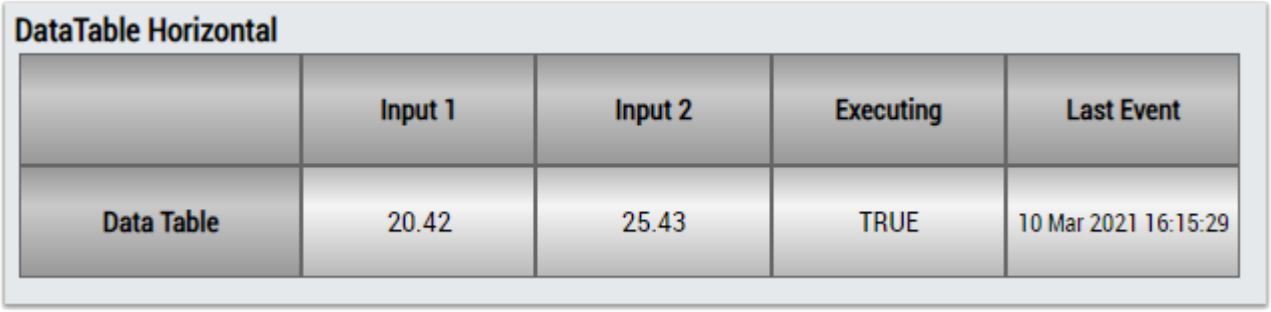
Für den Algorithmus Edge Counter 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.

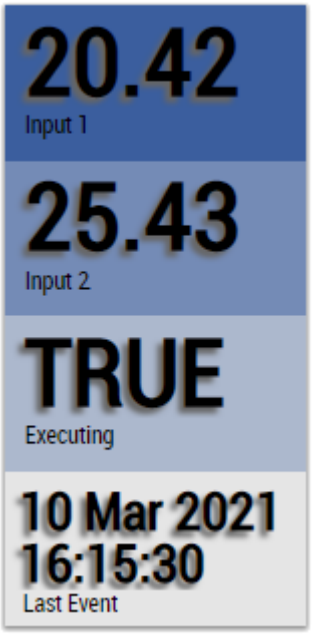
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Edge, Count, Last Event.



	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

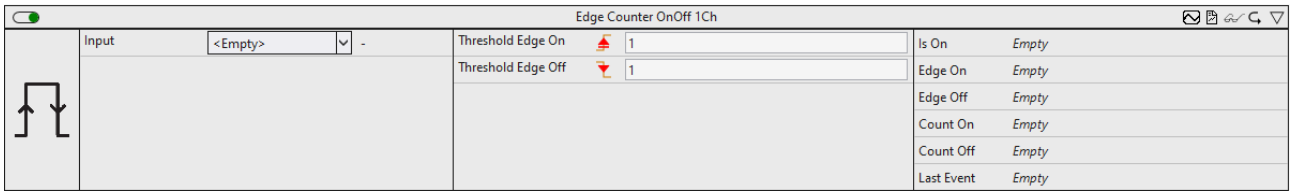


	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Edge Counter 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.6 Edge Counter On Off 1Ch



Der *Edge Counter On Off 1Ch* zählt die Anzahl der ausgelösten On- und Off-Ereignisse. Ein On-Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten On-Schwelle passiert, und ein Off-Ereignis wird ausgelöst, wenn die Off-Schwelle von demselben Signal passiert wird.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold On:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das On-Ereignis ausgelöst.
- **Threshold Off:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Off-Ereignis ausgelöst.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

- **Is On:** Zeigt innerhalb der Zeitspanne zwischen dem On-Ereignis und dem Off-Ergebnis *TRUE* an, ansonsten *FALSE*.
- **Edge On:** Zeigt bei einer steigenden Flanke TRUE an.
- **Edge Off:** Zeigt bei einer fallenden Flanke TRUE an.
- **Count On:** Zählt die Anzahl der ausgelösten On-Ereignisse.
- **Count Off:** Zählt die Anzahl der ausgelösten Off-Ereignisse.

- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

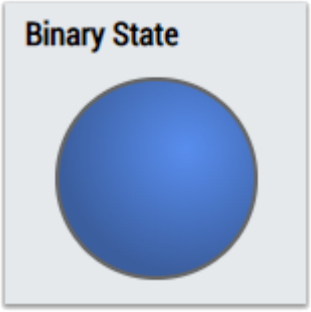
Standard HMI Controls

Für den Algorithmus Edge Counter On Off 1Ch stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das EdgeCounterOnOff Control visualisiert die Ausgangswerte Is On, Count On, Count Off und Last Event.

2. Das SingleValue Control visualisiert die Ausgangswerte Count On und Last Event.

3. Das BinaryState Control visualisiert den Ausgangswert Is On.



4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Flanken (Is On, Edge On, Edge Off), Count On, Count Off, Last Event.

Data Table Vertical

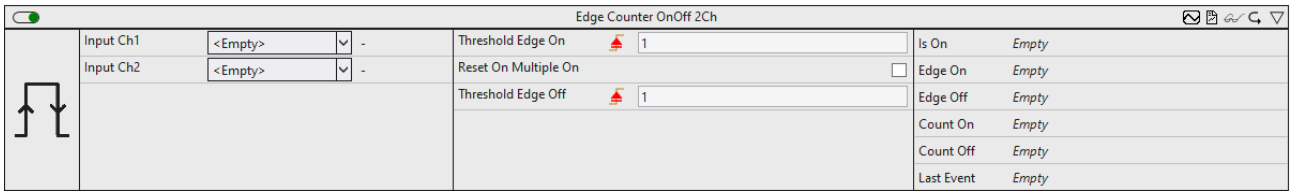
	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

Alternativ können im Algorithmus Edge Counter On Off 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.7 Edge Counter On Off 2Ch



Der *Edge Counter On Off 2Ch* zählt die Anzahl der ausgelösten On- und Off-Ereignisse. Ein On-Ereignis wird ausgelöst, wenn das Signal des ersten Eingangskanals die konfigurierte Flanke bei einer bestimmten On-Schwelle passiert, und ein Off-Ereignis wird ausgelöst, wenn die Off-Schwelle vom Signal des zweiten Kanals passiert wird.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold On:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das On-Ereignis ausgelöst.
- **Reset On Multiple On:** Wenn die Checkbox aktiviert ist, zählt der „Count On“-Zähler bei jedem On-Ereignis hoch. Anderenfalls werden die On-Ereignisse nur nach einem Reset des Zählers gezählt (Off-Ereignis).
- **Threshold Off:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Off-Ereignis ausgelöst.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

- **Is On:** Zeigt innerhalb der Zeitspanne zwischen dem On-Ereignis und dem Off-Ergebnis *TRUE* an, ansonsten *FALSE*.
- **Edge On:** Zeigt bei einer steigenden Flanke TRUE an.
- **Edge Off:** Zeigt bei einer fallenden Flanke TRUE an.
- **Count On:** Zählt die Anzahl der ausgelösten On-Ereignisse.
- **Count Off:** Zählt die Anzahl der ausgelösten Off-Ereignisse.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

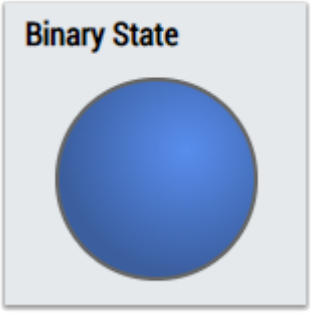
Standard HMI Controls

Für den Algorithmus Edge Counter On Off 2Ch stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

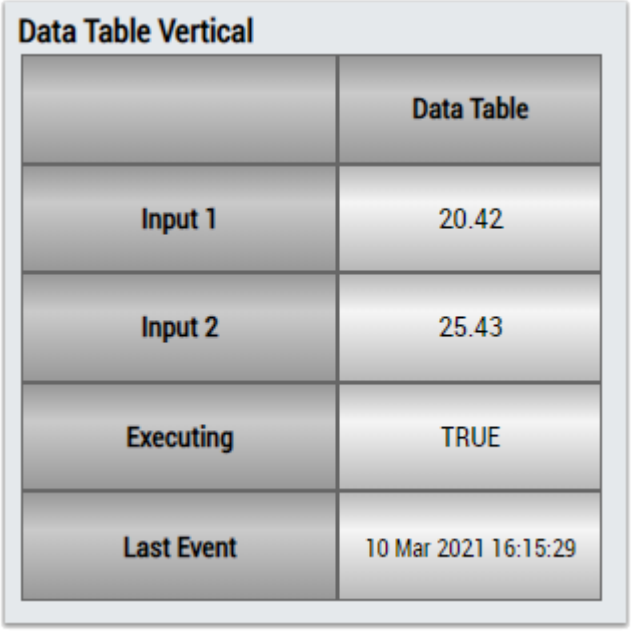
1. Das EdgeCounterOnOff Control visualisiert die Ausgangswerte Is On, Count On, Count Off und Last Event.

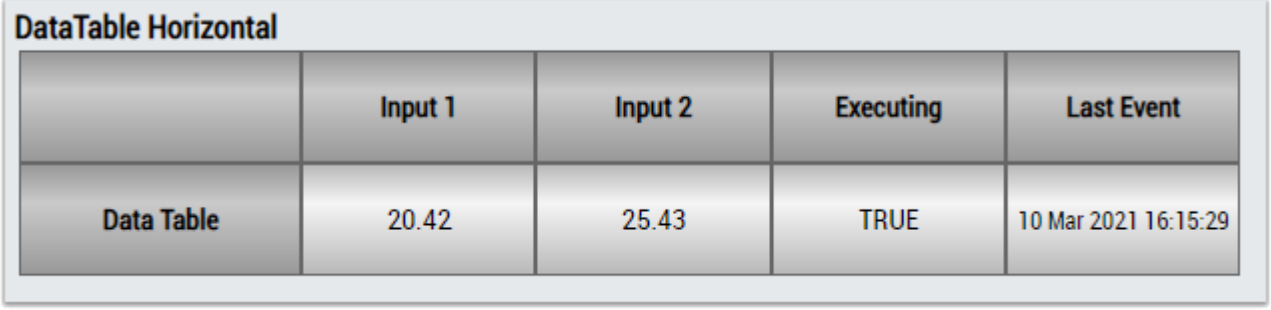
2. Das SingleValue Control visualisiert die Ausgangswerte Count On und Last Event.

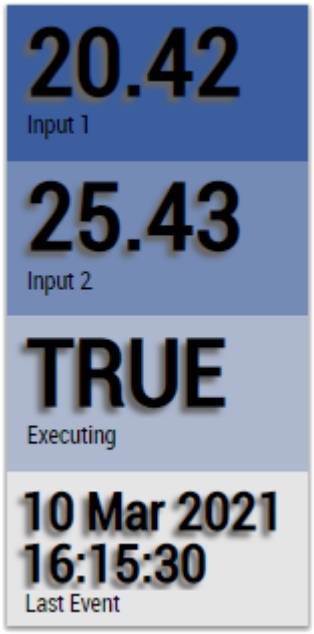
3. Das BinaryState Control visualisiert den Ausgangswert Is On.



4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Flanken (Is On, Edge On, Edge Off), Count On, Count Off, Last Event.

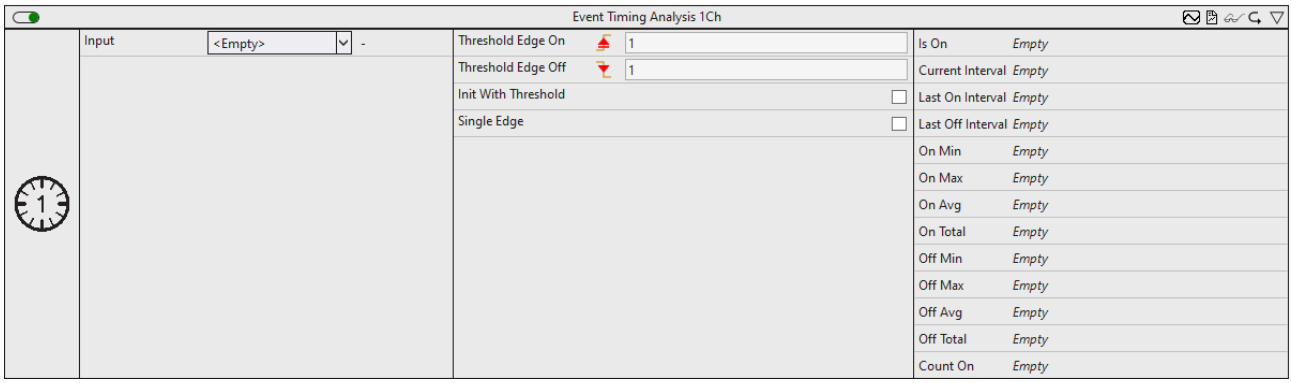






Alternativ können im Algorithmus Edge Counter On Off 2Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.8 Event Timing Analysis 1Ch



Der *Event Timing Analysis 1Ch* misst Zeitdifferenzen zwischen On- und Off-Ereignissen und zählt die Anzahl der ausgelösten Ereignisse. Ein On-Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten On-Schwelle passiert, und ein Off-Ereignis wird ausgelöst, wenn die Off-Schwelle von demselben Signal passiert wird.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold On:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das On-Ereignis ausgelöst.
- **Threshold Off:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Off-Ereignis ausgelöst.
- **Init With Threshold:** Ist der Wert TRUE, verwendet der Algorithmus zur Initialisierung des internen Zustandes einen Schwellwert, anstatt auf eine Flanke zu warten.
- **Single Edge:** Ist der Wert TRUE, erfolgt die Ausführung des Algorithmus nur auf Basis der Flanken des On-Ereignisses.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

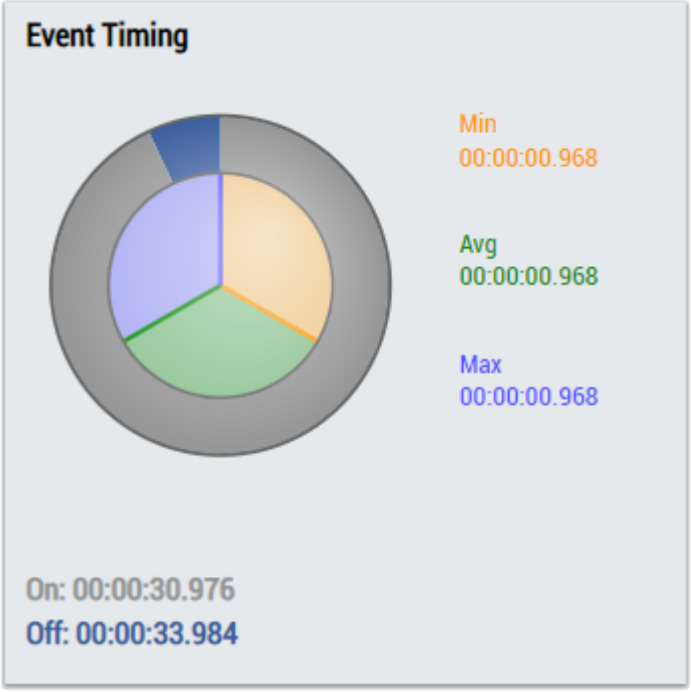
Ausgangswerte

- **Is On:** Zeigt innerhalb der Zeitspanne zwischen dem On-Ereignis und dem Off-Ergebnis *TRUE* an, ansonsten *FALSE*.
- **Current Interval:** Zeigt die Zeit des aktuellen Intervalls an.
- **Last On Interval:** Zeitspanne des zuletzt abgeschlossenen Intervalls zwischen On-Ereignis und Off-Ereignis.
- **Last Off Interval:** Zeitspanne des zuletzt abgeschlossenen Intervalls zwischen Off-Ereignis und On-Ereignis.
- **On Min:** Minimale Zeit zwischen On-Ereignis und Off-Ereignis.
- **On Max:** Maximale Zeit zwischen On-Ereignis und Off-Ereignis.
- **On Avg:** Durchschnittliche Zeit zwischen On-Ereignis und Off-Ereignis.
- **On Total:** Gesamtzeit zwischen On-Ereignissen und Off-Ereignissen.
- **Off Min:** Minimale Zeit zwischen Off-Ereignis und On-Ereignis.
- **Off Max:** Maximale Zeit zwischen Off-Ereignis und On-Ereignis.
- **Off Avg:** Durchschnittliche Zeit zwischen Off-Ereignis und On-Ereignis.
- **Off Total:** Gesamtzeit zwischen Off-Ereignissen und On-Ereignissen.
- **Count On:** Zählt die Anzahl der ausgelösten On-Ereignisse.

Standard HMI Controls

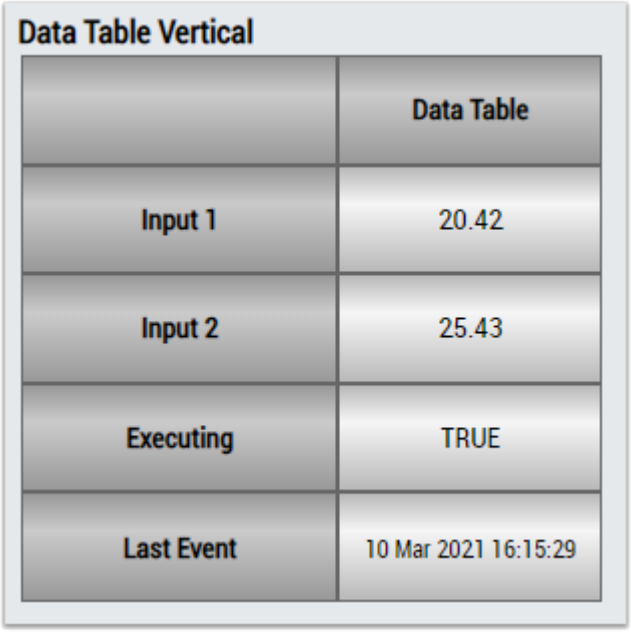
Für den Algorithmus Event Timing Analysis 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

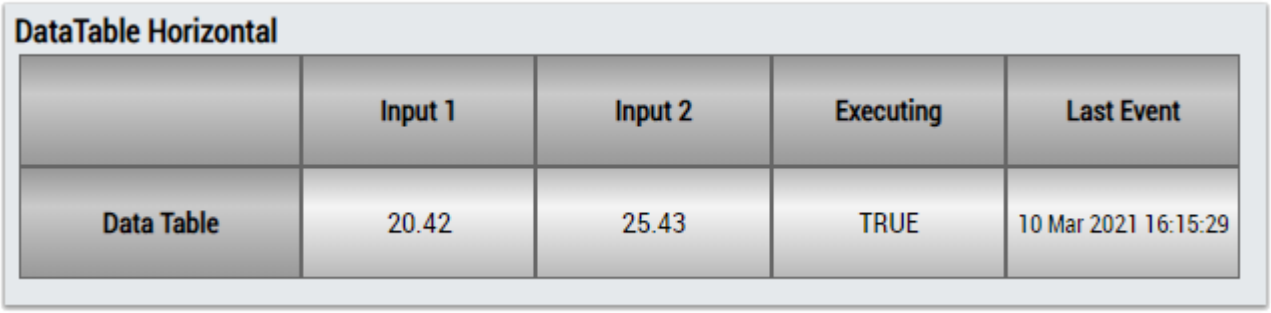
1. Das EventTiming Control visualisiert die Ausgangswerte Is On, Count On, Current Interval, On Min, On Max, On Avg, On Total, Off Min, Off Max, Off Avg und Off Total.

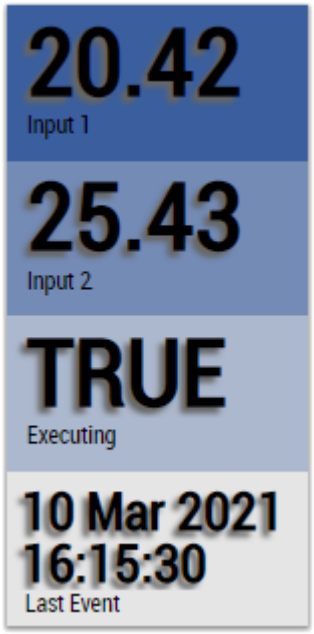


2. Das SingleValue Control visualisiert den Ausgangswert Count On.

3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Is On, Count On, Current Interval, On Min, On Max, On Avg, On Total, Off Min, Off Max, Off Avg, Off Total.

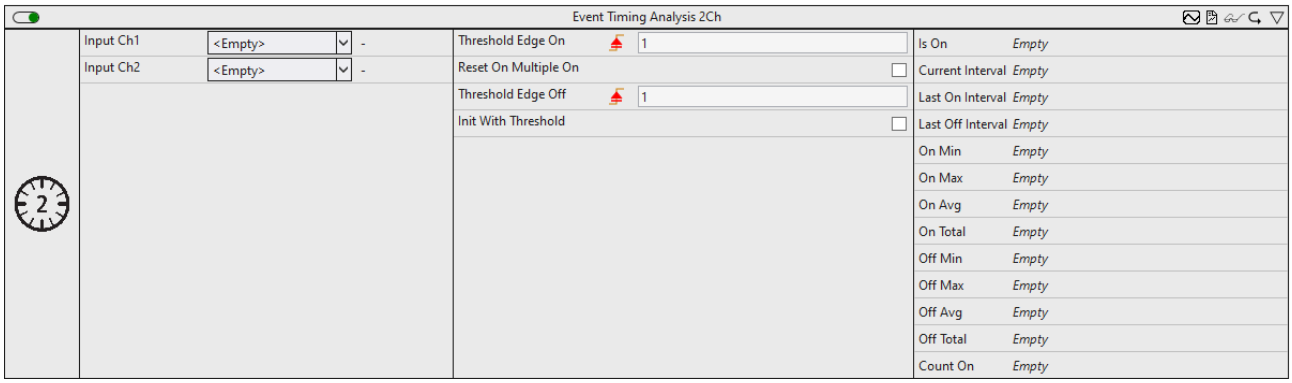






Alternativ können im Algorithmus Event Timing Analysis 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.9 Event Timing Analysis 2Ch



Der *Event Timing Analysis 2Ch* misst Zeitdifferenzen zwischen On- und Off-Ereignissen und zählt die Anzahl der ausgelösten Ereignisse. Ein On-Ereignis wird ausgelöst, wenn das Signal des ersten Eingangskanals die konfigurierte Flanke bei einer bestimmten On-Schwelle passiert, und ein Off-Ereignis wird ausgelöst, wenn die Off-Schwelle vom Signal des zweiten Kanals passiert wird.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold On:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das On-Ereignis ausgelöst.
- **Reset On Multiple On:** Wenn die Checkbox aktiviert ist, zählt der „Count On“-Zähler bei jedem On-Ereignis hoch. Anderenfalls werden die On-Ereignisse nur nach einem Reset des Zählers gezählt (Off-Ereignis).
- **Threshold Off:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Off-Ereignis ausgelöst.
- **Init With Threshold:** Ist der Wert TRUE, verwendet der Algorithmus zur Initialisierung des internen Zustandes einen Schwellwert, anstatt auf eine Flanke zu warten.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

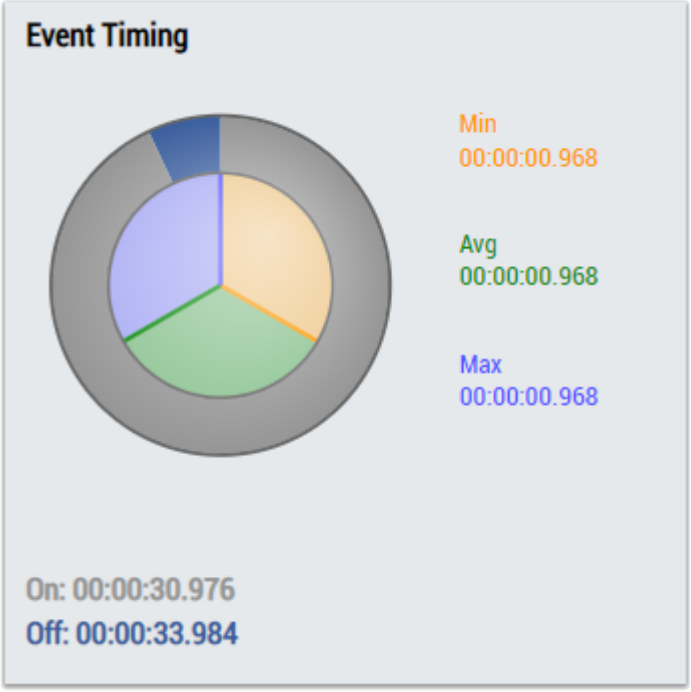
Ausgangswerte

- **Is On:** Zeigt innerhalb der Zeitspanne zwischen dem On-Ereignis und dem Off-Ergebnis *TRUE* an, ansonsten *FALSE*.
- **Current Interval:** Zeigt die Zeit des aktuellen Intervalls an.
- **Last On Interval:** Zeitspanne des zuletzt abgeschlossenen Intervalls zwischen On-Ereignis und Off-Ereignis.
- **Last Off Interval:** Zeitspanne des zuletzt abgeschlossenen Intervalls zwischen Off-Ereignis und On-Ereignis.
- **On Min:** Minimale Zeit zwischen On-Ereignis und Off-Ereignis.
- **On Max:** Maximale Zeit zwischen On-Ereignis und Off-Ereignis.
- **On Avg:** Durchschnittliche Zeit zwischen On-Ereignis und Off-Ereignis.
- **On Total:** Gesamtzeit zwischen On-Ereignissen und Off-Ereignissen.
- **Off Min:** Minimale Zeit zwischen Off-Ereignis und On-Ereignis.
- **Off Max:** Maximale Zeit zwischen Off-Ereignis und On-Ereignis.
- **Off Avg:** Durchschnittliche Zeit zwischen Off-Ereignis und On-Ereignis.
- **Off Total:** Gesamtzeit zwischen Off-Ereignissen und On-Ereignissen.
- **Count On:** Zählt die Anzahl der ausgelösten On-Ereignisse.

Standard HMI Controls

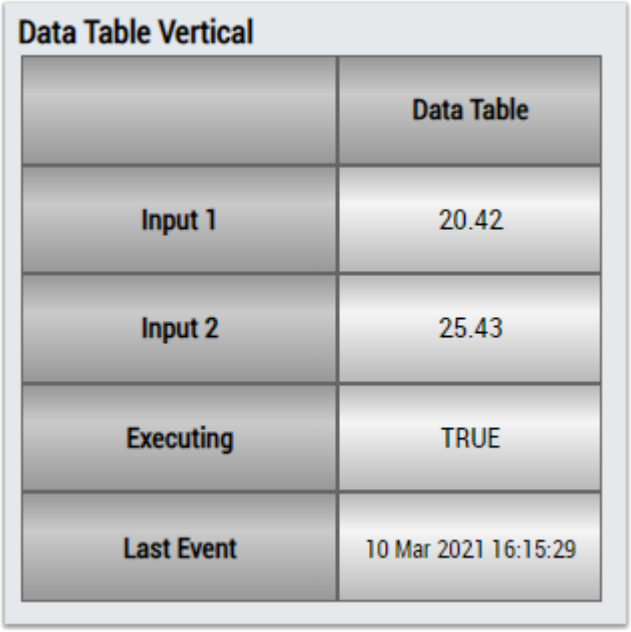
Für den Algorithmus Event Timing Analysis 2Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

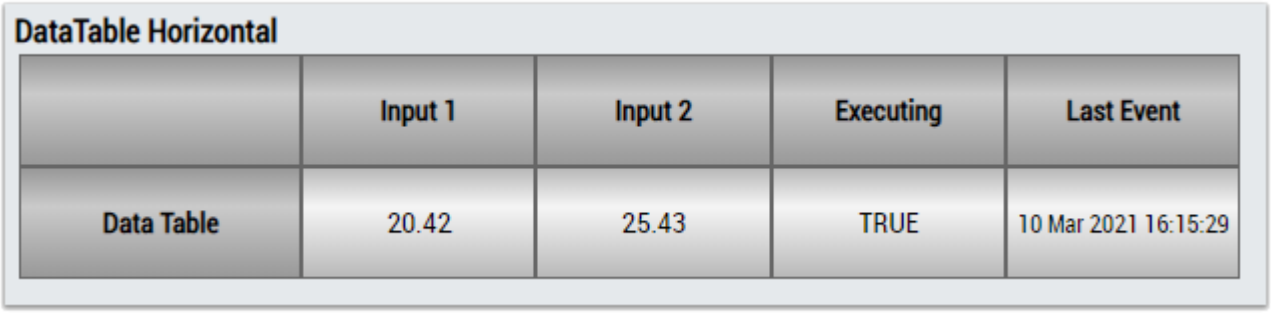
1. Das EventTiming Control visualisiert die Ausgangswerte Is On, Count On, Current Interval, On Min, On Max, On Avg, On Total, Off Min, Off Max, Off Avg und Off Total.

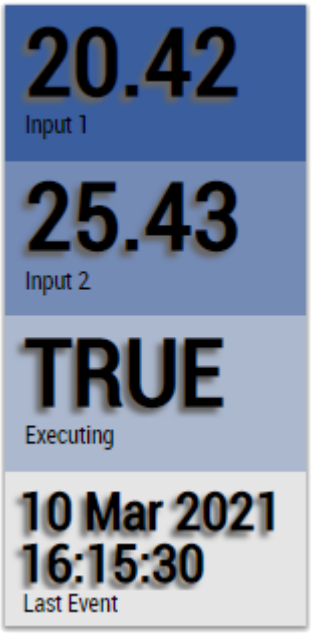


2. Das SingleValue Control visualisiert den Ausgangswert Count On.

3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Is On, Count On, Current Interval, On Min, On Max, On Avg, On Total, Off Min, Off Max, Off Avg, Off Total.

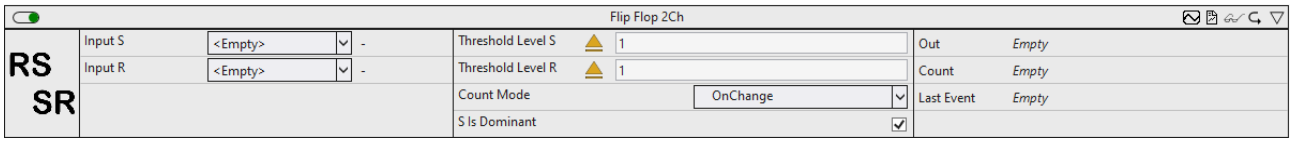






Alternativ können im Algorithmus Event Timing Analysis 2Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.10 Flip Flop 2Ch



Der *Flip Flop 2Ch* realisiert eine bistabile Kippstufe. Die Dominanz für das Setzen (RS) oder das Rücksetzens (SR) des Ausgangswertes kann konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Operator:** Gibt an, ob der Eingangswert größer, größer oder gleich, gleich, kleiner oder gleich, kleiner oder ungleich der Schwelle sein soll.
- **Threshold Level S:** Schwelle des Signals für das Setzen.
- **Threshold Level R:** Schwelle des Signals für das Rücksetzen.
- **Count Mode:** Modus des Ergebniszählers.
  OnChange: Der Zähler zählt jedes Mal, wenn sich das Ergebnis auf TRUE ändert.
  Cyclic: Der Zähler inkrementiert jeden Zyklus, wenn die Bedingung *TRUE* ist.
- **S Is Dominant:** Die Domianz des Setzens (RS) oder des Rücksetzens (SR) kann konfiguriert werden.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

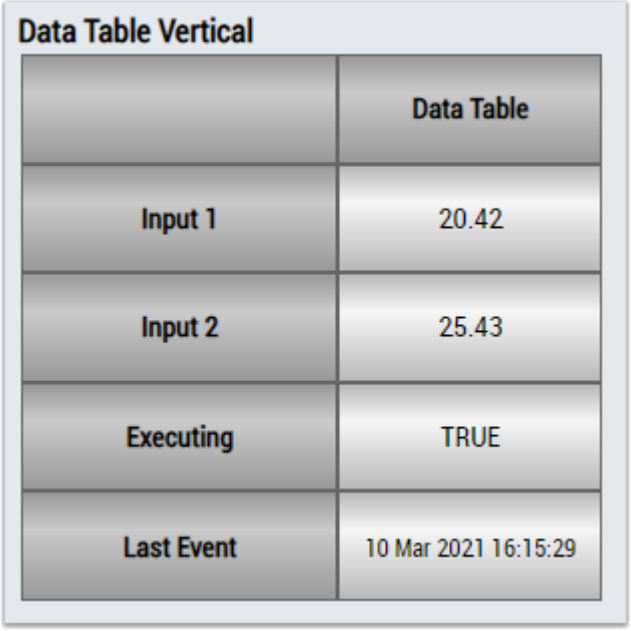
- **Out:** Ergebnis der bistabilen Kippstufe.
- **Count:** Wird inkrementiert, wenn der Ausgangswert TRUE ist. Das Verhalten ist abhängig von dem Parameter *Count Mode*.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

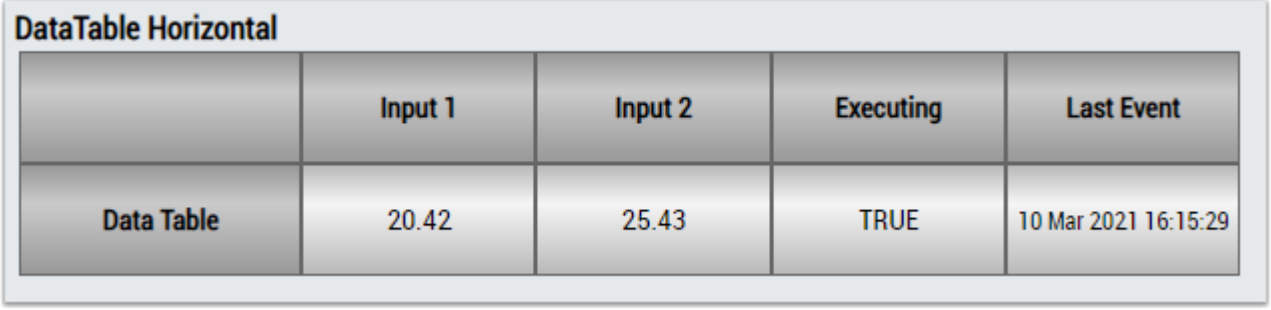
Standard HMI Controls

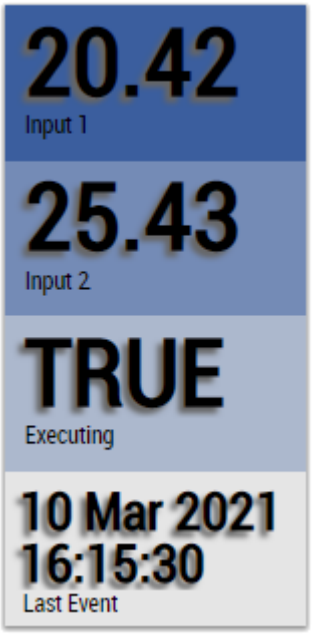
Für den Algorithmus *Flip Flop 2Ch* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das BinaryState Control visualisiert den Ausgangswert *Out*.

2. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte: *Out*, *Count*, *Last Event*.

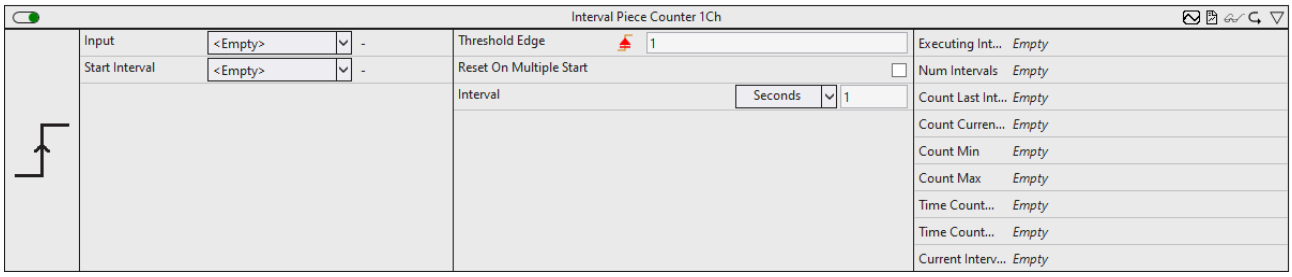






Alternativ können im Algorithmus *Flip Flop 2Ch* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.11 Interval Piece Counter 1Ch



Der *Interval Piece Counter 1Ch* zählt die Anzahl der ausgelösten Ereignisse innerhalb eines konfigurierten Intervalls, das beginnt, wenn der Wert des Flags Start Intervall *TRUE ist.* Ein Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten Schwelle passiert. Die Berechnung wird neu gestartet, wenn die Zeit des Intervalls abgelaufen ist und der Wert des Flags Start Intervall wieder *True* ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Ereignis ausgelöst.
- **Reset On Multiple Start**: Wenn die Checkbox aktiviert ist, beginnt das Intervall neu, wenn das Flag Start Intervall wieder *TRUE* wird. Ansonsten beginnt das Intervall automatisch neu, wenn die Zeit abgelaufen ist.
- **Interval:** Zeitintervall, in dem die Werte berechnet werden sollen.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

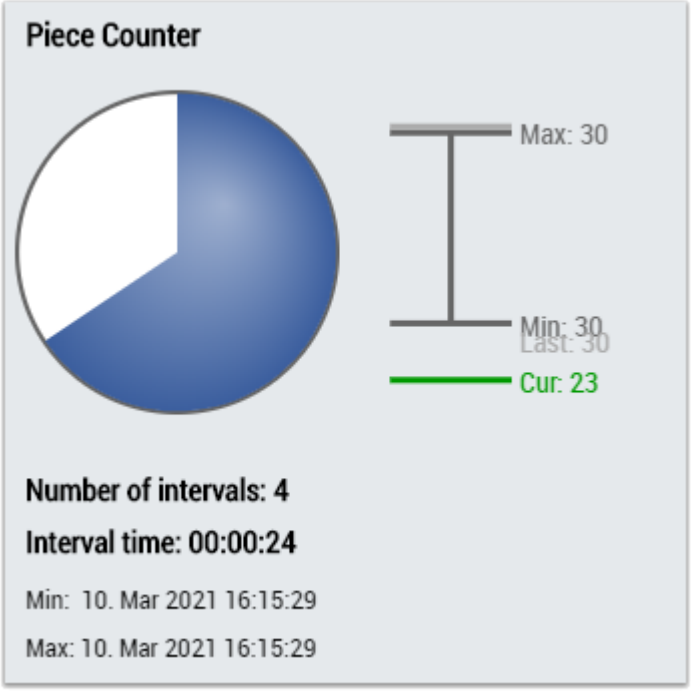
- **Executing Interval:** Zeigt *True* an, wenn die Berechnung aktiv ist und das Intervall läuft, ansonsten *False*.
- **Num Intervals:** Zeigt die Anzahl der Intervalle an.

- **Count Last Interval:** Zeigt die Anzahl der ausgelösten Ereignisse im letzten Intervall an.
- **Count Current Interval:** Zeigt die Anzahl der ausgelösten Ereignisse im aktuellen Intervall oder, wenn die Berechnung derzeit inaktiv ist, die Anzahl der ausgelösten Ereignisse im letzten Intervall an.
- **Count Min:** Zeigt das Minimum der ausgelösten Ereignisse in einem Intervall an.
- **Count Max:** Zeigt das Maximum der ausgelösten Ereignisse in einem Intervall an.
- **Time Count Min:** Zeigt den Zeitwert des Minimums an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Time Count Max:** Zeigt den Zeitwert des Maximums an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Current Interval Time:** Zeigt die Zeit des aktuellen Intervalls an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

Standard HMI Controls

Für den Algorithmus Interval Piece Counter 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das PieceCounter Control visualisiert die Ausgangswerte Num Intervals, Count Last Interval, Count Current Interval, Count Min, Count Max, Time Count Min, Time Count Max und Time Current Interval.



2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Num Intervals, Count Last Interval, Count Current Interval, Count Min, Count Max, Time Count Min, Time Count Max, Time Current Interval.

Data Table Vertical

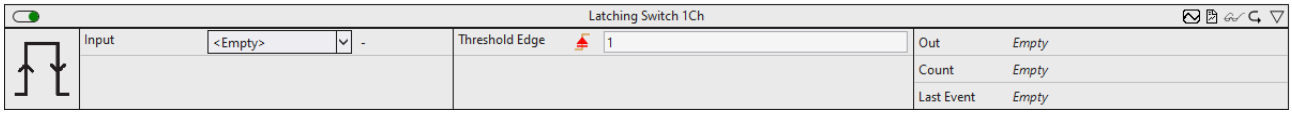
	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

Alternativ können im Algorithmus Interval Piece Counter 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.12 Latching Switch 1Ch



Der *Latching Switch 1Ch* realisiert einen virtuellen Stromstoßschalter. Der Ausgang wechselt zwischen TRUE und FALSE bei jeder erkannten Flanke am Eingang.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob auf eine steigende oder fallende Flanke am Eingang reagiert werden soll
- **Threshold Edge:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Ereignis ausgelöst.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

- **Out:** Ergebnis des virtuellen Stromstoßschalters.
- **Count:** Wird inkrementiert, wenn der Ausgangswert wechselt.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

Standard HMI Controls

Für den Algorithmus *Latching Switch 1Ch* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.

2. Das BinaryState Control visualisiert den Ausgangswert *Out*.

3. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte.

Data Table Vertical

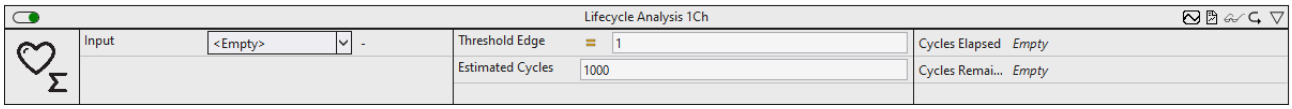
	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

Alternativ können mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.13 Lifecycle Analysis 1Ch



Der *Lifecycle Analysis 1Ch* berechnet die abgelaufenen und die geschätzten verbleibenden Zyklen eines Geräts. Wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten Schwelle passiert, werden die abgelaufenen Zyklen erhöht und die verbleibenden Zyklen verringert.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird ein Ereignis ausgelöst.
- **Estimated Cycles:** Geschätzte Zyklen in der Lebensdauer des jeweiligen Geräts.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

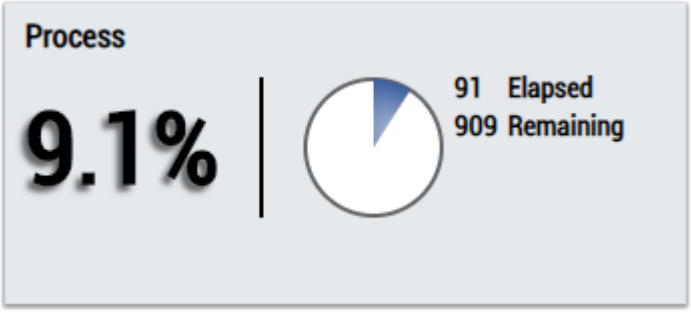
Ausgangswerte

- **Elapsed Cycles:** Zählt die Anzahl der Zyklen, die bereits abgelaufen sind.
- **Remaining Cycles:** Zeigt die verbleibenden Zyklen des Geräts als Differenz der geschätzten und abgelaufenen Zyklen an.

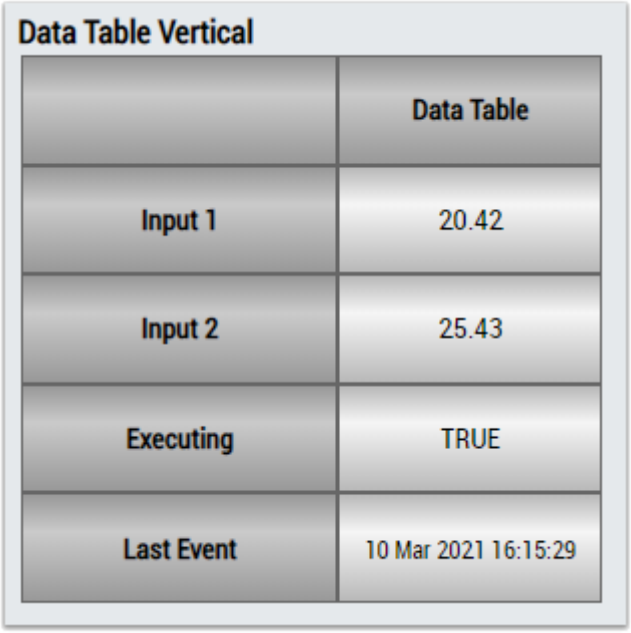
Standard HMI Controls

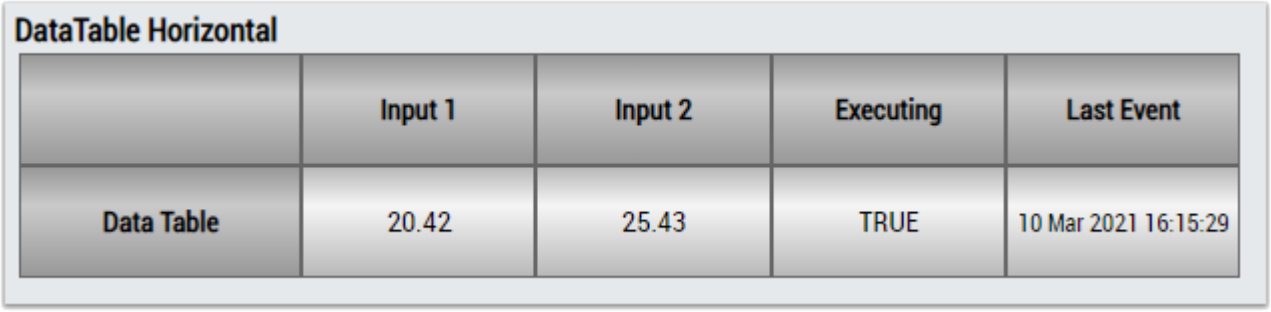
Für den Algorithmus Lifecycle Analysis 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Process Control visualisiert die Ausgangswerte Elapsed Cycles und Remaining Cycles.



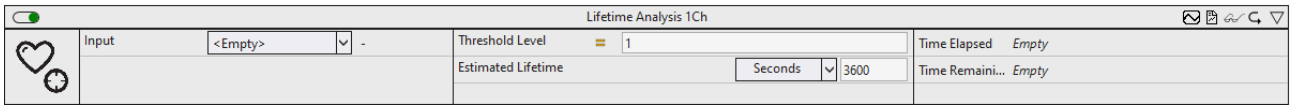
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Elapsed Cycles, Remaining Cycles.





Alternativ können im Algorithmus Lifecycle Analysis 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.14 Lifetime Analysis 1Ch



Der *Lifetime Analysis 1Ch* berechnet die abgelaufene und die geschätzte verbleibende Lebensdauer eines Geräts. Wenn der Eingangswert die konfigurierte Bedingung erfüllt, wird die Lebensdauer verringert.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Operator:** Gibt an, ob der Eingangswert größer, größer oder gleich, gleich, kleiner oder gleich, kleiner oder ungleich der Schwelle sein soll.
- **Threshold:** Schwelle des Signals.
- **Estimated Lifetime:** Geschätzte Lebensdauer des jeweiligen Geräts.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

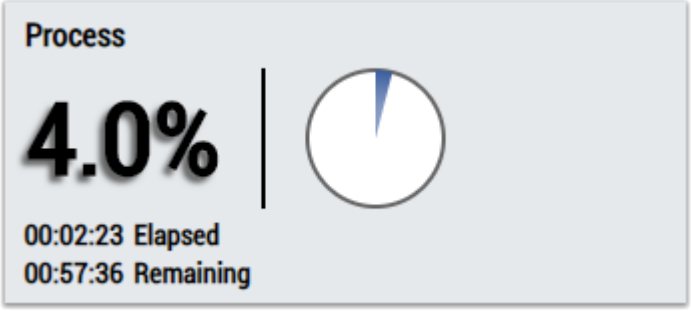
Ausgangswerte

- **Elapsed Lifetime:** Zeigt die Lebensdauer an, die bereits abgelaufen ist.
- **Remaining Lifetime:** Zeigt die verbleibende Lebensdauer des Geräts als Differenz der geschätzten und abgelaufenen Lebensdauer an.

Standard HMI Controls

Für den Algorithmus Lifetime Analysis 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Process Control visualisiert die Ausgangswerte Elapsed Lifetime und Remaining Lifetime.



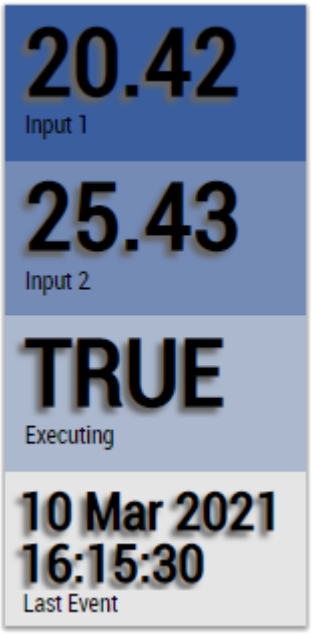
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Elapsed Lifetime, Remaining Lifetime.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

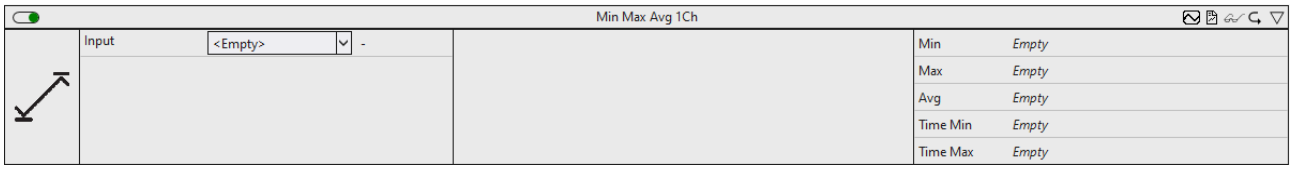
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Lifetime Analysis 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.15 Min Max Avg 1Ch



Der *Min Max Avg 1Ch* berechnet das Minimum, Maximum und den Durchschnitt der Eingangswerte vom Beginn der Analyse bis zum gegenwärtigen Zeitpunkt. Des Weiteren werden die Zeitwerte des Minimums und Maximums angezeigt.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

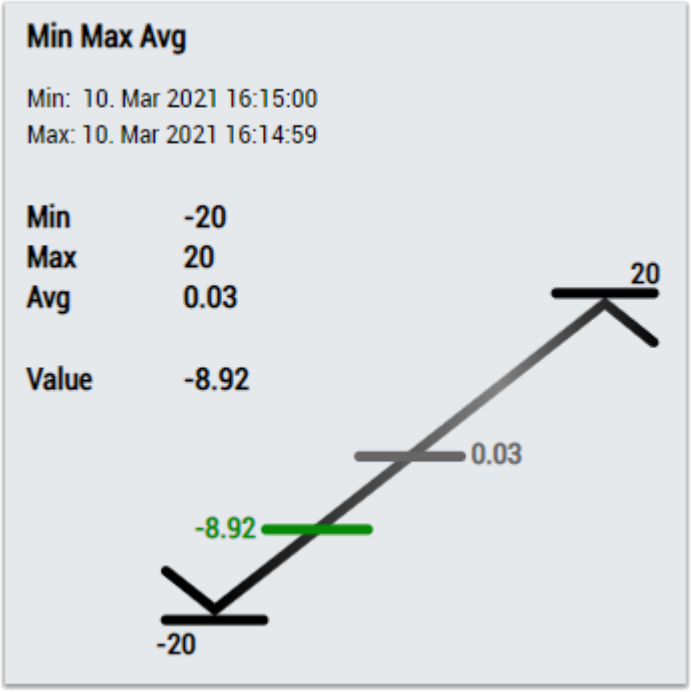
Ausgangswerte

- **Min:** Zeigt das Minimum der Eingangswerte an.
- **Max:** Zeigt das Maximum der Eingangswerte an.
- **Avg:** Zeigt den Durchschnitt der Eingangswerte an.
- **Time Min:** Zeigt den Zeitwert des Minimums an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Time Max:** Zeigt den Zeitwert des Maximums an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

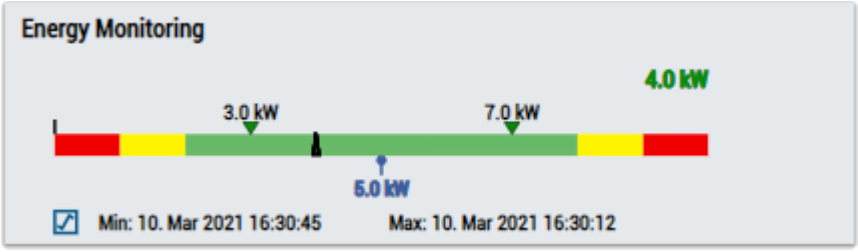
Standard HMI Controls

Für den Algorithmus Min Max Avg 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

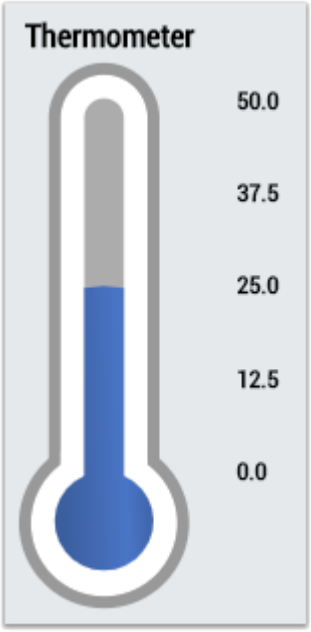
1. Das MinMaxAvg Control visualisiert die Ausgangswerte Min, Max, Avg, Time Min und Time Max sowie den Eingangswert der Daten.



2. Das EnergyMonitoring Control visualisiert alle Ausgangswerte: Min, Max, Avg, Time Min, Time Max und den aktuellen Inputwert.



3. Das Thermometer Control visualisiert die durchschnittliche (Avg) Temperatur.



4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Min, Max, Avg, Time Min, Time Max.

Data Table Vertical

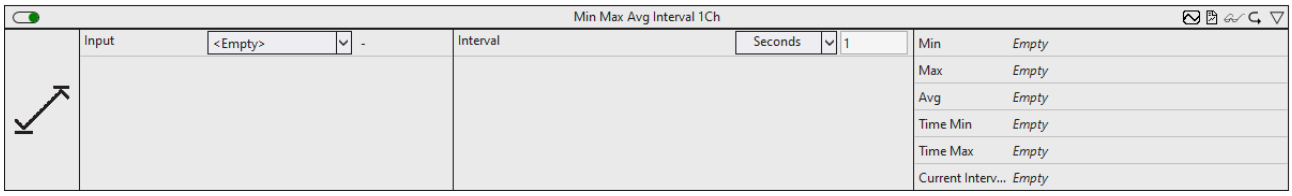
	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

Alternativ können im Algorithmus Min Max Avg 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.16 Min Max Avg Interval 1Ch



Der *Min Max Avg Interval 1Ch* berechnet das Minimum, Maximum und den Durchschnitt der Eingangswerte für den Zeitraum des konfigurierten Intervalls. Des Weiteren werden die Zeitwerte des Minimums und Maximums angezeigt. Zu beachten ist, dass alle Werte aus dem relativen letzten Intervall stammen und dass sie erst aktualisiert werden, wenn das Intervall vorüber ist. Die Berechnung wird neu gestartet, wenn die Zeit des Intervalls abgelaufen ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Sollen nur einzelne Samples innerhalb des konfigurierten Zeitsignals mit in die Berechnung der Ausgangswerte einfließen, so kann dies optional über den booleschen Eingang *Add Sample* realisiert werden. Es werden alle Eingangswerte berücksichtigt, bei denen der der Eingang *Add Sample* den Wert *TRUE* hat.

Konfigurationsoptionen

- **Interval:** Zeitintervall, in dem die Werte berechnet werden sollen.

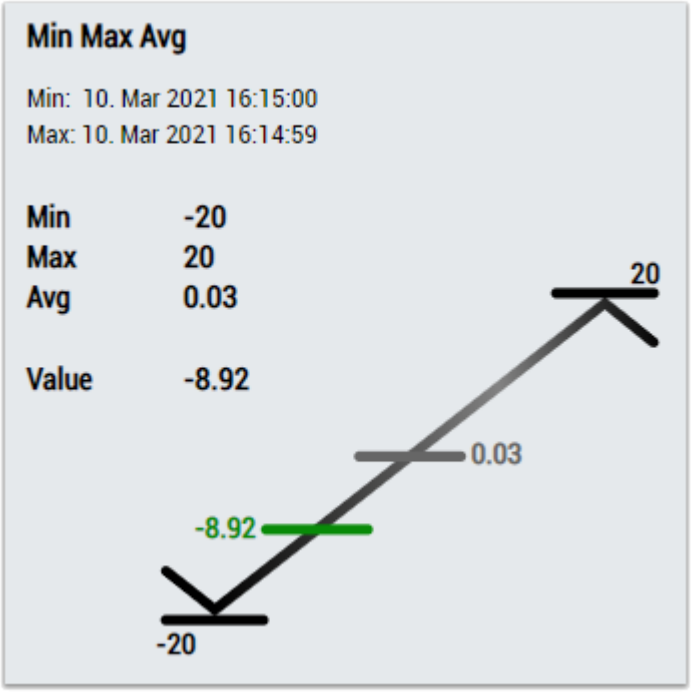
Ausgangswerte

- **Min:** Zeigt das Minimum der Eingangswerte im letzten Zeitintervall an.
- **Max:** Zeigt das Maximum der Eingangswerte im letzten Zeitintervall an.
- **Avg:** Zeigt den Durchschnitt der Eingangswerte im letzten Zeitintervall an.
- **Time Min:** Zeigt den Zeitwert des Minimums im letzten Zeitintervall an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Time Max:** Zeigt den Zeitwert des Maximums im letzten Zeitintervall an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Current Interval Time:** Zeigt die Zeitspanne an, die vom aktuellen Intervall bereits verstrichen ist.

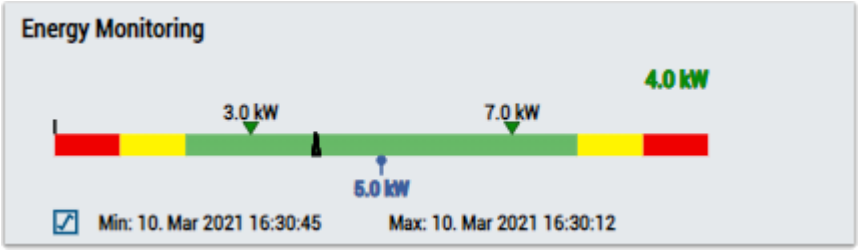
Standard HMI Controls

Für den Algorithmus Min Max Avg Interval 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

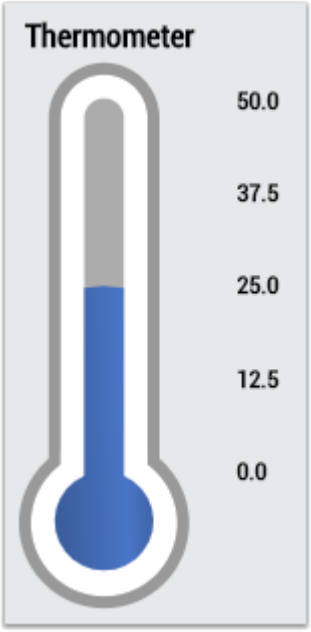
1. Das MinMaxAvg Control visualisiert die Ausgangswerte Min, Max, Avg, Time Min und Time Max sowie den Eingangswert der Daten.



2. Das EnergyMonitoring Control visualisiert die Ausgangswerte: Min, Max, Avg, Time Min, Time Max und den aktuellen Inputwert.



3. Das Thermometer Control visualisiert die durchschnittliche (Avg) Temperatur.



4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Min, Max, Avg, Time Min, Time Max, Current Interval Time.

Data Table Vertical

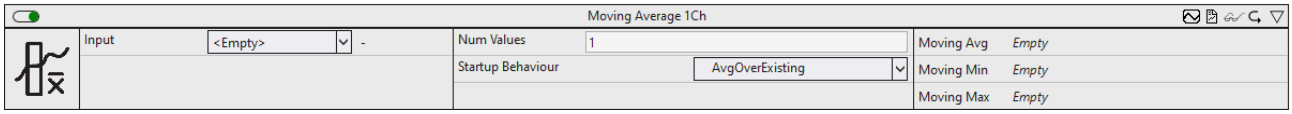
	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

Alternativ können im Algorithmus Min Max Avg Interval 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.17 Moving Average 1Ch



Der *Moving Average 1Ch* berechnet den gleitenden Durchschnitt, das Minimum und das Maximum der neuesten Eingangswerte in einem Intervall einer bestimmten Länge. Des Weiteren werden die Zeitwerte des Minimums und Maximums angezeigt. Die Berechnung des gleitenden Durchschnitts hängt von der Konfiguration der Parameter *Num Values* und *Startup Behaviour* ab.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Values:** Anzahl der Werte, die in die Berechnung des gleitenden Durchschnitts, des Minimums und des Maximums einfließen.
- **Startup Behaviour:** Berechnungsverhalten am Anfang der Analyse, bevor mindestens *Num Values* Eingangswerte vorhanden sind.

  ZeroPadding: Die fehlenden Werte werden mit Nullen gefüllt.

  UseFirstValue: Der erste Wert wird verwendet, bis die Anzahl der Werte *Num Values* entspricht.

  WaitUntilFilled: Das erste Ergebnis wird berechnet, wenn die Anzahl der Werte *Num Values* entspricht.

  AvgOverExisting: Der Durchschnitt wird mit den bereits vorhandenen Werten berechnet, bis die Anzahl der Werte *Num Values* entspricht.

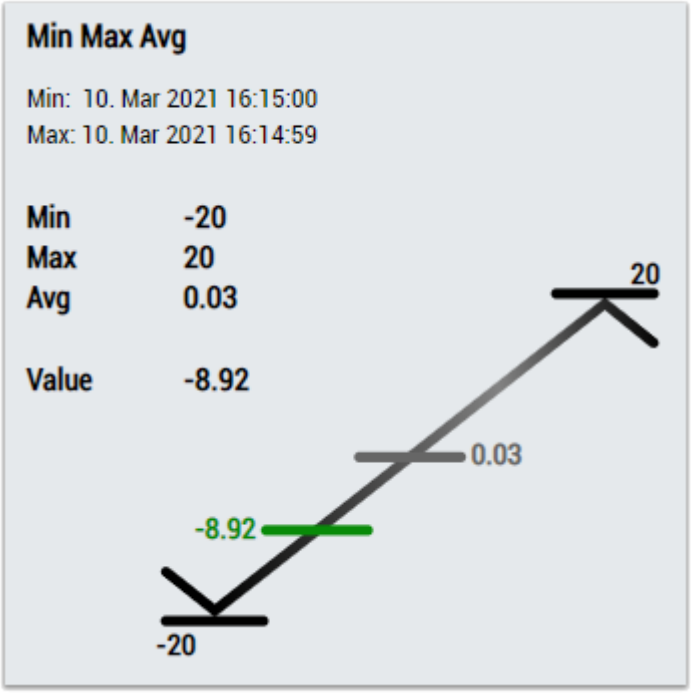
Ausgangswerte

- **Moving Avg:** Zeigt den aktuellen Mittelwert an.
- **Moving Min:** Zeigt das Minimum der letzten n Eingangswerte an.
- **Moving Max:** Zeigt das Maximum der letzten n Eingangswerte an.

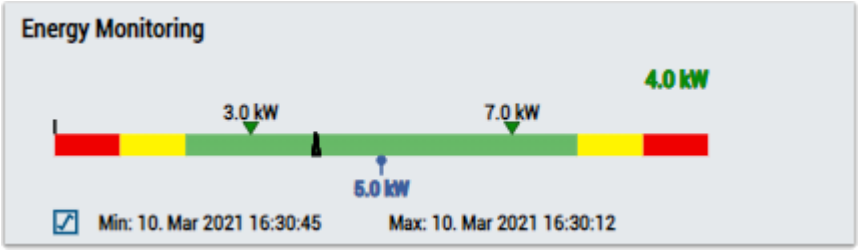
Standard HMI Controls

Für den Algorithmus Moving Average 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

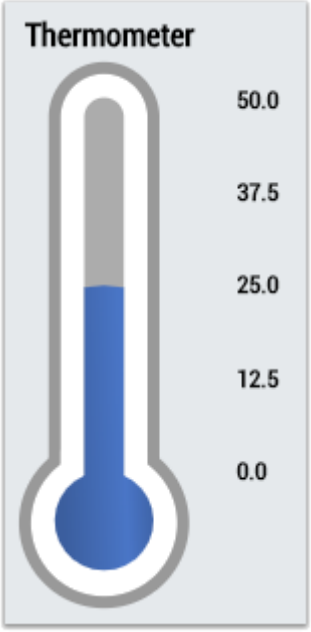
1. Das MinMaxAvg Control visualisiert die Ausgangswerte Moving Min, Moving Max und Moving Avg sowie den Eingangswert der Daten.



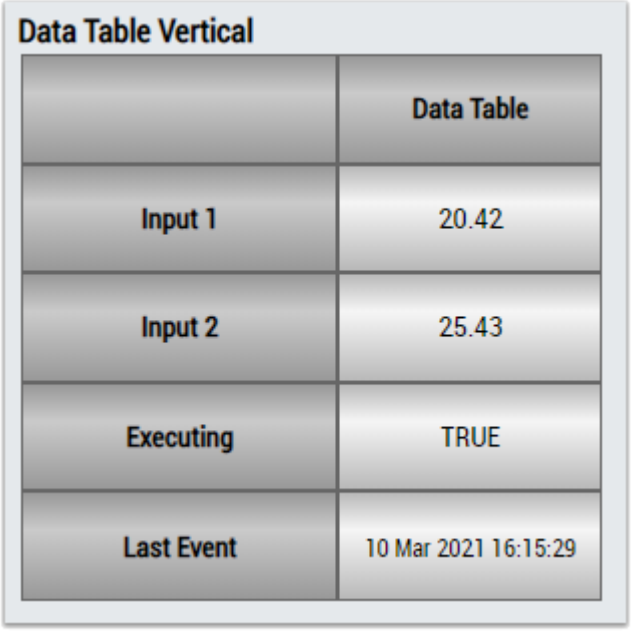
2. Das EnergyMonitoring Control visualisiert die Ausgangswerte Moving Min, Moving Max und Moving Avg sowie den Eingangswert der Daten.



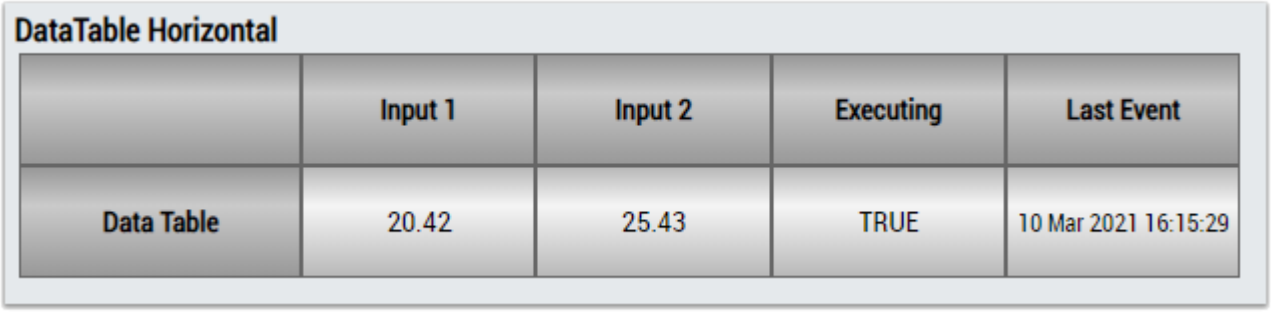
3. Das Thermometer Control visualisiert die durchschnittliche (Moving Avg) Temperatur.



4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Moving Min, Moving Max, Moving Avg.

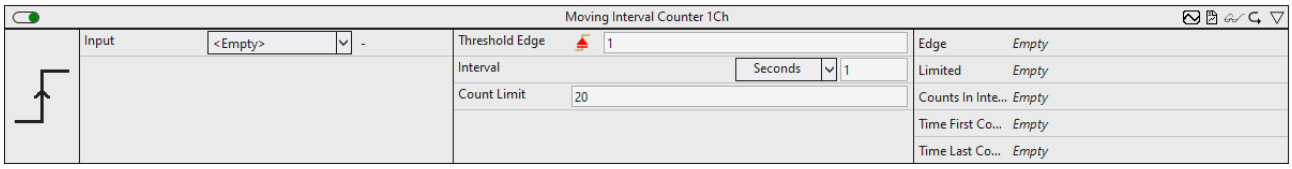


	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Moving Average 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.18 Moving Interval Counter 1Ch



Der *Moving Interval Counter 1Ch* zählt die Anzahl der ausgelösten Ereignisse innerhalb eines konfigurierten Intervalls. Ein Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten Schwelle passiert. Die Berechnung wird neu gestartet, wenn die Zeit des Intervalls abgelaufen ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Ereignis ausgelöst.
- **Interval:** Zeitintervall, in dem die Werte berechnet werden sollen.
- **Count Limit:** Begrenzt die Anzahl der Flanken, die in einem Intervall gezählt werden können.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

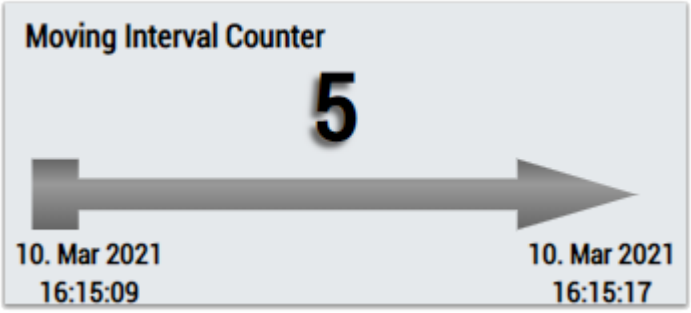
Ausgangswerte

- **Edge:** Zeigt zu dem Zeitpunkt, zu dem das Ereignis ausgelöst wird, *TRUE* an, ansonsten *FALSE*.
- **Limited:** Zeigt TRUE an, wenn die Anzahl der Flanken im aktuellen Intervall das eingestellte Count Limit überschreitet.
- **Counts in Interval:** Zeigt die Anzahl der ausgelösten Ereignisse im aktuellen Intervall an.
- **Time First Count:** Gibt den Zeitpunkt des ersten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Time Last Count:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

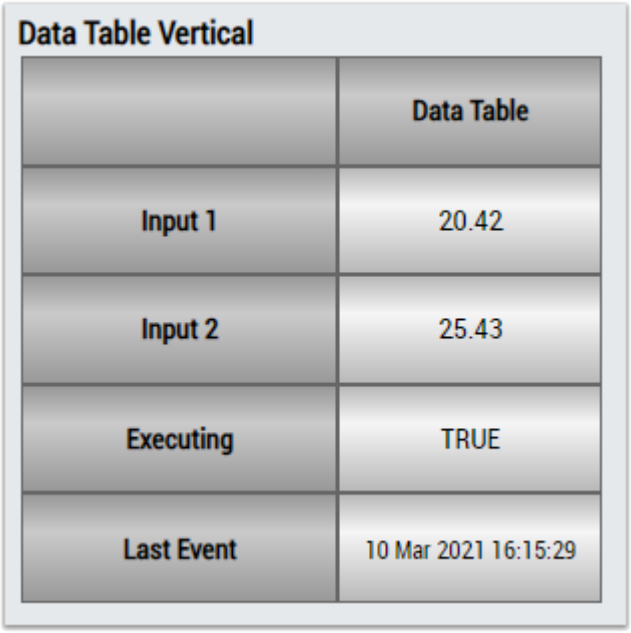
Standard HMI Controls

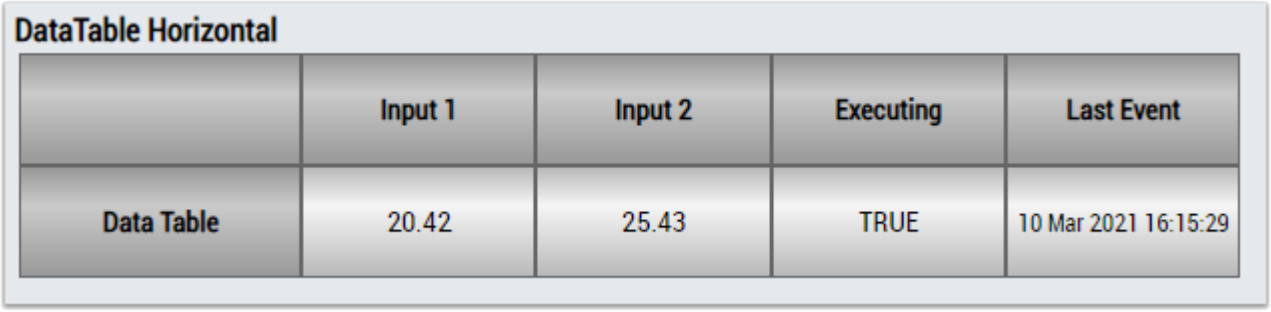
Für den Algorithmus Moving Interval Counter 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das MovingIntervalCounter Control visualisiert die Ausgangswerte Counts in Interval, Time First Count und Time Last Count.



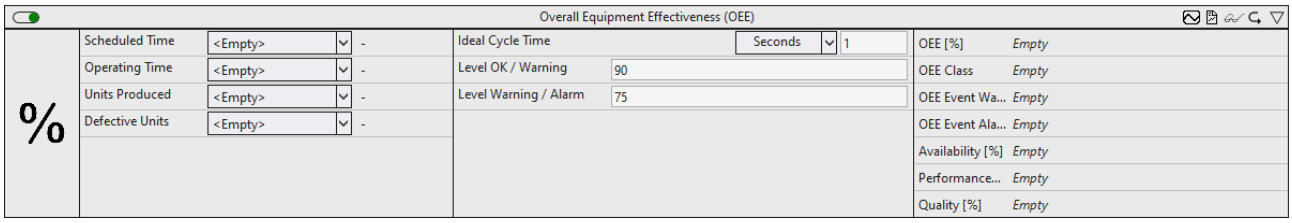
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Edge, Counts in Interval, Time First Count, Time Last Count, Limited.





Alternativ können im Algorithmus Moving Interval Counter 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.19 Overall Equipment Effectiveness (OEE)



Der *Overall Equipment Effectiveness (OEE)* berechnet Kennzahlen, die es ermöglichen, den aktuellen Zustand des Herstellungsprozesses mit seinem maximalen Potential zu vergleichen.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Ideal Cycle Time:** Ideale Zykluszeit für die Herstellung einer Einheit.
- **Level Ok / Warning:** Die Gesamtanlageneffektivität, die größter als die konfigurierte Schwelle ist, wird in die Klasse *OK* eingestuft. Ist die Gesamtanlageneffektivität kleiner oder gleich der konfigurierten Schwelle, wird sie in die Klasse *Warning* eingestuft.
- **Level Warning / Alarm:** Die Gesamtanlageneffektivität, die größer als die konfigurierte Schwelle ist, wird in die Klasse *Warning* eingestuft. Ist die Gesamtanlageneffektivität kleiner oder gleich der konfigurierten Schwelle, wird sie in die Klasse *Alarm* eingestuft.

Eingangswerte

- **Scheduled Time:** Die Betriebszeit errechnet sich aus der Kalenderzeit abzüglich der geplanten Nichtproduktion.
- **Operating Time:** Die Laufzeit errechnet sich aus der Betriebszeit abzüglich der Stillstandszeiten.
- **Units Produced:** Entspricht der Anzahl Produzierter Einheiten inklusive defekter Einheiten.
- **Defective Units:** Entspricht der Anzahl defekter Einheiten.

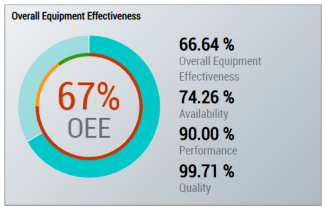
Ausgangswerte

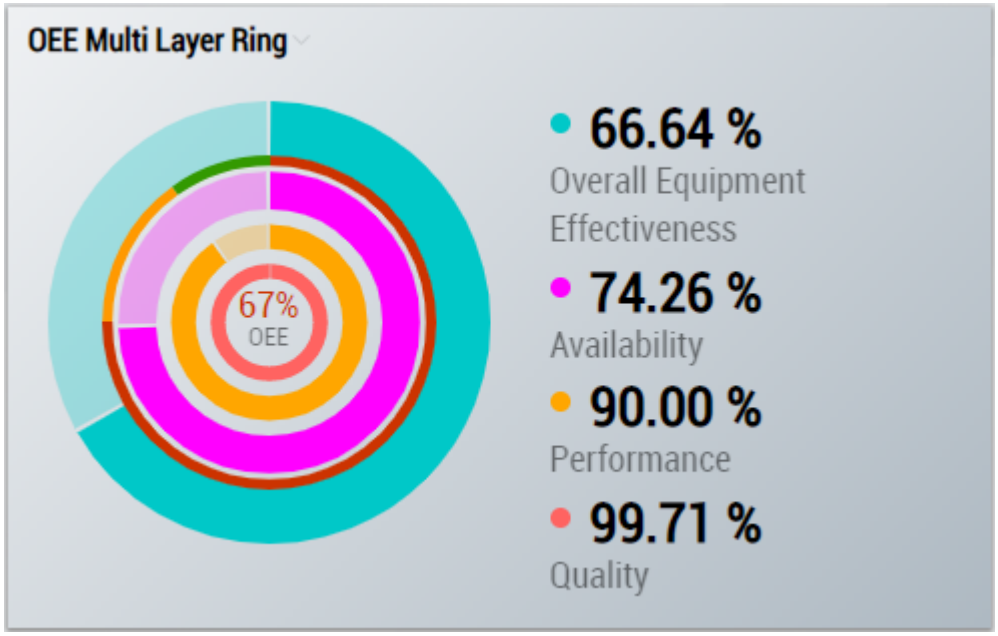
- **OEE:** Gesamtanlageneffektivität in Prozent. Sie berechnet sich aus der Multiplikation des Verfügbarkeitsfaktors, des Leistungsfaktors sowie des Qualitätsfaktors.
- **OEE Class:** Klassifizierung der Gesamtanlageneffektivität.
- **OEE Event Warning:** Gibt den Zeitpunkt des letzten ausgelösten Warnungsereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **OEE Event Alarm:** Gibt den Zeitpunkt des letzten ausgelösten Alarmereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Availability:** Verfügbarkeitsfaktor in Prozent. Sie berechnet sich aus dem Verhältnis zwischen der Laufzeit und der Betriebszeit.
- **Performance:** Leistungsfaktor in Prozent. Er berechnet sich aus dem Verhältnis tatsächlich produzierter Einheiten und der Anzahl produzierter Einheiten im Idealfall.
- **Quality:** Qualitätsfaktor in Prozent. Er berechnet sich aus dem Verhältnis intakter produzierter Einheiten im Verhältnis zu den produzierten Einheiten.

Standard HMI Controls

Für den Algorithmus *Overall Equipment Effectiveness (OEE)* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das OEE Control kommt mit drei Designs. Es visualisiert alle Ausgangswerte sowie die Parameter *Level OK / Warning* und *Level Warning / Alarm*.





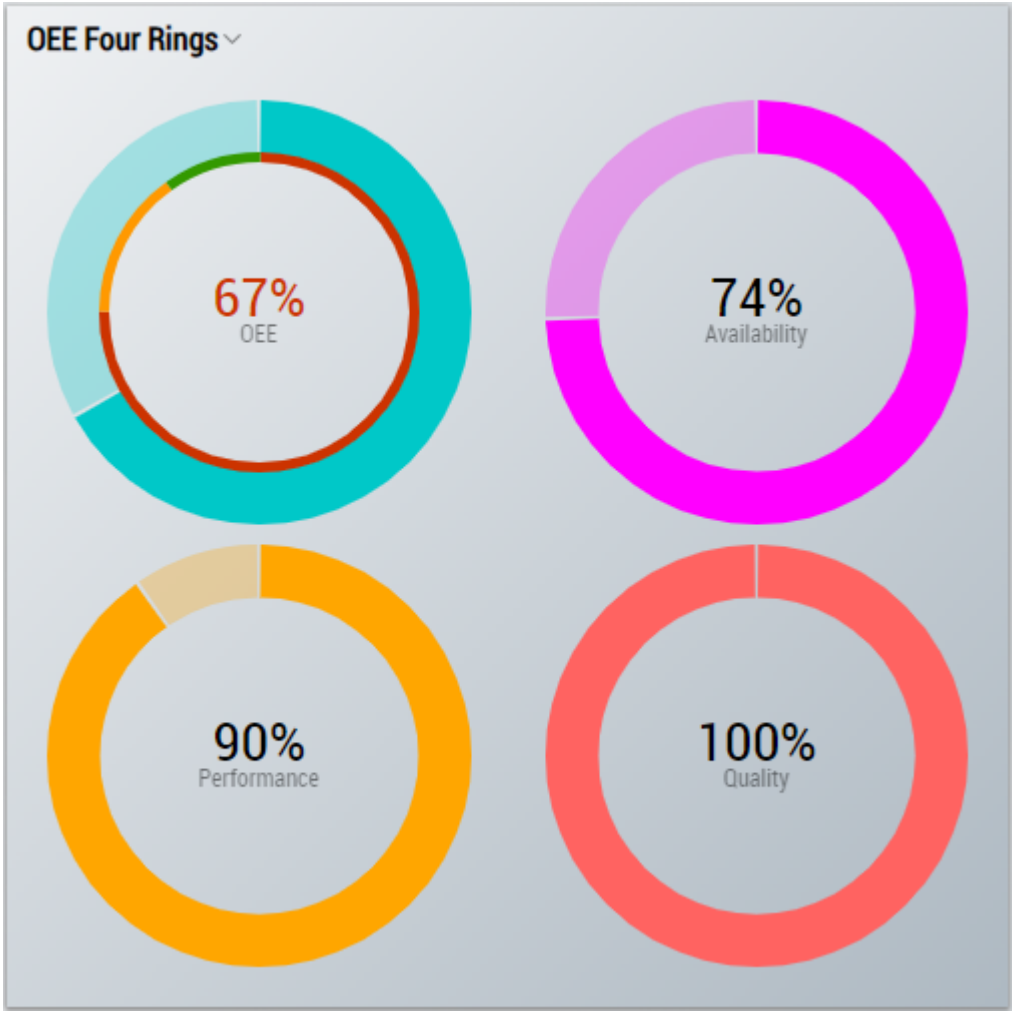
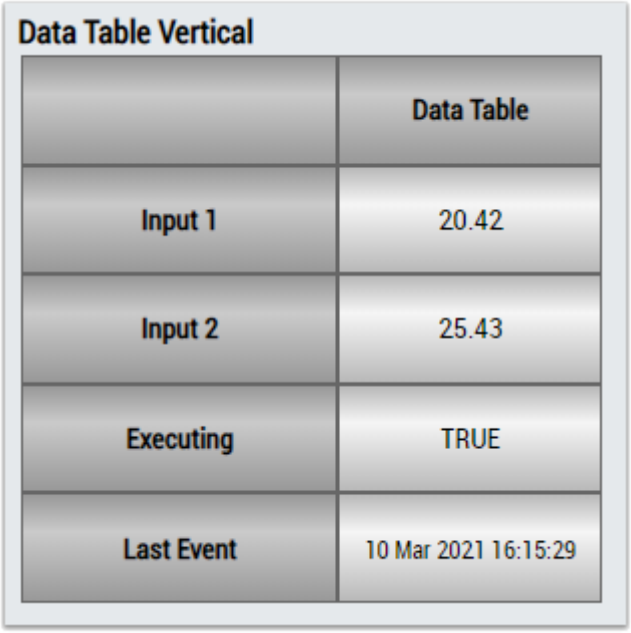
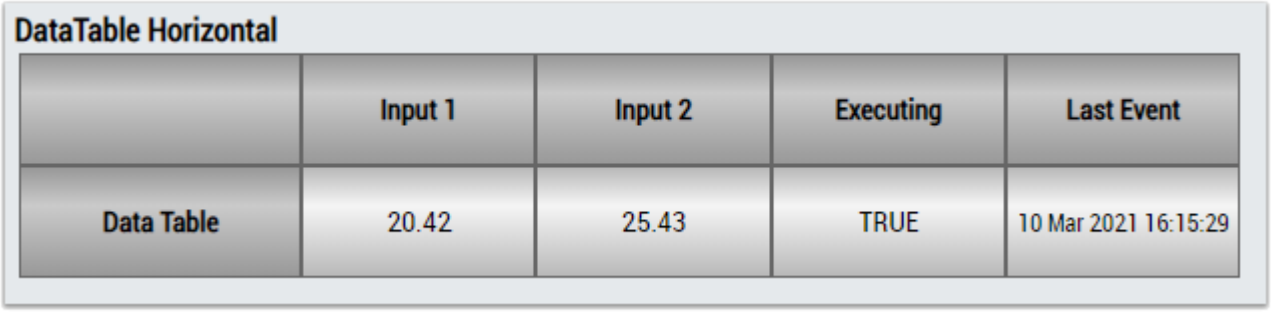


Abb. 1:

2. Das Table Control und Multivalue Control visualisieren die Ausgangswerte: *OEE, Availability, Performance und Quality*.

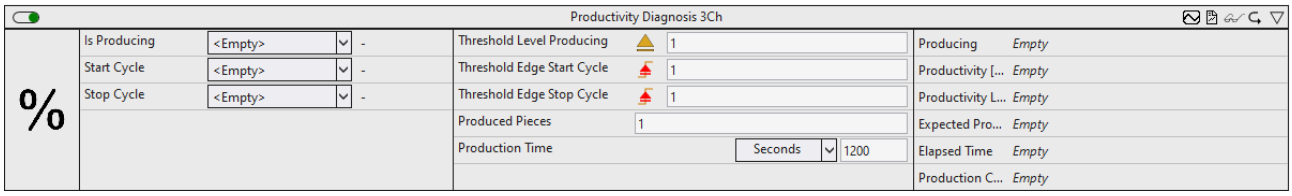


	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

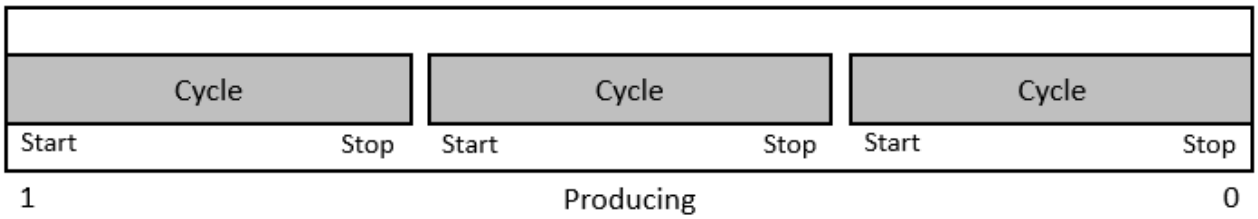


Alternativ können im Algorithmus *Overall Equipment Effectiveness (OEE)* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.20 Productivity Diagnosis 3Ch



Der Algorithmus *Productivity Diagnosis 3Ch* berechnet die Produktivität des Prozesses während eines Produktionsintervalls. Die folgende Abbildung illustriert schematisch den Zusammenhang zwischen dem Produktionsprozess und den einzelnen Produktionszyklen.



Das Produktionsintervall kann durch den Eingang *Is Producing* gestartet und gestoppt werden. Während der Ausführung des Produktionsintervalls werden die Produktionszyklen gezählt. Jeder Produktionszyklus entspricht einem produzierten Teil. Ein Produktionszyklus wird mit einer Flanke an *Start Cycle* gestartet und mit einer Flanke an *Stop Cycle* gestoppt. Die Produktivität über gesamte Produktionsintervall (*Productivity*) wird nach dem Stoppen des Intervalls, wenn das Signal *Is Producing* die Bedingung für *Threshold Level Producing* nicht mehr erfüllt, berechnet. Hierbei werden die vollständigen Produktionszyklen und demnach alle fertig gestellten Teile berücksichtigt. Die Produktivität ergibt sich als das Verhältnis der tatsächlich

produzierten Teile pro Zeit und des Sollwerts der in einer bestimmten Zeit zu produzierenden Teile. Der Ausgang *Productivity Last Cycle* errechnet sich aus der für den letzten Produktionszyklus benötigten Zeit bezogen auf die konfigurierte Zeit für ein Teil. Eventuelle Pausenzeiten zwischen den Zyklen werden dabei nicht berücksichtigt. Der Ausgang *Expected Productivity* schätzt die Gesamtproduktivität während des Produktionsintervalls. Dafür erfolgt eine Hochrechnung der bisherigen Produktionszeit auf die Gesamtproduktivität für den Sollwert der zu produzierenden Teile. Der Algorithmus kann mit dem Sollwert der produzierten Teile *(Produced Pieces)* innerhalb eines konfigurierten Intervalls *(Production Time)* konfiguriert werden, z. B. 1 Teil in 30 Sekunden oder 50 Teile pro Stunde.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the Edge:** Gibt an, ob der Algorithmus auf eine steigende oder eine fallende Flanke reagieren soll. Kann für jede Schwelle einzeln konfiguriert werden.
- **Threshold Level Producing**: Schwelle von *Input Producing* an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das *Production Time Interval* gestartet.
- **Threshold Edge Start Cycle:** Schwelle von *Input Start Cycle* an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird der Produktionszyklus gestartet.
- **Threshold Edge Stop Cycle:** Schwelle von *Input Stop Cycle* an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird der Produktionszyklus gestoppt.
- **Produced Pieces:** Sollwert der produzierten Teile während des konfigurierten Zeitintervalls (*Production Time*).
- **Production Time:** Zeitintervall der Produktionszeit. Es kann in Tagen, Stunden, Minuten oder Sekunden konfiguriert werden.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

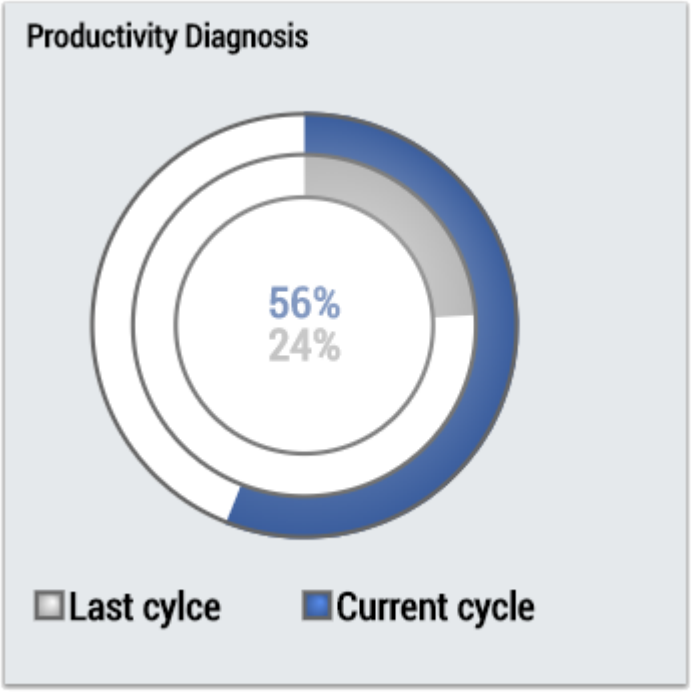
Ausgangswerte

- **Producing:** Gibt an, ob das Produktionsintervall aktiv ist.
- **Productivity:** Produktivität des gesamten Produktionsintervalls in Prozent.
- **Productivity Last Cycle:** Produktivität des letzten Produktionszyklus in Prozent.
- **Expected Productivity:** Schätzt die Produktivität des Produktionsintervalls. Angabe in Prozent.
- **Elapsed Time:** Zeitspanne seit dem Start des Produktionsintervalls.
- **Production Cycles:** Anzahl der vollständigen Produktionszyklen im aktuellen Produktionsintervall.

Standard HMI Controls

Für den Algorithmus Productivity Diagnosis 3Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das ProductivityDiagnosis Control visualisiert die Ausgangswerte Productivity und Productivity Last Cycle.



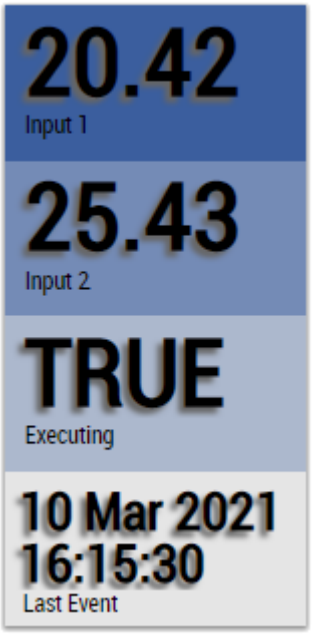
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: New Result, Producing, Cycle Finished, Productivity, Productivity Last Cycle.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

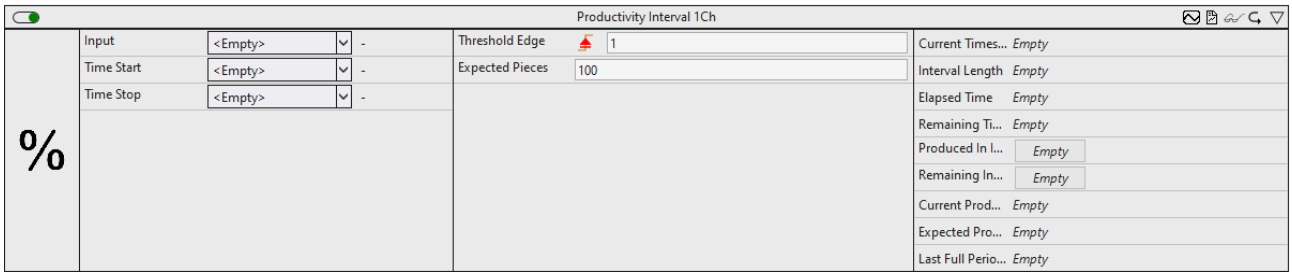
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Productivity Diagnosis 3Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.21 Productivity Interval 1Ch



Der Algorithmus *Productivity Interval 1Ch* berechnet die Produktivität des Prozesses während eines vorgegebenen Intervalls. Das Intervall kann durch die Eingänge *tTimeStart* und *tTimeStop* definiert werden. Bei der Ausführung werden die produzierten Teile berücksichtigt. Ein produziertes Element wird gezählt, wenn eine Flanke am Eingang anliegt. Als Ausgangswerte werden die geschätzte Produktivität des aktuellen Intervalls sowie die Produktivität des letzten vollständigen Intervalls bereitgestellt. Der Algorithmus kann mit dem Sollwert der produzierten Teile innerhalb eines vorgegebenen Intervalls konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Type of the Edge:** Gibt an, ob der Teilezähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold Edge**: Schwelle von *Input* an der ein gefertigtes Teil gezählt wird.
- **Expected Pieces:** Vorgabe der zu produzierenden Teile innerhalb der definierten Zeitspanne.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Eingangswerte

- **Input:** Eingangswert zum Zählen der produzierten Stücke.
- **Time Start:** Startzeitpunkt des vorgegebenen Intervalls. Ermöglicht die Vorgabe des Intervalls aus Prozessdaten. Angabe als Tageszeit in einer Auflösung von 1 ns.
- **Time Stop:** Endzeitpunkt des vorgegebenen Intervalls. Ermöglicht die Vorgabe des Intervalls aus Prozessdaten. Angabe als Tageszeit in einer Auflösung von 1 ns.

Ausgangswerte

- **Within Interval:** Gibt an, ob sich die aktuelle Uhrzeit innerhalb des Intervalls befindet.
- **Current Timestamp:** Aktueller Zeitstempel.
- **Interval Length:** Länge des Intervalls.
- **Elapsed Time:** Verstrichene Zeit innerhalb des Intervalls.
- **Remaining Time:** Verbleibende Zeit innerhalb des Intervalls.
- **Produced In Interval:** Produzierte Teile innerhalb des Intervalls.
- **Remaining In Interval:** Verbleibende Teile innerhalb des Intervalls.
- **Current Productivity:** Aktuelle Produktivität des Intervalls in Prozent. Berücksichtigt die Länge des Intervalls, die bereits verstrichene Zeit, die zu produzierenden Teile sowie die bereits produzierten Teile. Die Ausgabe erfolgt in Prozent.
- **Expected Productivity:** Geschätzte Produktivität des Intervalls in Prozent. Zur Ermittlung der möglichen produzierbaren Teile innerhalb der verbleibenden Zeit wird die Produktionszeit des letzten Teils verwendet.
- **Last Full Period Productivity:** Produktivität des letzten komplett durchlaufenden Intervalls in Prozent. Wird nur berechnet, wenn das Intervall vollständig durchlaufen wurde.

Standard HMI Controls

Für den Algorithmus Productivity Interval 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das ProductivityInterval Control visualisiert die Ausgangswerte: Vergangene Zeit, verbleibende Zeit, Anzahl produzierter Teile im Intervall, Anzahl verbleibender Teile im Interval, Produktivität.

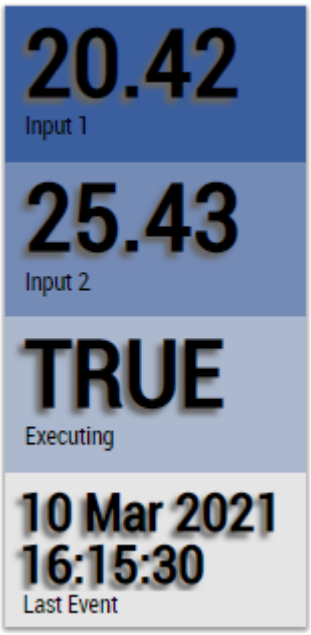
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Aktuelle Zeit, Intervalllänge, Vergangene Zeit, verbleibende Zeit, Anzahl produzierter Teile im Intervall, Anzahl verbleibender Teile im Interval, Produktivität.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

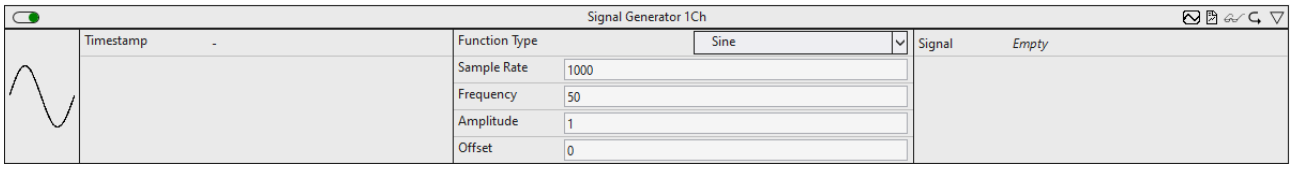
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Productivity Interval 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.22 Signal Generator 1Ch



Der *Signal Generator 1Ch* kann zur Erzeugung verschiedener Signalverläufe genutzt werden. Die Signalform, die Frequenz, die Amplitude sowie der Offset können individuell eingestellt werden. Als Referenzwert ist ein Zeitstempel erforderlich, da der Algorithmus einen Zeitkontext benötigt in dem er arbeiten soll. Dieser Referenz-Zeitstempel wird automatisch gesetzt, wenn die Konfiguration einen weiteren Algorithmus umfasst. Daher ist es nicht möglich, den Signal Generator 1Ch einzeln zu verwenden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Amplitude:** Konfiguration der Signalamplitude.
- **Frequency:** Frequenz des generierten Signals.
- **Function Type:** Funktionstyp des generierten Signals.
  Const: Konstant
  Rectangle: Rechteckfunktion
  Sawtooth: Sägezahnfunktion
  Sine: Sinusfunktion
  Triangle: Dreiecksfunktion
- **Offset:** Konstanter Versatz des generierten Signals.
- **Sample Rate:** Sample Rate des zu analysierenden Systems.

Ausgangswerte

- **Signal:** Gibt das generierte Signal aus.

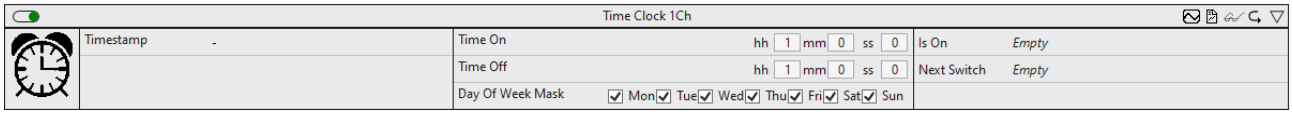
Standard HMI Controls

Für den Signal Generator 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SingleValue Control visualisiert den Ausgangswert Signal.

Alternativ können im Algorithmus Signal Generator 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.23 Time Clock 1Ch



Der *Time Clock 1Ch* führt eine Zeitschaltung aus, die mit Einschaltzeit, Ausschaltzeit und den Tagen der Woche, an denen die Zeitschaltung aktiv sein soll, konfiguriert werden kann. Als Referenzwert ist ein Zeitstempel erforderlich, da der Algorithmus einen Zeitkontext benötigt in dem er arbeiten soll. Dieser Referenz-Zeitstempel wird automatisch gesetzt, wenn die Konfiguration einen weiteren Algorithmus umfasst. Daher ist es nicht möglich, den Time Clock 1Ch einzeln zu verwenden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Time On:** Einschaltzeit.
- **Time Off:** Ausschaltzeit.
- **Day of Week Mask:** Wochentage, an denen die Zeitschaltung aktiv sein soll.

Ausgangswerte

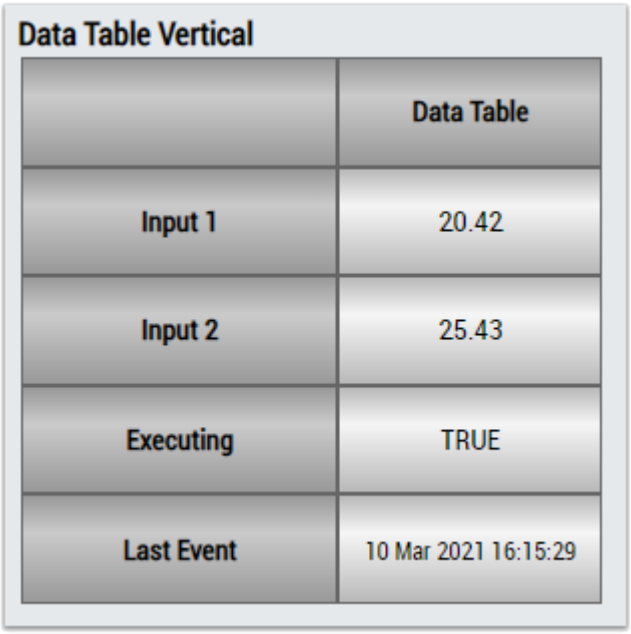
- **Is On:** Zeigt *TRUE* an, wenn die Zeitschaltung eingeschaltet ist, ansonsten *FALSE*.
- **Next Switch:** Zeigt die verbleibende Zeit bis zur nächsten Schaltung an.

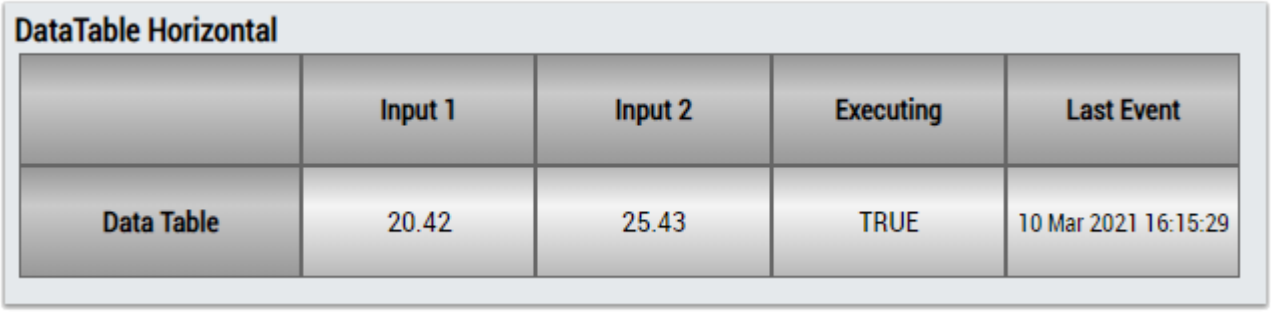
Standard HMI Controls

Für den Algorithmus Time Clock 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Timer Control visualisiert den Ausgangswert Next Switch.

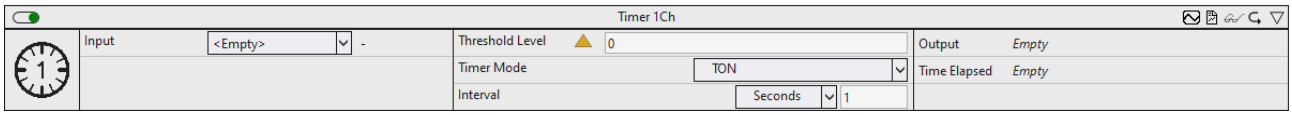
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Output, Next Switch.





Alternativ können im Algorithmus Time Clock 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.24 Timer 1Ch



Der *Timer 1Ch* startet einen Timer, der nach Timermodus und Intervall konfiguriert werden kann. Gemäß dem spezifischen Timermodus wird der Timer gestartet, wenn die konfigurierte Bedingung *TRUE* (TON, TP) oder *FALSE* (TOF) wird.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Operator:** Gibt an, ob der Eingangswert größer, größer oder gleich, gleich, kleiner oder gleich oder kleiner als die Schwelle sein soll.
- **Threshold:** Schwelle des Signals.
- **Timer Mode:** Modus des Timers:

  TON: Der TON-Timer ist ein Einschaltverzögerungstimer, der den Ausgang aktiviert, nachdem die Schwellenbedingung *TRUE* wird und die im Intervall festgelegte Zeitspanne abgelaufen ist.

  TOF: Der TOF-Timer ist ein Ausschaltverzögerungstimer, der den Ausgang deaktiviert, nachdem die Schwellenbedingung *FALSE* wird und die im Intervall festgelegte Zeitspanne abgelaufen ist.

  TP: Der TP-Timer ist ein Impulsgeber, der den Ausgang für die im Intervall festgelegte Zeit aktiviert, nachdem die Schwellenbedingung *TRUE* wird.
- **Interval:** Zeitintervall des konfigurierten Timers.

- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

- **Output:** Zeigt den Ausgangswert an, der von dem konfigurierten Timer betroffen ist.
- **Elapsed Time:** Zeigt die abgelaufene Zeit des Timers an.

Standard HMI Controls

Für den Algorithmus Timer 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Timer Control visualisiert den Ausgangswert Elapsed Time.

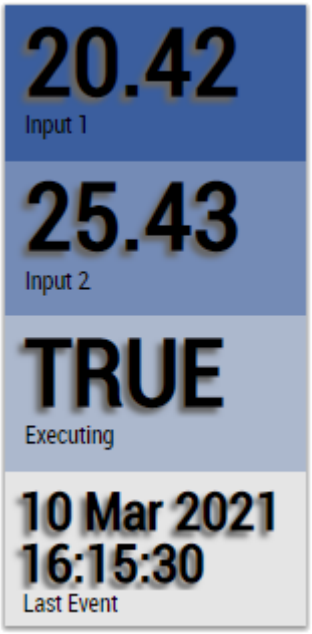
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Output, Elapsed Time.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

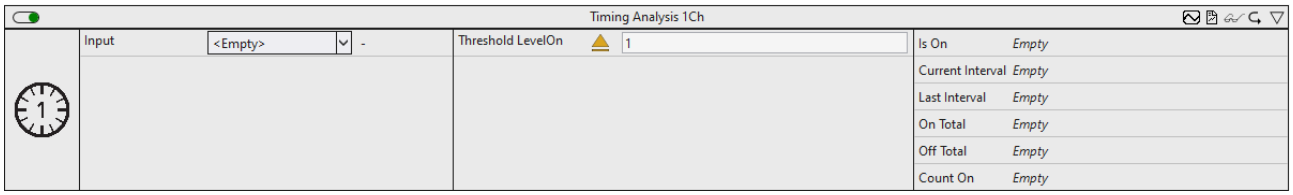
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Timer 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.1.25 Timing Analysis 1Ch



Der *Timing Analysis 1Ch* misst Zeitdifferenzen zwischen On- und Off-Perioden und zählt die Anzahl der On-Perioden. Die On-Periode beginnt, wenn die Bedingung von Operator und Schwelle erfüllt ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Operator:** Gibt an, ob der Eingangswert größer, größer oder gleich, gleich, kleiner oder gleich, kleiner oder ungleich der Schwelle sein soll.
- **Threshold:** Schwelle des Signals.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

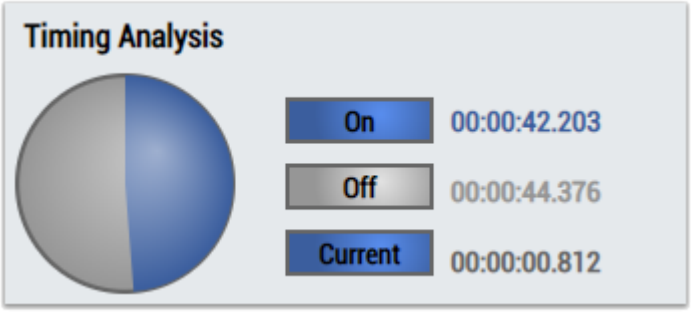
Ausgangswerte

- **Is On:** Zeigt innerhalb der Zeitspanne der On-Periode *TRUE* an, ansonsten *FALSE*.
- **Current Interval:** Zeigt die Zeit des aktuellen Intervalls an.
- **Last Interval:** Zeitspanne des zuletzt abgeschlossenen Intervalls.
- **On Total:** Zeigt die Gesamtzeit an, für die der „Is On“-Wert *TRUE* ist.
- **Off Total:** Zeigt die Gesamtzeit an, für die der „Is On“-Wert *FALSE* ist.
- **Count On:** Zählt die Anzahl der ausgelösten On-Ereignisse.

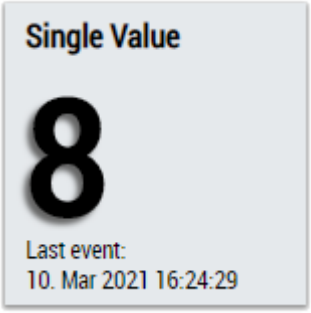
Standard HMI Controls

Für den Algorithmus Timing Analysis 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das TimingAnalysis Control visualisiert die Ausgangswerte Is On, On Total, Off Total und Current Interval.



2. Das SingleValue Control visualisiert den Ausgangswert Count On.



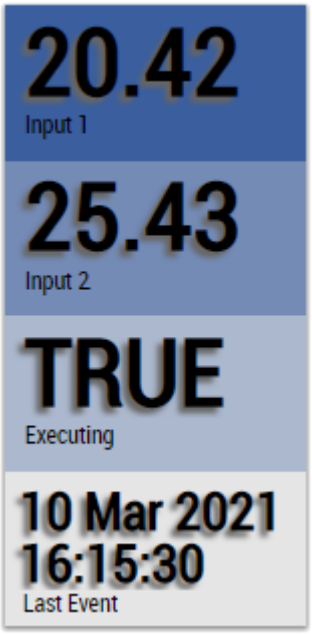
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Is On, Count On, Current Interval, On Total, Off Total.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

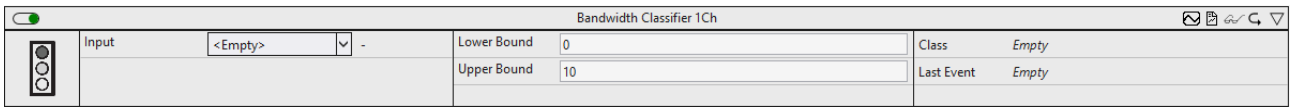


Alternativ können im Algorithmus Timing Analysis 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2 Analytics - Classification

Die Algorithmen der Kategorie *Analytics-Classification* bieten Funktionen für die Klassifizierung und Zustandserkennung.

6.4.2.1 Bandwidth Classifier 1Ch



Der *Bandwidth Classifier 1Ch* bestimmt, ob das Eingangssignal innerhalb der konfigurierten Grenzen liegt oder kleiner oder größer als die Grenzen ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Lower Bound:** Untere Grenze für den Vergleich.
- **Upper Bound:** Obere Grenze für den Vergleich.

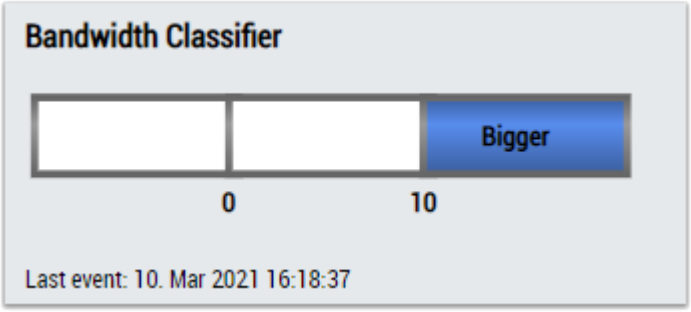
Ausgangswerte

- **Class:** Zeigt die Klasse an, zu der die Eingangswerte gehören (WithinBounds/Smaller/Bigger).
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

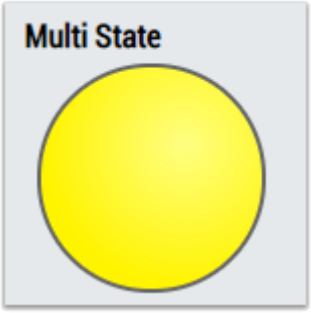
Standard HMI Controls

Für den Algorithmus Bandwidth Classifier 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

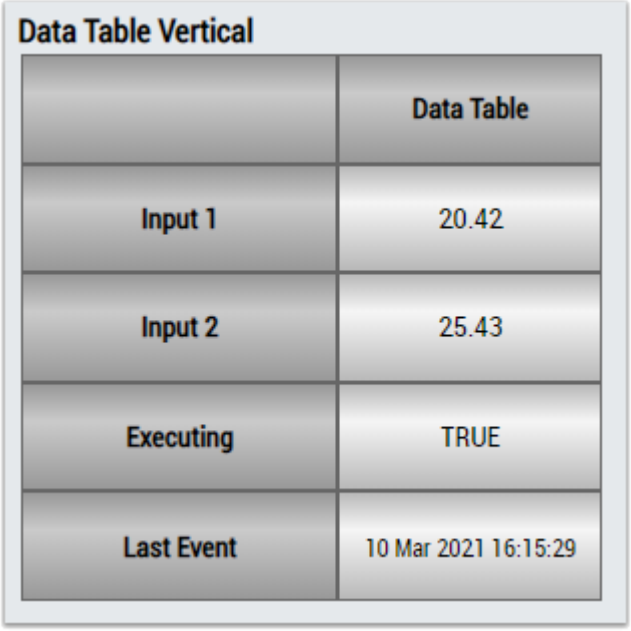
1. Das BandwidthClassifier Control visualisiert die Ausgangswerte Class und Last Event und die Konfigurationsoptionen Lower Bound und Upper Bound.

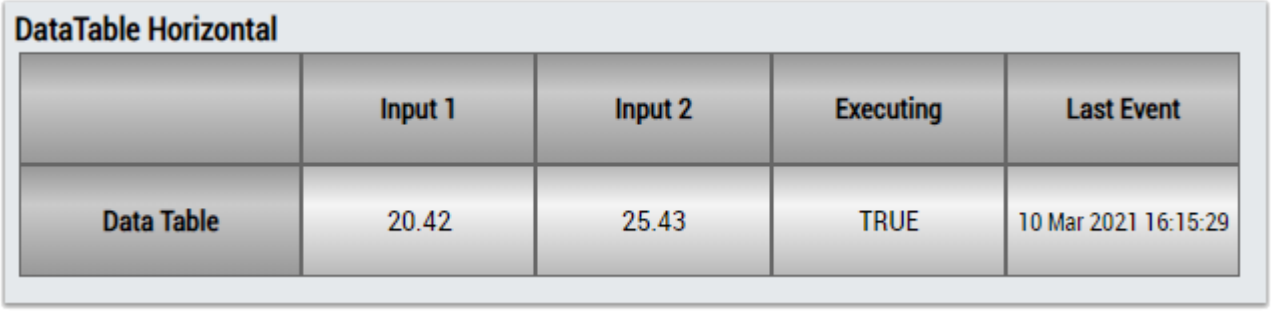


2. Das MultiState Control visualisiert den Ausgangswert Class.



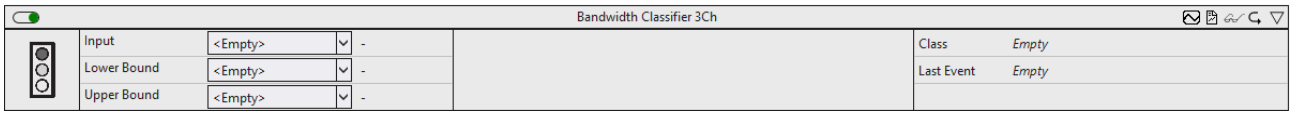
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Class, Last Event.





Alternativ können im Algorithmus Bandwidth Classifier 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.2 Bandwidth Classifier 3 Ch



Der *Bandwidth Classifier 3Ch* bestimmt, ob das Eingangssignal innerhalb der Grenzen liegt oder kleiner oder größer als die Grenzen ist. Die Grenzen können mit Eingangssignalen konfiguriert werden, sodass es möglich ist, Kurvenverläufe als unteres und oberes Band zu verwenden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

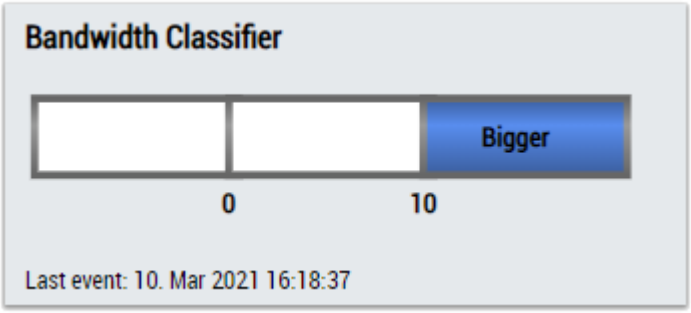
Ausgangswerte

- **Class:** Zeigt die Klasse an, zu der die Eingangswerte gehören (WithinBounds/Smaller/Bigger).
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

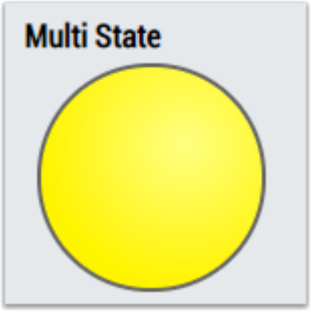
Standard HMI Controls

Für den Algorithmus Bandwidth Classifier 3Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das BandwidthClassifier Control visualisiert die Ausgangswerte Class und Last Event und Input Lower Bound und Input Upper Bound.



2. Das MultiState Control visualisiert den Ausgangswert Class.



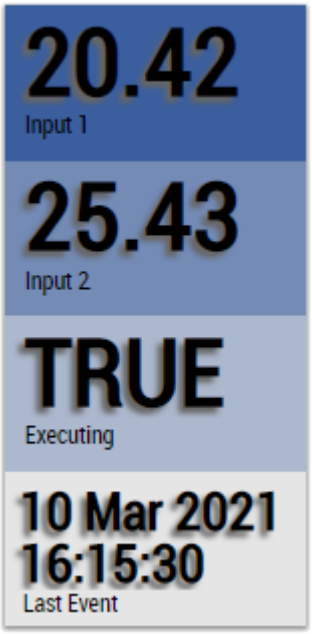
3. Das Table Control visualisiert alle Ausgangswerte: Class, Last Event.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

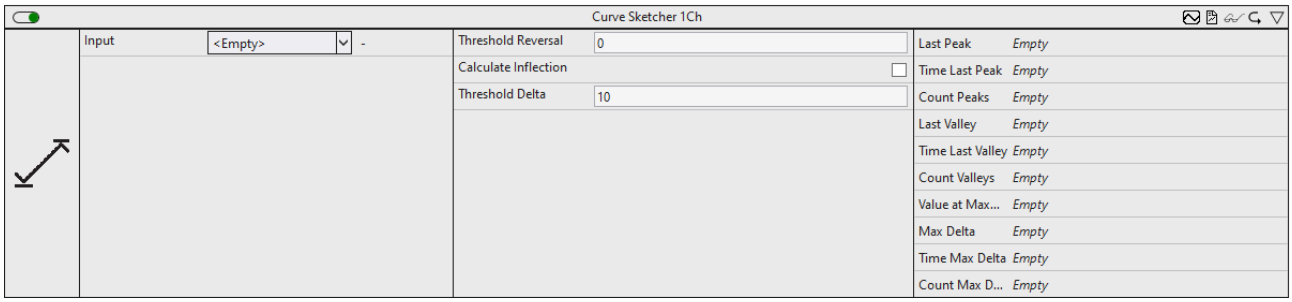
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Bandwidth Classifier 3Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.3 Curve Sketcher 1Ch



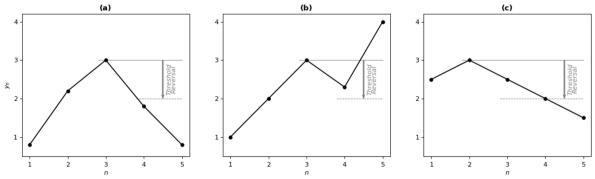
Curve Sketcher 1Ch identifiziert Umkehrungen (*Peaks* und *Täler*) in einem Eingangsdatenstrom. Des Weiteren können lokale Maxima der absoluten Differenz zwischen zwei aufeinanderfolgenden Werten (bezeichnet als *Delta*) identifiziert werden. Analog zu einer kontinuierlichen Kurve entsprechen die identifizierten Peaks und Täler lokalen Maxima und Minima. Das Delta entspricht der Steigung, sodass ein Maximum der absoluten Werte des Deltas mit einem Wendepunkt verbunden werden kann.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

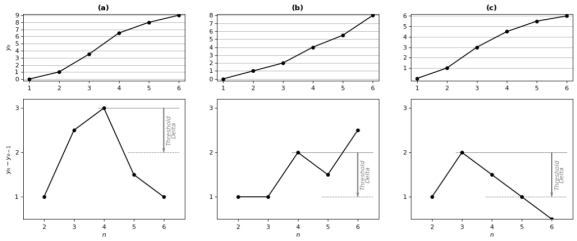
Konfigurationsoptionen

- **Calculate Inflection:** Boolsches Flag. Maxima der Änderungsrate werden nur identifiziert, wenn dieses Flag *True* ist. Anderenfalls werden die Werte für *Count Max Delta, Max Delta*, *Time Max Delta* und *Value at Max Delta* nicht berechnet.
- **Threshold Reversal:** Schwellenwert für die Identifizierung von Umkehrungen. Umkehrungen werden nur erkannt, wenn ihre Differenz zur nächsten Umkehrung den Wert von Threshold Reversal überschreitet.
  Nachfolgend sind drei Beispiele für die Identifizierung von Peaks mit dem Parameter *Threshold Reversal* aufgeführt.
  (a) Der Wert y_3 wird direkt nach der Verarbeitung des Werts y_4 als Peak identifiziert, da die Differenz zwischen y_3 und y_4 größer ist als *Threshold Reversal*.
  (b) Der Wert y_3 wird nicht als Peak identifiziert, da die Differenz zwischen y_3 und y_4 kleiner ist als *Threshold Reversal* und die Kurve nach y_4 wieder ansteigt.
  (c) Der Wert y_2 wird nach der Verarbeitung des Werts y_5 als Peak identifiziert, da die Differenz

zwischen y_2 und y_5 *Threshold Reversal* überschreitet. Der Wert y_2 kann vorher nicht als Peak identifiziert werden, da die Differenz zwischen y_2 und y_3 (y_4) kleiner als/gleich *Threshold Reversal* ist und nicht bekannt ist, ob die Werte weiter sinken.



- **Threshold Delta:** Schwellenwert für die Identifizierung der Delta-Maxima. Maxima der absoluten Differenz von zwei aufeinanderfolgenden Werten (Delta) werden nur erkannt, wenn die Differenz zwischen aufeinanderfolgenden Deltas *Threshold Delta* überschreitet.
  Nachfolgend sind drei Beispiele für die Identifizierung der Delta-Maxima mit dem Parameter *Threshold Delta* aufgeführt. Die oberen Diagramme zeigen die ursprünglichen Eingangssignale, die unteren das zugehörige Delta.
  (a) Der Wert y_4 wird nach der Verarbeitung des Werts y_5 als Maximum identifiziert, da die Differenz der zwei Deltas *Threshold Delta* überschreitet.
  (b) Es wird kein Maximum identifiziert, da die Differenz zwischen den Deltas kleiner ist als *Threshold Delta*.
  (c) Der Wert y_3 wird nach der Verarbeitung des Werts y_6 als Maximum identifiziert.



Unabhängig von *Threshold Delta* wird mindestens ein Maximum des Deltas zwischen zwei Umkehrungen erkannt.

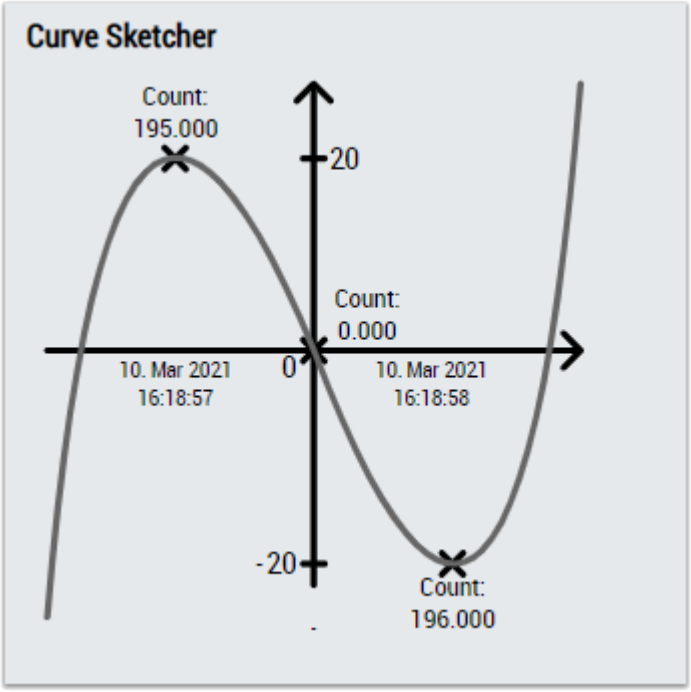
Ausgangswerte

- **Last Peak:** Gibt den y-Wert des zuletzt identifizierten Peaks an.
- **Time Last Peak:** Gibt den Zeitstempel des zuletzt identifizierten Peaks an.
- **Count Peaks:** Gibt die Gesamtanzahl der gezählten Peaks an.
- **Last Valley:** Gibt den y-Wert des zuletzt identifizierten Tals an.
- **Time Last Valley:** Gibt den Zeitstempel des zuletzt identifizierten Tals an.
- **Count Valleys:** Gibt die Gesamtanzahl der gezählten Täler an.
- **Value at Max Delta:** Gibt den y-Wert des analysierten Streams (Eingangsvariable) an, der von dem zuletzt erkannten Maximum von Delta geleitet wird. Der Wert Delta ist die Differenz zwischen *Value at Max Delta* und dem Wert, der einen Zyklus zuvor erreicht wurde.
- **Max Delta:** Gibt das zuletzt erkannte lokale Maximum der absoluten Differenz zwischen zwei aufeinanderfolgenden Werten im Eingangsstrom an.
- **Time Max Delta:** Dies ist der Zeitstempel von *Value at Max Delta.*
- **Count Max Delta:** Gibt die Gesamtanzahl der gezählten lokalen Maxima von Delta an.

Standard HMI Controls

Für den Algorithmus Curve Sketcher 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das CurveSketcher Control visualisiert die Ausgangswerte Last Peak, Time Last Peak, Count Peaks, Last Valley, Time Last Valley, Count Valley, Last Delta, Time Last Delta und Count Delta sowie den Eingangswert der Daten.



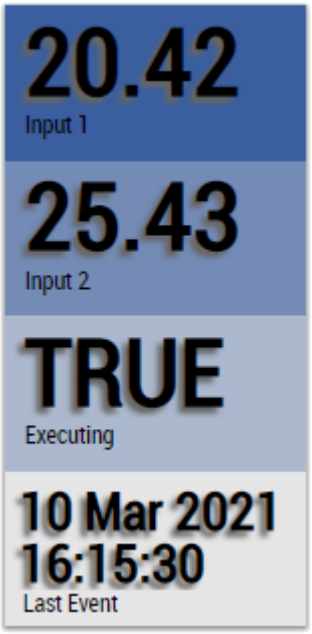
2. Das Table Control visualisiert alle Ausgangswerte: Last Peak, Time Last Peak, Count Peaks, Last Valley, Time Last Valley, Count Valley, Value at Max Delta, Last Delta, Time Last Delta und Count Delta.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

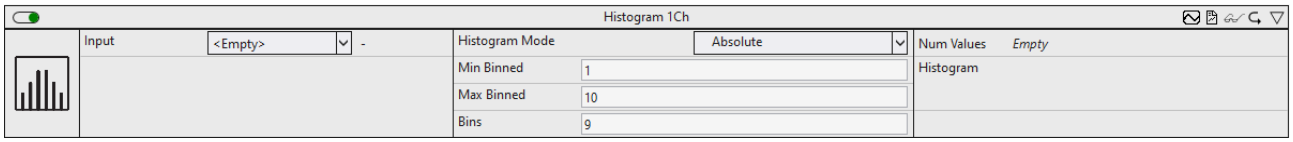
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Curve Sketcher 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.4 Histogram 1Ch



Histogram 1Ch berechnet die Verteilung eines Einzelkanal-Eingangswerts zyklisch. Er kann mit minimaler Klasse, maximaler Klasse und Gesamtanzahl der Klassen konfiguriert werden. Die Dimension des Ausgangs-Arrays ist die Anzahl der Klassen + 2, da Werte, die kleiner als die minimale Klasse sind, im ersten Array-Element gespeichert werden und Werte, die größer als die maximale Klasse sind, im letzten Array-Element gespeichert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Histogram Mode:** Die Betriebsart des Histogramms. Es wird zwischen einer absoluten und einer relativen Ausgabe unterschieden. Letzteres kann verwendet werden, um die prozentuale Verteilung anzuzeigen.
- **Min Binned:** Minimaler Wert, der analysiert werden soll.
- **Max Binned:** Maximaler Wert, der analysiert werden soll.
- **Bins:** Gesamtanzahl der Klassen für das Histogramm, die berechnet werden sollen.
- **Enable Display Names**: Aktivierung von Histogram Display Names. Diese Konfiguration ist versteckt in dem Property Fenster sichtbar.
- **Histogram Display Names:** Displaynamen der einzelnen Bins. Diese Konfiguration ist versteckt in dem Property Fenster sichtbar und wird für das HMI verwendet.

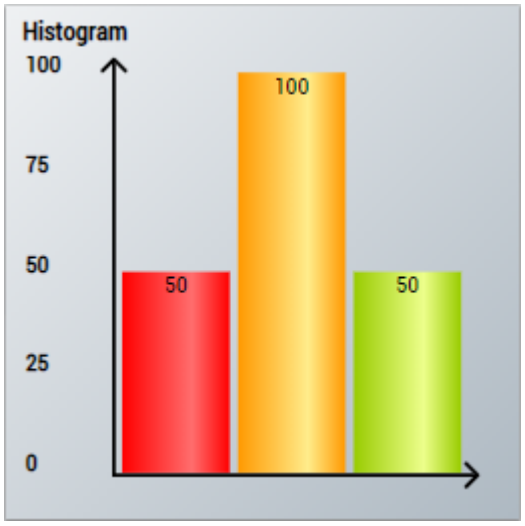
Ausgangswerte

- **Num Values:** Zeigt die Gesamtanzahl der analysierten Werte für die Verteilung an.
- **Histogramm:** Unter den Num Values wird das Histogramm angezeigt. Wenn Sie den Cursor über die Balken bewegen, sehen Sie einen Tooltip mit dem Wertintervall und dem zugehörigen Wert des Histogramm-Bins.

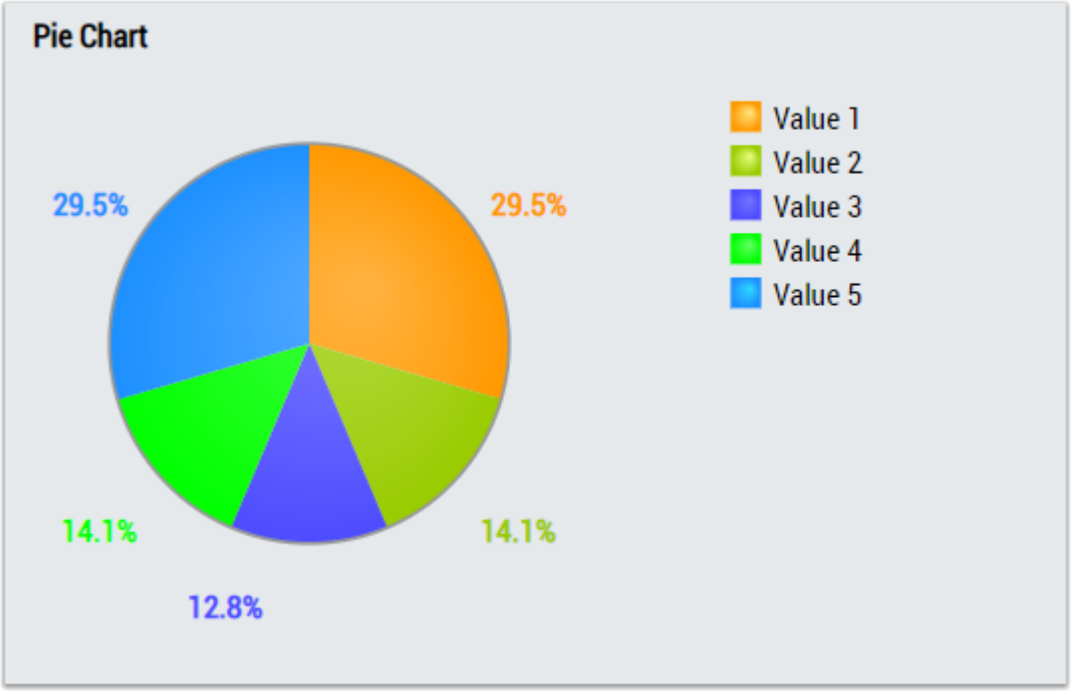
Standard HMI Controls

Für den Algorithmus Histogram 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Histogram Control visualisiert den Ausgangswert Num Values.



2. Das PieChart Control visualisiert den Ausgangswert Num Values.



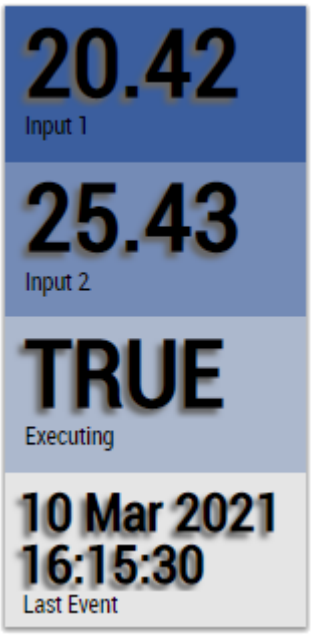
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Num Values, Histogram.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

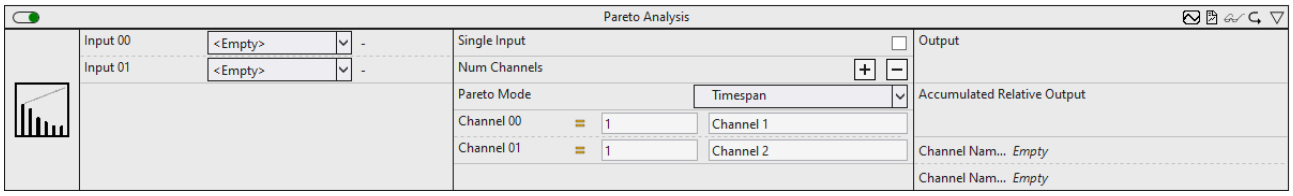
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Histogram 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.5 Pareto Analysis



Das *Pareto Analysis*-Modul ermöglicht die Kombination verschiedener Eingangsdaten zur Anzeige eines Pareto-Charts. Verwendung findet dies in unter anderem in folgenden Anwendungsfällen:

Fehleranalyse: Identifizierung der häufigsten Fehlerquellen

Qualitätsmanagement: Bestimmung der Hauptursachen von Qualitätsproblemen

Zeitmanagement: Diagnose, welche Aufgaben den größten Teil der Zeit beanspruchen

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Single Input:** Auswahl, ob ein einzelner Eingang oder mehrere unabhängige Eingangskanäle verwendet werden sollen.
- **Num Channels:** Anzahl der Eingangskanäle, wenn *Single Input* `FALSE` ist.
- **Num Elements:** Anzahl der zu vergleichenden Elemente, wenn *Single Input* `TRUE` ist.
- **Pareto Mode:** Modus für die Berechnung der Ausgangswerte.

  Count: Es werden die Zyklen gezählt, in denen der jeweilige Eingangskanal `TRUE` ist.

  Timespan: Es werden die Zeitspannen aufsummiert, in denen der Eingangskanal `TRUE` ist.
- **Threshold 00, Threshold 01, …, Threshold n**: Vergleichsoperator und Schwelle für die Überwachung von Eingangswerten.
- **Channel Name 00, Channel Name 01, …, Channel Name n**: Alias des Eingangskanals.

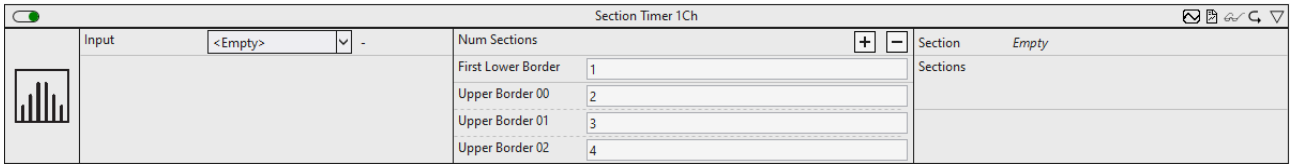
Ist die Konfigurationsoption *Single Input* gesetzt und wird ein Eingang vom Typ Enumeration auf den Eingang verlinkt, erfolgt ein automatisches Setzen von Konfigurationsoptionen:

Anzahl der Eingangskanäle: 1
Anzahl der zu vergleichenden Elemente: Anzahl der Enumerationen
Schwellwerte: Werte der einzelnen Enumerationen

Ausgangswerte

- **Output:** Sortierte Absolutwerte. Der Ausgabetyp ist abhängig vom Pareto-Modus.
- **Accumulated Relative Output**: Akkumulierte relative Ausgangswerte.
- **Channel Name 00, Channel Name 01, … Channel Name n:** Sortierte Alias-Namen der Eingangskanäle.

6.4.2.6 Section Timer 1 Ch



Der *Section Timer 1Ch* berechnet die Zeitspanne, in der sich der Eingang im Bereich jedes konfigurierten Abschnitts befindet. Er kann mit der Anzahl der Abschnitte und den Grenzen jedes Abschnitts konfiguriert werden. Jeder Abschnitt wird mit einer unteren Grenze (größer als oder gleich) und einer oberen Grenze

(kleiner als) definiert. Die untere Grenze des folgenden Abschnitts wird durch die vorherige obere Grenze gesetzt. Werte, die kleiner als die minimale Grenze sind, werden im ersten Array-Element gespeichert. Werte, die gleich oder größer als die maximale Grenze sind, werden im letzten Array-Element gespeichert.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Sections:** Dies ist die Anzahl der Abschnitte.
- **First Lower Border:** Dies ist die untere Grenze des ersten Abschnitts.
- **Upper Border 00, Upper Border 01, …, Upper Border n**: Dies sind die oberen Grenzen aller Abschnitte.
- **Enable Display Names**: Aktivierung von Section Display Names. Diese Konfiguration ist versteckt in dem Property-Fenster sichtbar.
- **Section Display Names:** Displaynamen der einzelnen Sections. Diese Konfiguration ist versteckt in dem Property-Fenster sichtbar und wird für das HMI verwendet.

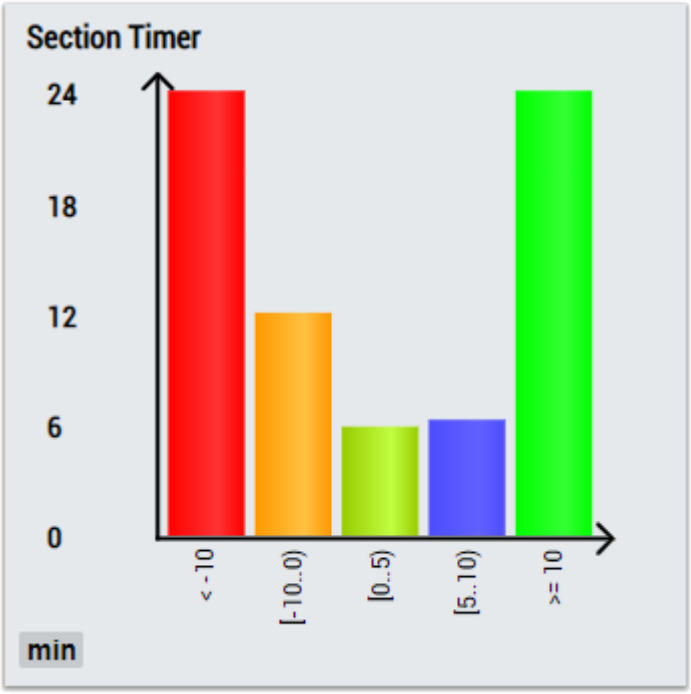
Ausgangswerte

- **Section:** Gibt den Abschnitt des letzten Eingangswerts an. Wenn der Eingangswert kleiner ist als die *First Lower Border*, ist der Rückgabewert null. Wenn der Eingangswert im Intervall [*First Lower Border*, *Upper Border 00*) liegt, ist der Rückgabewert eins, für das Intervall [*Upper Border 00*, *UpperBorder 01*) ist er zwei usw. Wenn der Eingangswert größer ist als die letzte festgelegte Grenze *Upper Border 0n*, ist der Rückgabewert *NumSections*+1.
- **Array of Timespans**: Die Gesamtzeit, in der der Eingangswert in jeden Abschnitt sortiert war.

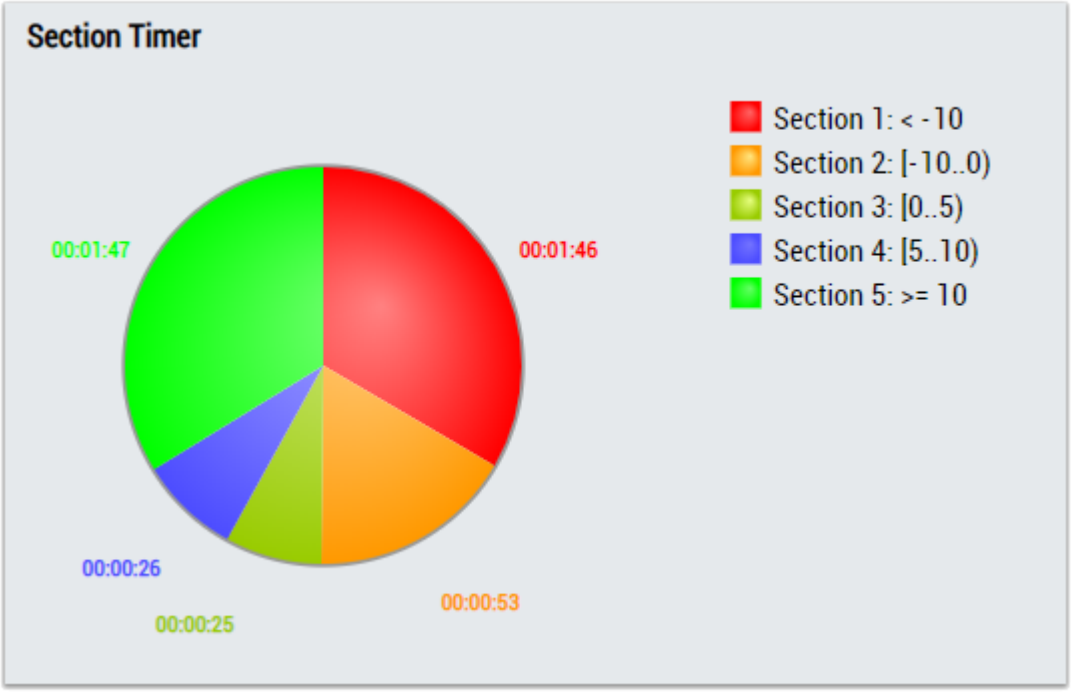
Standard HMI Controls

Für den Algorithmus Section Timer 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SectionTimer Control visualisiert den Eingangswert Array of Timespans und die Konfigurationsoptionen First Lower Border und Upper Border.

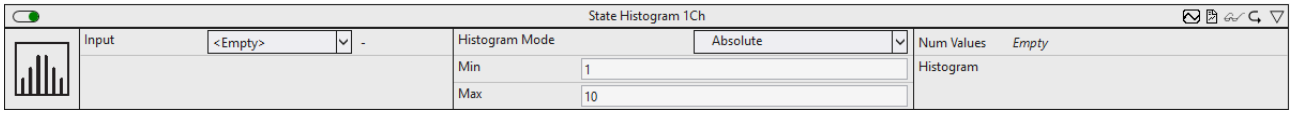


2. Das PieChart Control visualisiert den Eingangswert Array of Timespans und die Konfigurationsoptionen First Lower Border und Upper Border.



Alternativ können im Algorithmus Section Timer 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.7 State Histogram 1Ch



Der Algorithmus *State Histogram 1Ch* zählt, wie oft das Eingangssignal (INT) einen bestimmten Wert zwischen dem konfigurierten Minimum und Maximum einnimmt und zeigt die Verteilung in einem Histogramm. Der erste Balken stellt die Grenzwerte dar, die kleiner als das Minimum sind, und der letzte Balken stellt die Grenzwerte dar, die größer als das Maximum sind. Der *State Histogram 1Ch* eignet sich für Zustandsmaschinen, um zu zeigen, wie oft die verschiedenen Zustände ausgeführt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Histogram Mode:** Die Betriebsart des Histogramms. Es wird zwischen einer absoluten und einer relativen Ausgabe unterschieden. Letzteres kann verwendet werden, um die prozentuale Verteilung anzuzeigen.
- **Min:** Minimaler Wert, der analysiert werden soll.
- **Max:** Maximaler Wert, der analysiert werden soll.
- **Enable Display Names**: Aktivierung von Histogram Display Names. Diese Konfiguration ist versteckt in dem Property Fenster sichtbar.
- **Histogram Display Names:** Displaynamen der einzelnen Bins. Diese Konfiguration ist versteckt in dem Property Fenster sichtbar und wird für das HMI verwendet.

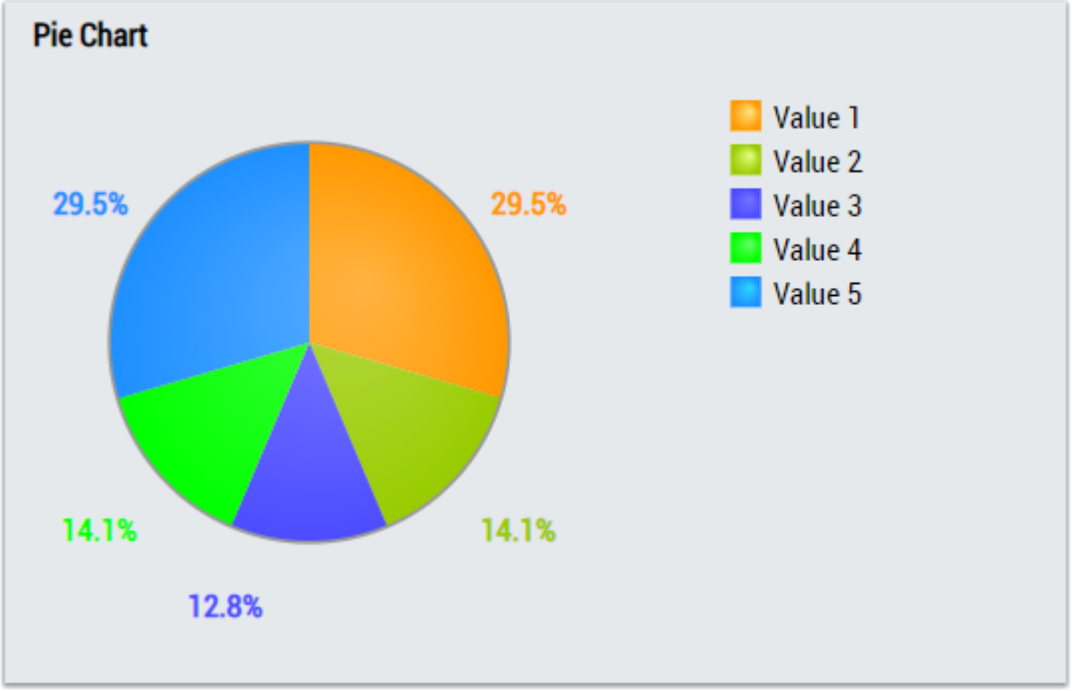
Ausgangswerte

- **Num Values:** Zeigt die Gesamtanzahl der ausgeführten Zustände zwischen den konfigurierten Grenzen an.
- **Histogramm:** Unter den Num Values wird das Histogramm angezeigt. Auf jedem Balken wird der jeweilige Wert angegeben. Wenn Sie den Cursor über die Balken bewegen, sehen Sie einen Tooltip mit dem Wertintervall und dem zugehörigen Wert des Histogramm-Bins.

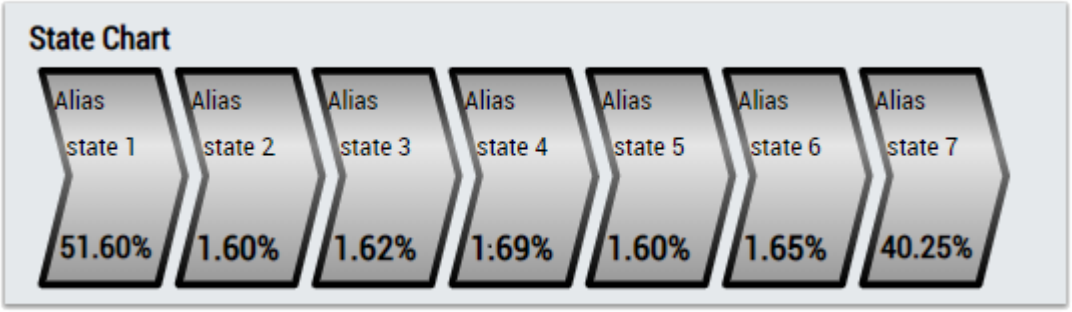
Standard HMI Controls

Für den Algorithmus State Histogram 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das PieChart Control visualisiert den Ausgangswert Histogram.



2. Das StateHistogram Control visualisiert den Ausgangswert Histogram.



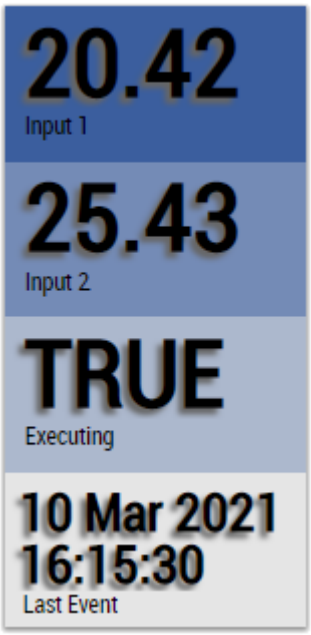
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Num Values, Histogram.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

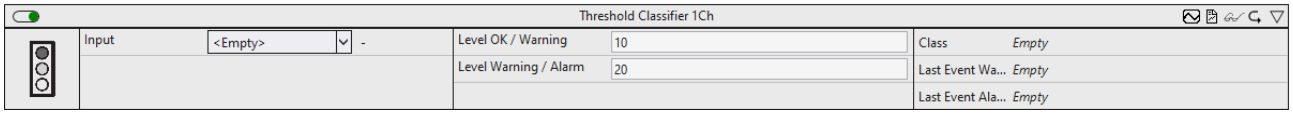
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus State Histogram 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.8 Threshold Classifier 1Ch



Der *Threshold Classifier 1Ch* stuft die Eingangswerte in drei verschiedene Klassen ein. *OK*, *Warning* und *Alarm* gemäß den konfigurierten Schwellen.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Level Ok / Warning:** Die Eingangswerte, die kleiner als die konfigurierte Schwelle sind, werden in die Klasse *OK* eingestuft, und die Eingangswerte, die gleich oder größer als die konfigurierte Schwelle sind, werden in die Klasse *Warning* eingestuft.
- **Level Warning / Alarm:** Die Eingangswerte, die kleiner als die konfigurierte Schwelle sind, werden in die Klasse *Warning* eingestuft, und die Eingangswerte, die gleich oder größer als die konfigurierte Schwelle sind, werden in die Klasse *Alarm* eingestuft.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

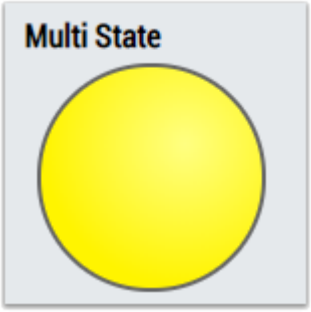
- **Class:** Zeigt die Klasse an, zu der die Eingangswerte gehören.
- **Last Event Warning:** Gibt den Zeitpunkt des letzten ausgelösten Warnungsereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Last Event Alarm:** Gibt den Zeitpunkt des letzten ausgelösten Alarmereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

Standard HMI Controls

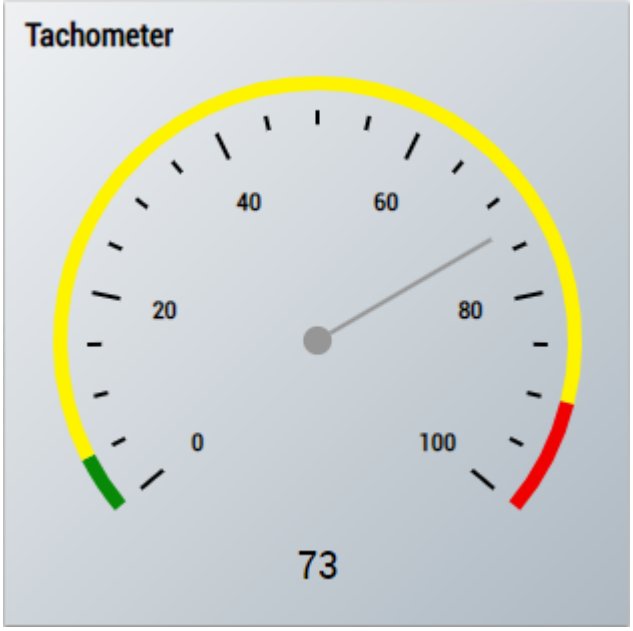
Für den Algorithmus Threshold Classifier 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das TrafficLight Control visualisiert die Ausgangswerte Class, Last Event Warning und Last Event Alarm sowie den Eingangswert der Daten.

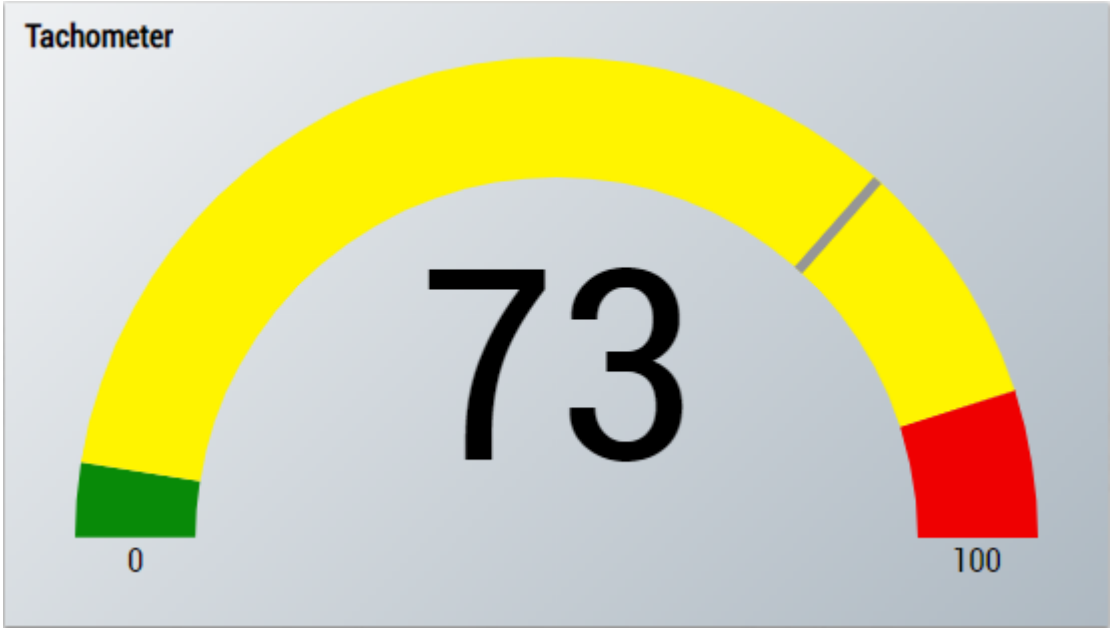
2. Das MultiState Control visualisiert den Ausgangswert Class, optional mit einem Smiley.



3. Das Tachometer Control visualisiert den Eingangswert inklusive der konfigurierten Grenzwerte.



4. Das Radial Gauge Control visualisiert den Eingangswert inklusive der konfigurierten Grenzwerte.



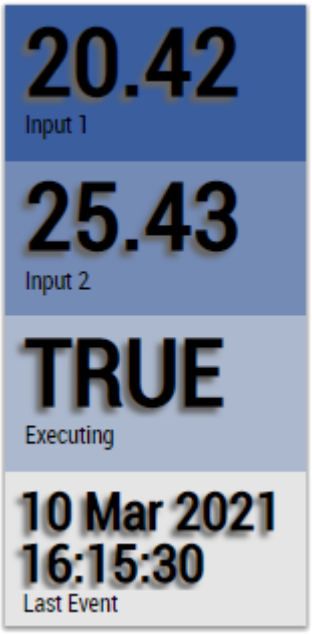
5. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Class, Last Event Warning Last Event Alarm.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

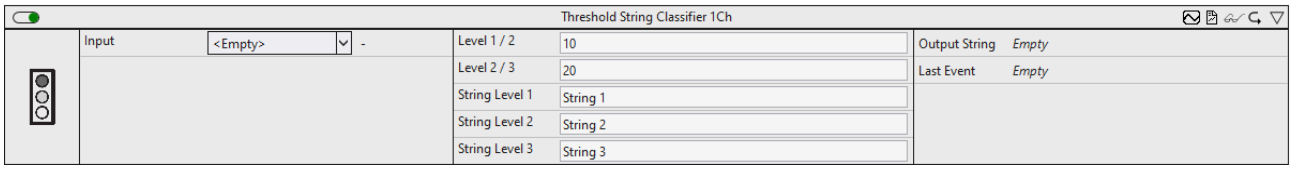
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Threshold Classifier 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.9 Threshold String Classifier 1Ch



Der Algorithmus *Threshold String Classifier 1Ch* stuft die Eingangswerte in drei verschiedene Klassen gemäß den konfigurierten Schwellen ein. Die Klassennamen (Output String) können einzeln als *String 1*, *String 2* und *String 3* konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Level 1 / 2:** Die Eingangswerte, die kleiner als die konfigurierte Schwelle sind, werden in die *erste* Klasse eingestuft, und die Eingangswerte, die gleich oder größer als die konfigurierte Schwelle sind, werden in die *zweite* Klasse eingestuft.
- **Level 2 / 3:** Die Eingangswerte, die kleiner als die konfigurierte Schwelle sind, werden in die *zweite* Klasse eingestuft, und die Eingangswerte, die gleich oder größer als die konfigurierte Schwelle sind, werden in die *dritte* Klasse eingestuft.
- **String 1:** Name der ersten Klasse.
- **String 2:** Name der zweiten Klasse.
- **String 3:** Name der dritten Klasse.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

Ausgangswerte

- **Output String:** Zeigt die Klasse an, zu der die Eingangswerte gehören.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

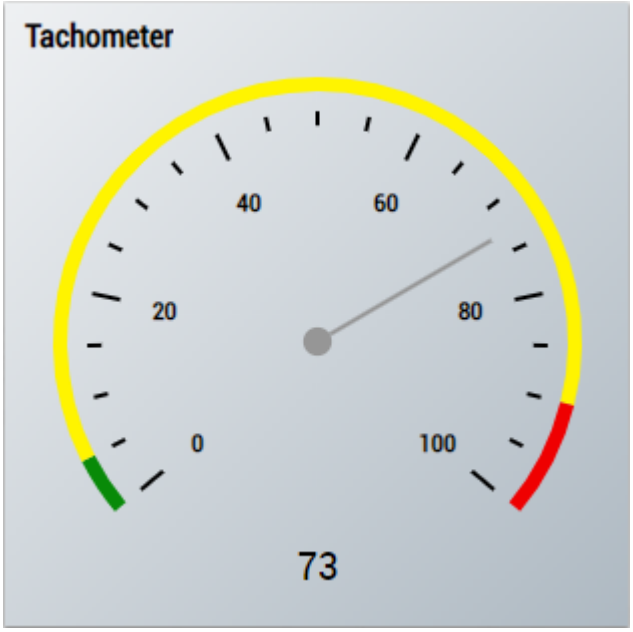
Standard HMI Controls

Für den Algorithmus Threshold String Classifier 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

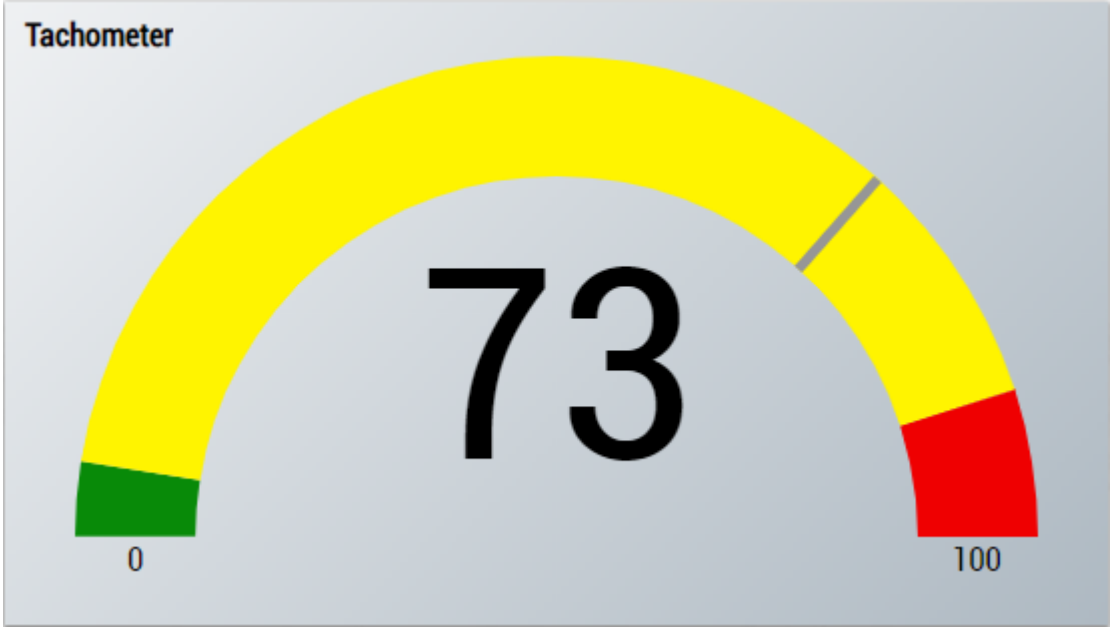
1. Das TrafficLight Control visualisiert die Ausgangswerte Output String und Last Event sowie den Eingangswert der Daten.

2. Das SingleValue Control visualisiert den Ausgangswert Output String.

3. Das Tachometer Control visualisiert den Eingangswert inklusive der konfigurierten Grenzwerte.



4. Das Radial Gauge Control visualisiert den Eingangswert inklusive der konfigurierten Grenzwerte.



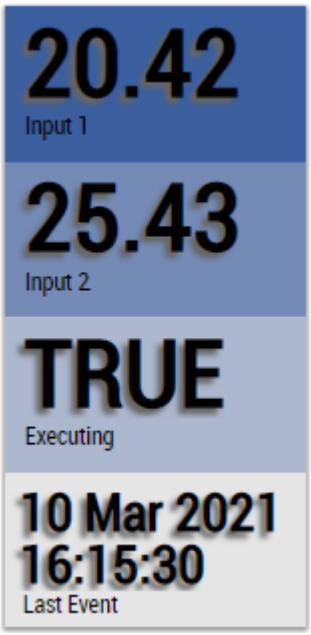
5. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Output String, Last Event.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

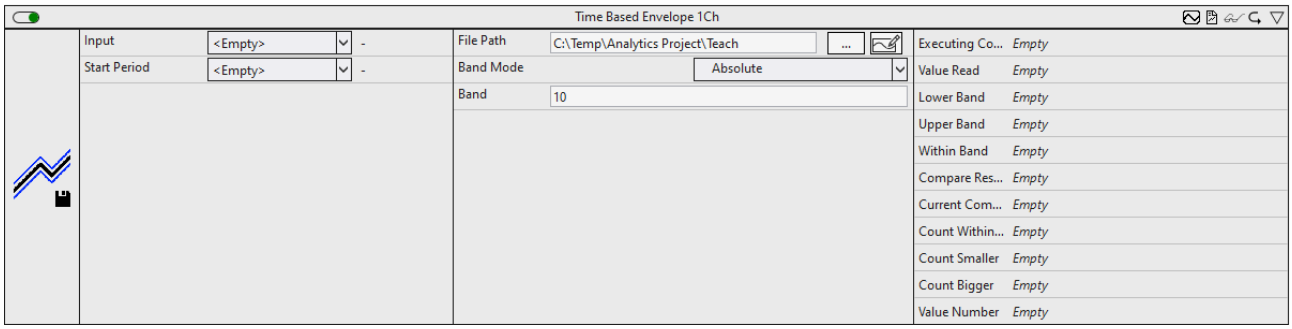
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Threshold String Classifier 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.10 Time Based Envelope 1Ch



Der Algorithmus *Time Based Envelope 1Ch* vergleicht, ob die periodischen Eingangsdaten innerhalb einer konfigurierten Bandbreite von Werten liegen, die aus einer Datei gelesen werden. Dabei kann es sich beispielsweise um ein Referenzsignal handeln, welches zuvor mit dem Time Based Teach Path 1Ch [▶ 259] erlernt wurde. Der Vergleich beginnt, wenn das Signal des Flags Start Period *TRUE* ist. Es wird empfohlen, *Time Based Envelope 1Ch* aufgrund des konkurrierenden Dateizugriffs nicht gleichzeitig mit Time Based Teach Path 1Ch [▶ 259] zu verwenden. Stattdessen sollte zunächst ein Referenzsignal mit dem Time Based Teach Path 1Ch [▶ 259] eingelernt werden und erst im Anschluss die Auswertung mithilfe des *Time Based Envelope 1Ch* erfolgen.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **File Path:** Pfad zur zuvor eingelernten Datendatei.
- **Band Mode:** Modus des Bandbetriebs (absolute oder relative Werte verwenden).
- **Band:** Bandbreite des Bandbetriebs.

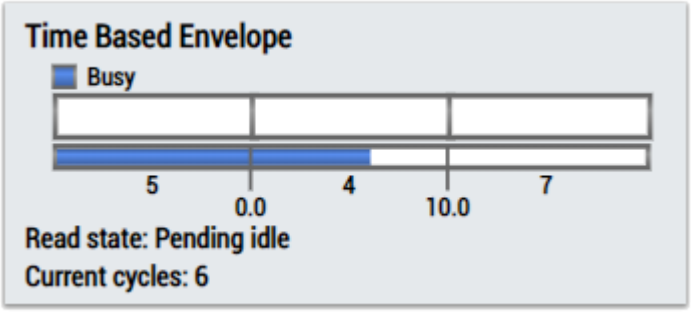
Ausgangswerte

- **Executing Compare:** Zeigt *TRUE* an, wenn der Algorithmus die Hüllkurve verarbeitet, ansonsten *FALSE*. Der Hüllkurvenprozess beginnt, wenn das Flag Start Period *TRUE* ist.
- **Lower Band:** Zeigt den Wert des unteren Bandes abhängig vom Bandmodus an.
- **Upper Band:** Zeigt den Wert des oberen Bandes abhängig vom Bandmodus an.
- **Within Band:** Zeigt *TRUE* an, wenn die aktuellen Werte innerhalb des Bandes liegen, ansonsten *FALSE*.
- **Compare Result:** Ergebnis des aktuellen Vergleichs. Zeigt an, ob die aktuellen Werte innerhalb des Bandes liegen oder kleiner oder größer als das Band sind.
- **Current Compared Cycles:** Anzahl der Zyklen, die verglichen worden sind.
- **Count Within Band:** Zählt, wie oft die Werte innerhalb des Bandes lagen.
- **Count Smaller:** Zählt, wie oft die Werte kleiner als das Band waren.
- **Count Bigger:** Zählt, wie oft die Werte größer als das Band waren.
- **Value Number:** Zeigt die Wertzahl des Datenpunkts in der Datei an, die gegenwärtig verglichen wird.

Standard HMI Controls

Für den Algorithmus Time Based Envelope 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das TimeBasedEnvelope Control visualisiert die Ausgangswerte Executing Comparison, Lower Band, Upper Band, Within Band, Compare Result, Count Within Band, Count Smaller, Count Bigger, Current Compared Cycles, State und Value Number.



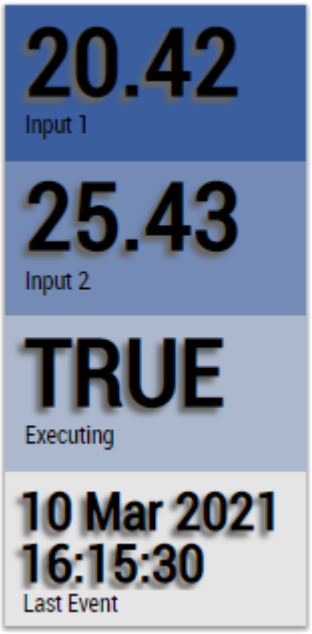
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Executing Comparison, State, Lower Band, Upper Band, Within Band, Compare Result, Current Compared Cycles, Count Within Band, Count Smaller, Count Bigger, Value Number.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

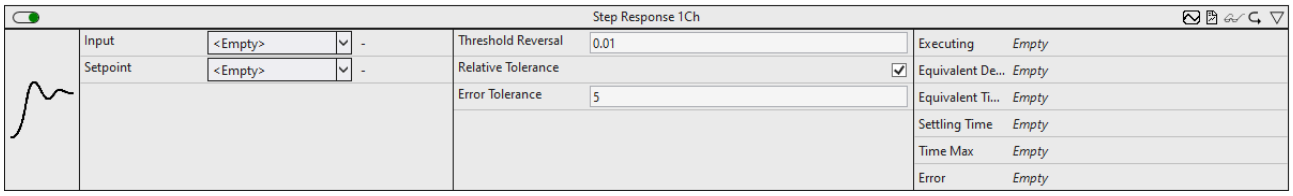
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

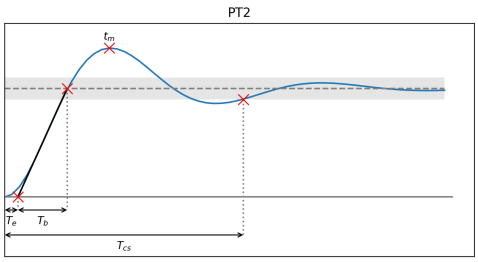


Alternativ können im Algorithmus Time Based Envelope mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.2.11 Step Response 1Ch



Step Response 1Ch identifiziert Parameter der Sprungantwort einer PT2-Strecke. Dazu zählen die Verzugszeit T_e, die Ausgleichszeit T_b, die Ausregelzeit T_{cs} sowie der Zeitpunkt des Maximums t_m.



Um zu erkennen, ob die Strecke eingeschwungen ist, wird nach einem lokalen Minimum oder Maximum gesucht. Ist dieses innerhalb des Toleranzbandes (grau eingezeichnet), wird angenommen, dass die Strecke eingeschwungen ist. Erst dann wird die Ausregelzeit gesetzt.

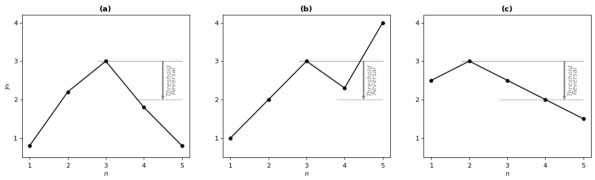
Der Algorithmus startet, wenn ein neuer Sollwert außerhalb des zuvor gespeicherten Toleranzbandes liegt.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Threshold Reversal:** Schwellenwert für die Identifizierung von Umkehrungen. Umkehrungen werden nur erkannt, wenn ihre Differenz zur nächsten Umkehrung den Wert von Threshold Reversal überschreitet.
  Nachfolgend sind drei Beispiele für die Identifizierung von Peaks mit dem Parameter *Threshold Reversal* aufgeführt.
  (a) Der Wert y_3 wird direkt nach der Verarbeitung des Werts y_4 als Peak identifiziert, da die Differenz

zwischen y_3 und y_4 größer ist als *Threshold Reversal*.
(b) Der Wert y_3 wird nicht als Peak identifiziert, da die Differenz zwischen y_3 und y_4 kleiner ist als *Threshold Reversal* und die Kurve nach y_4 wieder ansteigt.
(c) Der Wert y_2 wird nach der Verarbeitung des Werts y_5 als Peak identifiziert, da die Differenz zwischen y_2 und y_5 *Threshold Reversal* überschreitet. Der Wert y_2 kann vorher nicht als Peak identifiziert werden, da die Differenz zwischen y_2 und y_3 (y_4) kleiner als/gleich *Threshold Reversal* ist und nicht bekannt ist, ob die Werte weiter sinken.



- **Relative Tolerance:** Boolsches Flag. Ist dieses Flag *True*, so bezieht sich der Parameter *Error Tolerance* prozentual auf den Sollwert am Eingang. Andernfalls wird ein absolutes Toleranzband berücksichtigt.
- **Error Tolerance:** Gibt die Größe des Toleranzbandes in Abhängigkeit zum Parameter *Relative Tolerance* an. Da Toleranzband wird bei einem Neustart der Parameteridentifizierung aktualisiert.

Ausgangswerte

- **Executing:** True, wenn die Parameteridentifizierung aktiv ist.
- **Equivalent Dead Time:** Verzugszeit der Strecke. Zeitspanne zwischen dem Start der Sprungantwort und dem Schnittpunt der Wendetangenten mit dem Startwert.
- **Equivalent Time Constant:** Ausgleichszeit der Strecke. Zeitspanne zwischen dem Schnittpunt der Wendetangenten mit dem Startwert und dem Schnittpunt der Wendetangenten mit dem Sollwert.
- **Settling Time:** Ausregelzeit der Strecke.
- **Time Max:** Zeitpunkt des maximalen Überschwingers.
- **Error:** Differenz zwischen Eingangswert und Sollwert.

Standard HMI Controls

Für den Algorithmus *Step Response 1Ch* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte.

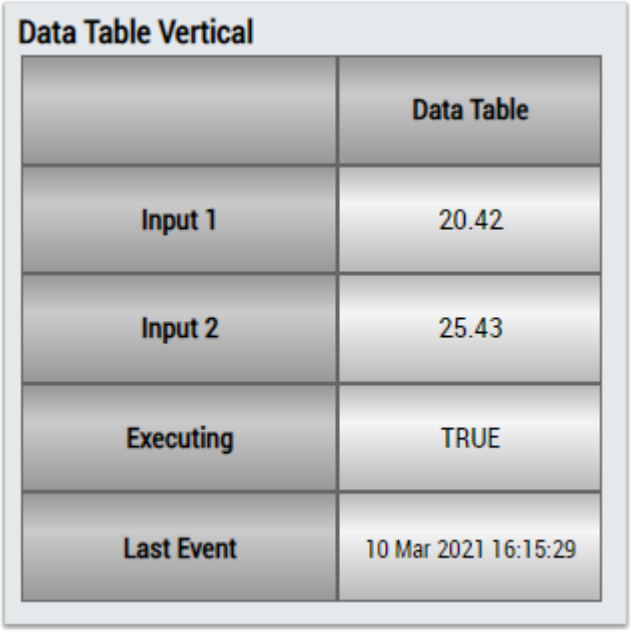


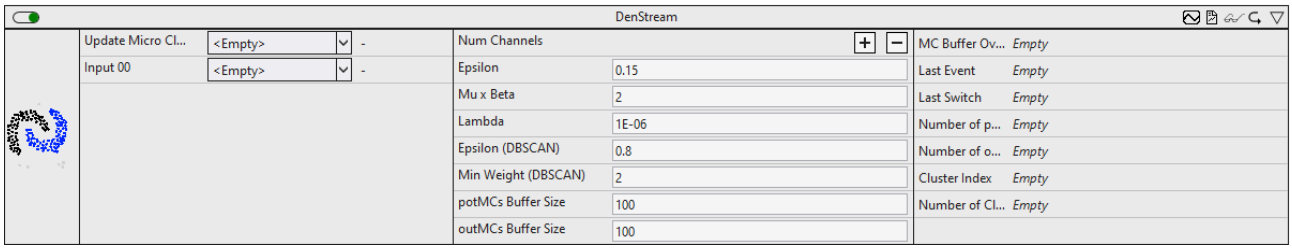
Abb. 2:

Alternativ können mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.3 Analytics - Clustering

Die Algorithmen der Kategorie *Clustering* bieten Funktionen für das Streamingdatenbasierte Clustering der Eingangsdaten in verschiedene, im Vorfeld nicht definierte, Cluster. Dabei werden die Cluster anhand der in den Eingangsdaten vorhandenen Strukturen detektiert. Beispiele für solche Cluster-Algorithmen sind der *Sequential k-Means* sowie der *DenStream-Algorithmus*.

6.4.3.1 DenStream



DenStream ist eine Implementierung des gleichnamigen unüberwachten, dichtebasierten Clustering-Algorithmus [1]. Er basiert auf dem bekannten Clustering-Algorithmus DBSCAN [2, 3] und eignet sich insbesondere für Datenströme, deren Strukturen sich im Zeitverlauf verändern.

Die Anzahl der Eingangskanäle (im Folgenden als *n* bezeichnet) für diesen Algorithmus kann vom Benutzer gewählt werden. Diese Eingänge bilden den n-dimensionalen Merkmalsraum, in dem Cluster zu finden sind. In jedem Analysezyklus liefert der Datenstrom dem Algorithmus einen neuen Merkmalsvektor, der als Datenpunkt in diesem Merkmalsraum interpretiert werden kann. Cluster sind separierbare Bereiche mit einer hohen Dichte von Datenpunkten im Merkmalsraum.

In der ersten Phase des Algorithmus werden die eingehenden Datenpunkte sogenannten Mikro-Clustern (*MCs*) zugeordnet. Diese Mikro-Cluster haben Eigenschaften (wie Mittelpunkt, Gewicht und Varianz), die von den enthaltenen Datenpunkten abhängen. Nur Mikro-Cluster, deren Gewicht eine bestimmte Schwelle überschreitet, gelangen in die zweite Phase und werden vom DBSCAN-Algorithmus geclustert. Somit ist es nicht erforderlich, die Informationen über jeden einzelnen Datenpunkt zu behalten. Dadurch reduziert sich der Speicherbedarf, da es mit der Zeit viel weniger Mikro-Cluster als Datenpunkte gibt. Und auch der Rechenaufwand für den DBSCAN-Algorithmus ist viel geringer, da er über den reduzierten Satz von Mikro-Clustern und nicht über alle Datenpunkte läuft. Außerdem ist es möglich, auf die Gewichte der Mikro-Cluster eine Fading-Funktion anzuwenden. Auf diese Weise verlieren alte Datenpunkte mit der Zeit ihre Bedeutung für den Clustering-Prozess. Dadurch können Änderungen (wie die Verschiebung von Clustern oder ihr Verschwinden/Erscheinen mit der Zeit) vom Algorithmus erfasst werden.

Der DenStream-Algorithmus hat weitere Vorteile gegenüber anderen Clustering-Algorithmen. Der Benutzer muss die Anzahl der Mikro-Cluster nicht im Voraus kennen, da der DenStream-Algorithmus diese Anzahl automatisch bestimmt. Zudem ist der Algorithmus in der Lage, Ausreißer in den Daten zu erkennen, die zu keinem Cluster gehören. Da es sich um einen dichtebasierten Algorithmus handelt, ist es sogar möglich, separate Cluster beliebiger Form zu erkennen (auch wenn sie ineinander verschlungen sind).

Parametereinstellung

Hier geben wir eine kurze Einführung in die Funktionsweise des Algorithmus, hauptsächlich um dem Leser einen schnellen Einstieg in die Parametereinstellung zu ermöglichen. Für ein tiefgreifendes Verständnis des Algorithmus und seiner Parameter weisen wir den Leser auf die genannten Veröffentlichungen hin. Die meisten der hier verwendeten Begriffe und Parameternamen stammen direkt aus diesen Veröffentlichungen.

Die Parameter des DenStream-Algorithmus beeinflussen vor allem folgende Eigenschaften des Algorithmus:

- Die Grobheit der Mikro-Cluster,
- Die maximale Entfernung zwischen Datenpunkten/Mikro-Clustern, damit sie demselben Cluster zugeordnet werden,
- Die Mindestdichte, damit Datenpunkte als Cluster und nicht als Ausreißer erkannt werden,
- Die Fading-Rate, mit der ältere Datenpunkte ihre Bedeutung verlieren.

Die Parameter *Epsilon, Lambda* und *Mu x Beta* gehören zur ersten Phase des Algorithmus, der Bildung von Mikro-Clustern.

Nach Möglichkeit wird ein Datenpunkt in das Mikro-Cluster eingefügt, dessen Mittelpunkt dem Datenpunkt am nächsten liegt. Dazu werden die euklidischen Abstände zwischen dem Datenpunkt und den Mittelpunkten aller Mikro-Cluster verglichen und das Mikro-Cluster mit dem geringsten Abstand ausgewählt. Der Datenpunkt kann nur in das Mikro-Cluster eingefügt werden, wenn der Radius des Mikro-Clusters nach dem Einfügen nicht die Schwelle *Epsilon* überschreitet. Der Radius ist analog zur Varianz aller Datenpunkte, die im Mikro-Cluster enthalten sind. D.h. es können auch Datenpunkte in ein Mikro-Cluster integriert werden, deren euklidische Distanz zum Mittelpunkt des Clusters größer ist als *Epsilon*, solange genügend andere Punkte im Mikro-Cluster einen geringeren Abstand haben.

Wenn der Datenpunkt nicht in das nächste Mikro-Cluster eingefügt werden kann, wird mit diesem Datenpunkt ein neues Mikro-Cluster erstellt. Das Gewicht des jeweiligen Mikro-Clusters wird mit dem Einfügen eines Datenpunktes um eins erhöht.

Im linken Plot der Abbildung ist die Zuordnung der Datenpunkte zu den Mikro-Clustern exemplarisch für zwei Eingangskanäle skizziert. Abgebildet sind 20 Datenpunkte, die vier verschiedenen Mikro-Clustern zugeordnet werden. Das erste Mikro-Cluster enthält sechs Datenpunkte (#1, rot markiert), das zweite (#2, grün markiert) auch sechs, das dritte (#3, blau markiert) enthält sieben und das vierte (#1, grau markiert) nur einen Datenpunkte. Farbig hinterlegt ist jeweils der Bereich um den Mittelpunkt der Mikro-Cluster in dem sich ein neuer Datenpunkt befinden müsste, um bei gegebenem Epsilon (markiert durch gestrichelte Linie) in das jeweilige Mikro-Cluster aufgenommen zu werden. Dieser Einflussbereich ist größer, wenn bereits mehrere Datenpunkte im Mikro-Cluster enthalten sind und diese eine geringere Streuung aufweisen (vergleiche z.B. Mikro-Cluster #1 und #2). Außerdem können sich die Einflussbereiche von mehreren Mikro-Clustern überschneiden, und sich durch ihre Existenz gegenseitig beeinflussen, siehe Mikro-Cluster #2 (grün) und #3 (blau). Die Datenpunkte werden stets dem näheren Mikro-Cluster zugewiesen, weshalb die Einflussbereiche durch eine gerade Linie abgetrennt sind. Im Plot ist ein Datenpunkt zu sehen, der Mikro-Cluster #2 zugeordnet ist, wenn dieses jedoch nicht existieren würde, dann wäre er Mikro-Cluster #3 zugeordnet.

Wie auch in der Originalarbeit [1] werden Mikro-Cluster abhängig von ihrem Gewicht in potenzielle und Ausreißer-Mikro-Cluster aufgeteilt. Nur potenzielle Mikro-Cluster werden anschließend vom DBSCAN-Algorithmus geclustert. Die Datenpunkte in den Ausreißer-Mikro-Clustern werden als Ausreißer gekennzeichnet. Aber auch Ausreißer-Mikro-Cluster werden gespeichert und durch neue Datenpunkte aktualisiert, da sie sich noch zu potenziellen Mikro-Clustern entwickeln können. Das Gewicht eines Mikro-Clusters muss die Schwelle *Beta x Mu* überschreiten, um als potenzielles Mikro-Cluster gezählt zu werden. In der linken Skizze der Abbildung enthält das Mikro-Cluster #4 (grau) zum Beispiel nur einen Datenpunkt, hat damit ein Gewicht kleiner oder gleich eins und würde für *Beta x Mu* = 1 als Ausreißer-Mikro-Cluster gezählt werden.

Wenn eine Fading-Funktion angewendet wird, nimmt das Gewicht der Mikro-Cluster mit der Zeit ab. Diese Fading-Rate wird durch den Parameter *Lambda* festgelegt. Wenn der Wert auf null gesetzt wird, wird keine Fading-Funktion angewendet, anderenfalls nehmen die Gewichte jede Sekunde um einen Faktor von 2^(-*Lambda*) ab. Wenn das Gewicht eines Ausreißer-Mikro-Clusters unter eine interne Schwelle fällt (abhängig von *Mu x Beta* und *Lambda*), wird er aus dem Speicher gelöscht.

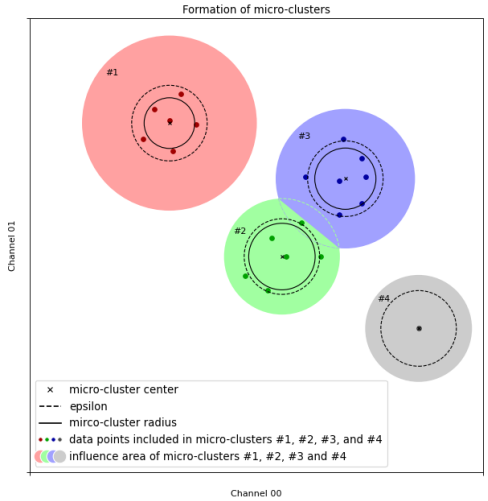
Die Parameter *Epsilon (DBSCAN)* und *Min Weight (DBSCAN)* betreffen die zweite Phase. Diese Parameter wurden vom DBSCAN-Algorithmus [3] übernommen.

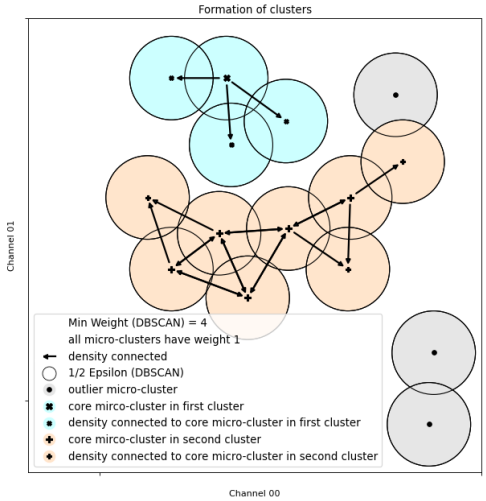
Der DBSCAN-Algorithmus läuft über den Satz potenzieller Mikro-Cluster und weist ihnen Cluster-Bezeichnungen zu. Dies kann entweder der Index des Clusters, zu dem sie gehören, oder die Bezeichnung Ausreißer sein. Dem gegenwärtig verarbeiteten Datenpunkt wird dann die Bezeichnung des Mikro-Clusters, zu dem er gehört, zugeordnet.

Wie clustert DBSCAN die Mikro-Cluster? Der Algorithmus arbeitet nach dem Konzept der Dichte-Erreichbarkeit. Objekte (in diesem Fall Mikro-Cluster) gehören zu demselben Cluster, wenn sie dichte-verbunden sind. Das bedeutet, dass es eine Kette von Mikro-Clustern mit einem maximalen Abstand *Epsilon (DBSCAN)* geben muss. Alle Mikro-Cluster, die diese Kette bilden, müssen eine zweite Bedingung erfüllen. Die Summe der Gewichte aller Mikro-Cluster innerhalb des Abstands *Epsilon (DBSCAN)* um jedes einzelne Mikro-Cluster in dieser Kette muss die Schwelle *Min Weight (DBSCAN)* überschreiten. Mikro-Cluster, die nicht mit mindestens einem Mikro-Cluster dichte-verbunden sind, welches diese zweite Bedingung erfüllt, werden als Ausreißer gekennzeichnet.

Dies ist in der rechten Skizze der Abbildung dargestellt. Zur Vereinfachung wird hier angenommen, dass die Gewichtung aller Mikro-Cluster gleich 1 ist. Dies entspricht dem Fall, dass in jedem Mikro-Cluster genau ein Datenpunkt enthalten ist und keine Fading-Funktion angewendet wurde. Die beiden Cluster (markiert mit einem „x“ (türkis) und einem „+“ (orange)) mit den zwei Ausreißer-Mikro-Clustern ergeben sich, wenn der Parameter *Min Weight (DBSCAN)* auf vier gesetzt ist. Die mit einem großen „x“ bzw. „+“ markierten Mikro-Cluster sind Core-Mikro-Cluster. Das heißt mindestens drei weitere Mikro-Cluster (plus das betrachtete Mikro-Cluster = 4) haben eine maximale Distanz von *Epsilon (DBSCAN)* zu diesen Mikro-Clustern. Die mit einem kleinen „x“ bzw. „+“ markierten Mikro-Cluster sind keine Core-Mikro-Cluster, aber befinden sich in der *Epsilon (DBSCAN)* – Nachbarschaft eines Core-Mikro-Clusters und gehören somit zum selben Cluster. Das mit einem kleinen Punkt markierte Mikro-Cluster in der rechten, oberen Ecke ist ein Ausreißer-Mikro-Cluster. Es befindet sich zwar in der *Epsilon (DBSCAN)* – Nachbarschaft eines Mikro-Clusters, welches zu einem Cluster gezählt wird, dieses ist jedoch kein Core-Mikro-Cluster.

Genauso sind die beiden Mikro-Cluster rechts unten Ausreißer. Diese befinden sich zwar in unmittelbarer *Epsilon (DBSCAN)* – Nachbarschaft, jedoch sind sie nur zu zweit. Die Schwelle *Min Weight (DBSCAN)* der Gewichte wird nicht überschritten.





Die Parameter *outMCs Buffer Size* und *potMCs Buffer Size* sind spezifisch für diese Implementierung des Algorithmus und erforderlich, weil der Speicher für Ausreißer und potenzielle Mikro-Cluster vor der Ausführung zugewiesen werden muss. Somit begrenzen *outMCs Buffer Size* und *potMCs Buffer Size* die mögliche Anzahl der Ausreißer und potenziellen Mikro-Cluster während der Laufzeit. Der Benutzer muss solche Werte für diese Parameter finden, dass diese Grenze nicht überschritten wird.

Die maximale Anzahl der Ausreißer und potenziellen Mikro-Cluster während der Ausführung des Algorithmus hängt von der Verteilung der Eingangsdaten, aber auch von der Einstellung der anderen Parameter ab. Es gibt weniger Mikro-Cluster bei höheren Werten von *Epsilon*, da dies zu gröberen Mikro-Clustern führt, die Datenpunkte aus einem größeren Bereich enthalten können. Im Allgemeinen steigt die Anzahl der Ausreißer-Mikro-Cluster am Anfang der Analyse, sinkt jedoch wieder, wenn Ausreißer-Mikro-Cluster in potenzielle Mikro-Cluster übergehen. Wenn sich die Muster im Datenstrom mit der Zeit nicht verändern, pendelt sich die Anzahl der Mikro-Cluster nach einer Anfangsphase ein.

Je mehr Mikro-Cluster vorhanden sind, desto höher sind die Rechenanforderungen. Für alle Ausreißer und potenzielle Mikro-Cluster wir die Distanz zu einem Datenpunkt verglichen und anschließend müssen alle potenziellen Mikro-Cluster in die Berechnung des DBSCAN-Algorithmus einbezogen werden. Es muss also ein Kompromiss zwischen Rechengeschwindigkeit und der Grobheit der Mikro-Cluster eingegangen werden.

Was geschieht, wenn die Werte von *outMCs Buffer Size* und *potMCs Buffer Size* zu niedrig gesetzt sind und zu irgendeinem Zeitpunkt während der Analyse mehr Mikro-Cluster erforderlich sind, um die Eingangsdatenpunkte zu erfassen? Der Algorithmus ordnet in diesem Fall weiterhin die Datenpunkte den vorhandenen Mikro-Clustern zu und kennzeichnet die Datenpunkte entsprechend, aber die vorhandenen Mikro-Cluster werden nicht mehr aktualisiert, um einen Überlauf des Puffers zu verhindern. Dies bedeutet, dass das Clustern der Datenpunkte fortgesetzt wird, aber mit einem insgesamt stagnierten Merkmalsraum (älterer Satz von Mikro-Clustern). Änderungen im Muster des Datenstroms könnten nicht mehr erkannt werden.

[1] F. Cao, M. Ester, W. Qian, A. Zhou. Density-Based Clustering over an Evolving Data Stream with Noise. In Proceedings of the 2006 SIAM International Conference on Data Mining, S. 326-337. SIAM.

[2] M. Ester, H.-P. Kriegel, J. Sander und X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of KDD*, 1996.

[3] J. Sander, M. Ester, H.-P. Kriegel, X. Xu. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications. Data Mining and Knowlage Discovery 2, 169-194 (1998)

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Eingangswerte

- **Update Mikro Cluster:** Falls TRUE, werden Mikro-Cluster durch die eingehenden Daten aktualisiert. Falls FALSE, bleiben die vorhandenen Mikro-Cluster unverändert und werden nur verwendet, um den Cluster-Index der eingehenden Datenpunkte zu bestimmen.
- **Input 01, …, Input 0n:** Diese Eingänge bilden den Merkmalsraum, für den das Clustering ausgeführt wird.

Konfigurationsoptionen

- **Num Channels:** Hier kann die Anzahl der Eingangskanäle geändert werden.
- **Epsilon:** Schwelle für den maximalen Radius von Mikro-Clustern.
- **Mu x Beta:** Schwelle für das Gewicht eines Mikro-Clusters zwischen Ausreißer und potenziellem Mikro-Cluster.
- **Lambda:** Gibt die Fading-Rate des Algorithmus an. Das Gewicht jedes Datenpunkts nimmt jede Sekunde um einen Faktor von 2^(-*Lambda*) ab.
- **Epsilon (DBSCAN):** Gibt den Parameter Epsilon des DBSCAN-Algorithmus an.
- **Min Weight (DBSCAN):** Schwelle für die Summe der Gewichte in der Epsilon-Nachbarschaft eines Mikro-Clusters für den DBSCAN-Algorithmus.
- **potMCs Buffer Size:** Die maximale Anzahl potenzieller Mikro-Cluster. Der Speicher wird *potMCs Buffer Size* Mikro-Clustern zugeordnet.
- **outMCs Buffer Size:** Die maximale Anzahl von Ausreißer-Mikro-Clustern. Der Speicher wird *outMCs Buffer Size* Mikro-Clustern zugeordnet.

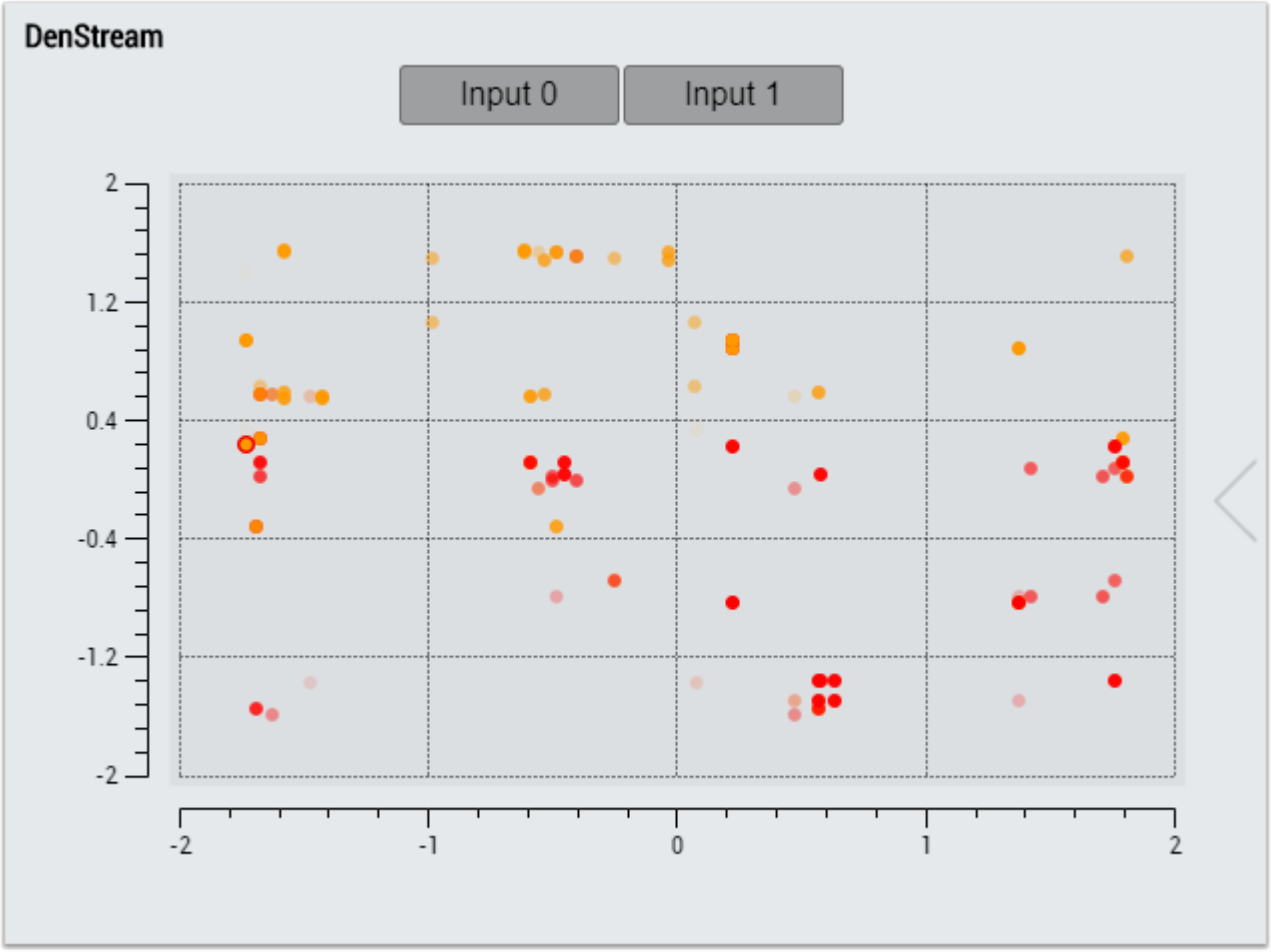
Ausgangswerte

- **New Result:** Ist TRUE, wenn sich der neue Cluster-Index vom Cluster-Index des letzten Zyklus unterscheidet.
- **MC Buffer Overflow:** Ist TRUE, wenn die Aktualisierung der Mikro-Cluster gestoppt wird, um ein Überlaufen von *potMCs* oder *outMCs Buffer Size* zu verhindern.
- **Last Event:** Dies ist der Zeitstempel des letzten Zyklus mit einer Änderung des Cluster-Index
- **Last Switch:** Dies ist der Zeitstempel des letzten Zyklus mit einem Wechsel zwischen Aktualisierung und Nichtaktualisierung von Mikro-Clustern (entweder durch Setzen des Eingangs *Update Mikro Cluster* auf TRUE oder durch internes Verhindern eines Überlaufs von *potMCs/outMCs Buffer Size*).
- **Number of potMCs:** Gibt die Anzahl der gegenwärtig vorhandenen potenziellen Mikro-Cluster an.
- **Number of outMCs:** Gibt die Anzahl der gegenwärtig vorhandenen Ausreißer-Mikro-Cluster an.
- **Cluster Index:** Gibt den Cluster-Index an, den der DBSCAN-Algorithmus für den Datenpunkt des aktuellen Zyklus ausgibt.
- **Number of Clusters:** Gibt die Gesamtanzahl der vom DBSCAN-Algorithmus erkannten Cluster an.

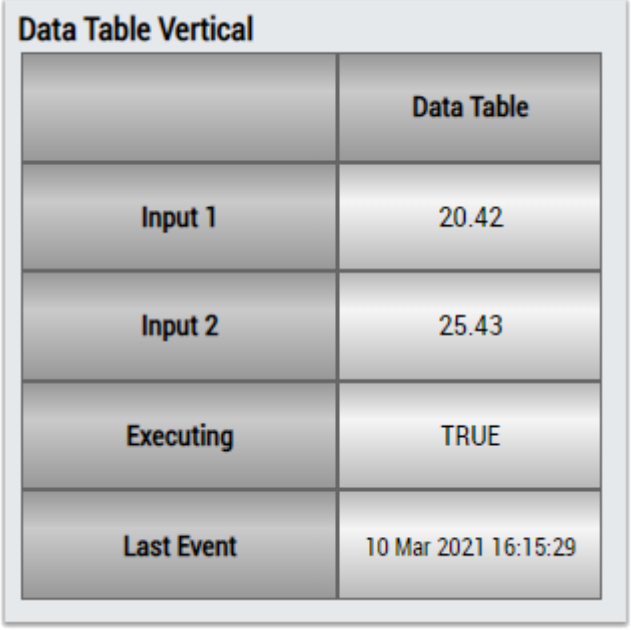
Standard HMI Controls

Für den Algorithmus DenStream sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

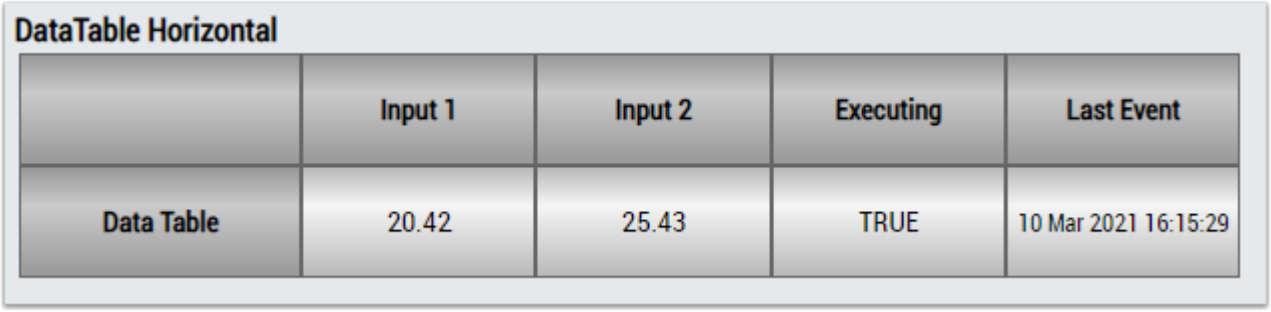
1. Das DenStream Control visualisiert im Chart die Inputs und farblich deren jeweilige Klassifikation zu einem Cluster (Cluster-Index). Über die Buttons können die anzuzeigenden Inputs ausgewählt werden, maximal zwei gleichzeitig sind möglich. Der Slider auf der rechten Seite führt alle Ausgangswerte auf: Clusteranzahl, Anzahl der potenziellen Mikro-Cluster, Anzahl der Ausreißer Mikro-Cluster, Clusterindex, letztes Event, letzter Switch.



2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Clusteranzahl, Anzahl der potenziellen Mikro-Cluster, Anzahl der Ausreißer Mikro-Cluster, Clusterindex, letztes Event, letzter Switch.

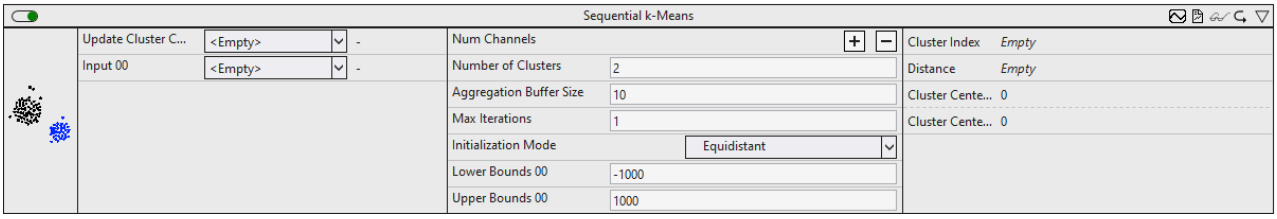


	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29



Alternativ können im Algorithmus DenStream mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.3.2 Sequential k-Means



Der Algorithmus *Sequential k-Means* ist eine Implementierung des gleichnamigen unüberwachten Clustering-Algorithmus und stellt eine sequentielle Variante des weit verbreiteten Clustering-Algorithmus k-Means für Streamingdaten dar. Das Ziel des Algorithmus ist es, anhand der Struktur der Daten Cluster zu finden, die jeweils ähnliche Datenpunkte beinhalten und unterschiedliche Datenpunkte voneinander abgrenzen.

Die Anzahl der Eingangskanäle (im Folgenden als *n* bezeichnet) für diesen Algorithmus kann vom Benutzer frei gewählt werden. Diese Eingänge spannen den n-dimensionalen Merkmalsraum auf, in dem die Cluster zu finden sind. In jedem Analysezyklus liefert der Datenstrom dem Algorithmus einen neuen Merkmalsvektor, der als Datenpunkt in diesem Merkmalsraum interpretiert werden kann. Datenpunkte, die in diesem Merkmalsraum nahe beieinanderliegen, werden dem gleichen Cluster zugewiesen. Die Anzahl der vorhandenen Cluster muss vor Beginn der Analyse vom Anwender gesetzt werden und bleibt fest.

Im Gegensatz zum k-Means Algorithmus für die klassische Batch-Analyse liegen die Daten für den *Sequential k-Means* zum Zeitpunkt der Analyse nicht vollständig vor. Stattdessen gehen die Datenpunkte in Form von Streamingdaten nach und nach ein und werden hier daher sequentiell verarbeitet und dem

entsprechenden am nächsten liegenden Cluster zugeordnet. Aus diesem Vorgehen ergeben sich einige Unterschiede, wobei zwei davon für die Benutzung des Algorithmus sowie die Parametereinstellungen besonders relevant sind.

Zum einen liegen zu Beginn einer Batch-Analyse bereits alle Datenpunkte und damit auch die Wertebereiche der einzelnen Merkmale vor, bei der sequentiellen Analyse ist dies nicht der Fall, sodass die Wertebereiche vorher nicht zwangsläufig feststehen. Es ist jedoch hilfreich, die Wertebereiche der Eingangskanäle bereits vorher zu kennen, auch wenn die Werte erst im Laufe der Analyse eintreffen. Dies spielt insbesondere für die Initialisierung der Cluster-Mittelpunkte eine Rolle. Für die Initialisierung können drei verschiedene Vorgehensweisen gewählt werden. Die Mittelpunkte können mit konkreten Werten über ein Parameter-Array vorgegeben werden *(Values)*. Die Mittelpunkte können aber auch zufällig *(Random)* oder äquidistant *(Equidistant)* in einem definierten Wertebereich gesetzt werden. Für die Initialisierungsmodi *Random* und *Equidistant* werden die Wertebereiche benötigt und müssen über die Parameter *Lower Bounds* und *Upper Bounds* für die einzelnen Eingangskanäle eingestellt werden.

Zum anderen werden in einer Batch-Analyse üblicherweise alle Datenpunkte mehrfach durchlaufen, um die Cluster-Mittelpunkte zu aktualisieren, bis diese sich nur noch minimal verändern. Das ist in der Form im Rahmen der sequentiellen Analyse nicht möglich. Um die Cluster-Mittelpunkte jedoch trotzdem anpassen zu können und Datenpunkte mehrfach zu durchlaufen, verfügt der Algorithmus *Sequential k-Means* über einen Zwischenspeichermechanismus, den *Aggregation Buffer*. Mit diesem wird es möglich, eine begrenzte Anzahl von Werten zwischenzuspeichern. Beim Füllen des Buffers, werden alle eintreffenden Datenpunkte dem am nächsten liegenden Cluster zugeordnet. Die Distanz zwischen einem Datenpunkt und den Cluster-Mittelpunkten wird durch die Euklidische Norm bestimmt. Erst wenn der Buffer gefüllt ist, werden die Cluster-Mittelpunkte anhand der neu zugewiesenen Datenpunkte im Buffer aktualisiert. Dabei entspricht der neue Cluster-Mittelpunkt dem Mittelwert aller im Cluster enthaltener Datenpunkte. Dieser kann inkrementell berechnet werden, sodass die alten Datenpunkte nicht für die Berechnung benötigt werden. Die Größe des Buffers wird durch den Parameter *Aggregation Buffer Size* gesetzt, der Default-Wert ist 10. Durch den Parameter *Max Iterations* kann zusätzlich angegeben werden, wie oft durch den Buffer iteriert wird. Hier ist der Default-Wert gleich eins. Wird der Wert beispielsweise auf zwei gesetzt, werden nach dem ersten Anpassen der Cluster-Mittelpunkte, die Datenpunkte im Buffer den Clustern neu zugewiesen und anschließend die Cluster-Mittelpunkte erneut angepasst. Aufgrund der Verschiebung der Cluster-Mittelpunkte ist es möglich, dass einzelne Datenpunkte von einer Iteration zur nächsten unterschiedlichen Clustern zugewiesen werden können. Aufgrund der begrenzten Rechenkapazität für die Datenverarbeitung zwischen zwei Zyklen sollten für die Parameter *Aggregation Buffer Size* und *Max Iterations* keine zu hohen Werte vergeben werden, da die Aktualisierung der Cluster-Mittelpunkte andernfalls eventuell nicht gewährleistet werden kann. Werden die Cluster-Mittelpunkte bei großen Werten für diese Parameter nicht aktualisiert, bei kleineren Parameterwerten hingegen schon, ist dies ein Indiz dafür, dass die Rechenkapazität für die eingestellten Parameterwerte nicht ausreicht und kleinere Werte gewählt werden sollten.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Eingangswerte

- **Update Cluster Centers:** Falls TRUE, werden die Mittelpunkte der einzelnen Cluster durch die eingehenden Daten aktualisiert. Falls FALSE, bleiben die Cluster-Mittelpunkte Cluster unverändert und werden nur verwendet, um den Cluster-Index der eingehenden Datenpunkte zu bestimmen.
- **Input 01, …, Input 0n:** Diese Eingänge bilden den n-dimensionalen Merkmalsraum, für den das Clustering ausgeführt wird.

Konfigurationsoptionen

- **Num Channels:** Bestimmt die Anzahl der Eingangskanäle.
- **Number of Clusters:** Definiert die Anzahl der Cluster.
- **Aggregation Buffer Size:** Gibt die Größe des Aggregation Buffers und damit die Anzahl an Zyklen an, nach denen die Cluster-Mittelpunkte aktualisiert werden. Die Eingangswerte dieser Zyklen werden intern (im Aggregation Buffer) zwischengespeichert. Der Default-Wert für diesen Parameter ist 10.
- **Max Iterations:** Gibt an, wie oft über die Werte im Aggregation Buffer iteriert wird. Der Default-Wert für diesen Parameter ist 1.
- **Initialization Mode:** Gibt an, auf welche Art die Cluster-Mittelpunkte initialisiert werden:

- **Random:** Die Cluster-Mittelpunkte werden zufällig in den Grenzen gesetzt, die durch *Lower Bounds* und *Upper Bounds* gesetzt sind.
- **Euqidistant:** Die Cluster-Mittelpunkte werden äquidistant in dem Wertebereich verteilt, welcher durch die Grenzen *Lower Bounds* und *Upper Bounds* definiert ist.
- **Values:** Die Cluster-Mittelpunkte werden mit den Werten initialisiert, die durch das Array *Initial Cluster Centers* gesetzt werden.

- **Initial Cluster Centers:** Für den *Initialisierungsmodus Values* werden hier die Werte für die initialen Cluster-Mittelpunkte gesetzt. Die Werte für die einzelnen Cluster werden zeilenweise gesetzt. Das heißt die Anzahl der Zeilen der Matrix entspricht der Anzahl der Cluster *(Number of Clusters)* und die Anzahl der Spalten der Matrix der Anzahl der Eingangskanäle *(Number of Channels)*. In der ersten Zeile stehen die Werte für das erste Cluster für jeweils jeden Eingangskanal, usw.
- **Lower Bounds:** Für die Modi *Random* und *Equidistant* werden die unteren Grenzen für die einzelnen Eingangskanäle gesetzt.
- **Upper Bound:** Für die Modi *Random* und *Equidistant* werden die oberen Grenzen für die einzelnen Eingangskanäle gesetzt.

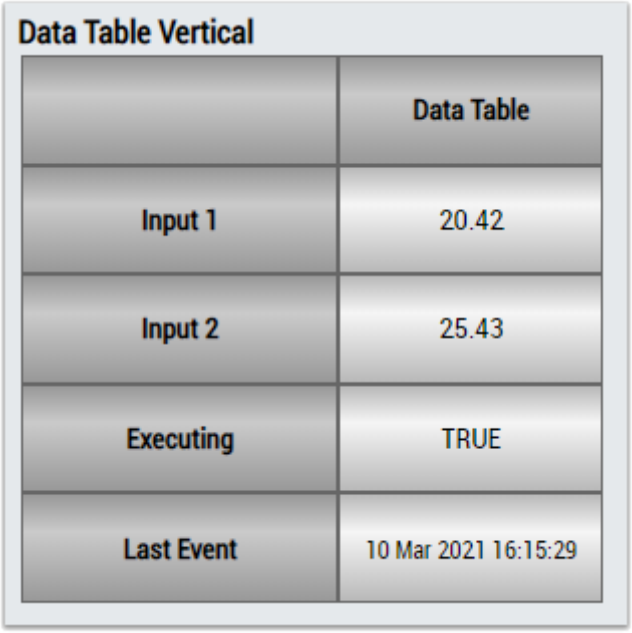
Ausgangswerte

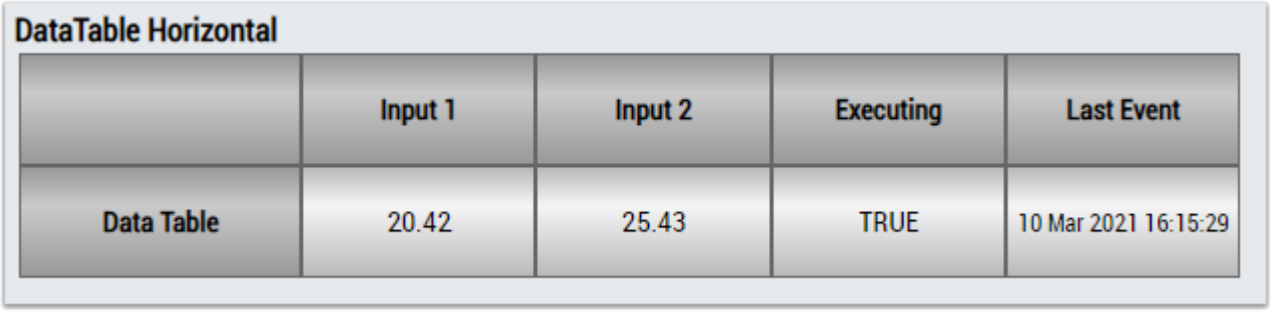
- **Cluster Index:** Gibt den Cluster-Index an, der dem Datenpunkt des letzten Zyklus zugeordnet wurde und zeigt damit das entsprechende zugeordnete Cluster an.
- **Distance:** Gibt die Euklidische Distanz zwischen dem Datenpunkt und dem zugeordneten Cluster-Mittelpunkt an.
- **Cluster Centers:** Gibt die Cluster-Mittelpunkte aller Cluster zeilenweise aus. Dies entspricht einer Matrix der Dimension *Number of Clusters* x *Number of Channels*.

Standard HMI Controls

Für den Algorithmus *Sequential k-Means* sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Table Control oder Multivalue Control visualisiert die Ausgangswerte: Lower Bounds, Upper Bounds und Initial Centers.



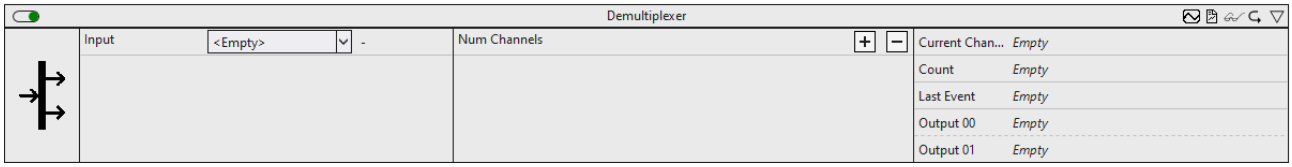


Alternativ können im Algorithmus *Sequential k-Means* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4 Analytics - Compare

Die Algorithmen der Kategorie *Analytics-Compare* bieten Funktionen für die vergleichende Analyse und logische Operationen.

6.4.4.1 Demultiplexer



Der Demultiplexer wählt einen Ausgangskanal anhand des Eingangswertes aus. Dafür wird der Eingangswert als Integer interpretiert. Dieser Wert entspricht dem Ausgangskanal. Ist der Wert Außerhalb der konfigurierten Anzahl an Kanälen, so wird der Ausgangskanal 0 gesetzt.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **NumChannels:** Fügt einen Ausgangskanal hinzu oder entfernt ihn.

Ausgangswerte

- **Current Channel:** Gibt die Nummer des ausgewählten Kanals an. Der Wert ist 0, wenn der ausgewählte Kanal außerhalb der konfigurierten Kanäle ist.

- **Count:** Beginnt mit 1 für den Kanal, der bei Beginn der Analyse ausgewählt wird, und zählt jedes Mal hoch, wenn ein anderer Kanal ausgewählt wird.
- **Last Event:** Zeitstempel des letzten Kanalwechsels.
- **Out 00..n:** Boolescher Ausgang für die Kanäle 0 (ausgewählter Kanal < 1 oder >n) bis n

Standard HMI Controls

Für den Algorithmus Demultiplexer sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

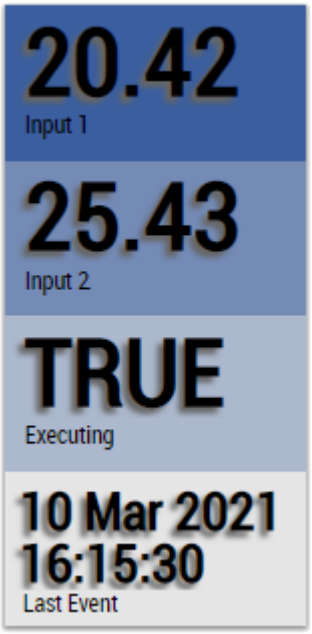
1. Das Table Control oder Multivalue Control visualisiert die Ausgangswerte Count, Current Channel und Last Event und Out.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

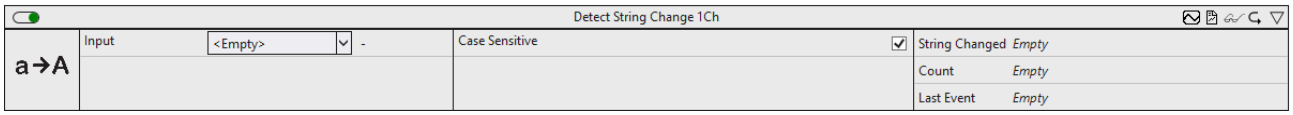
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Demultiplexer mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.2 Detect String Change 1Ch



Der *Detect String Change 1Ch* erkennt und zählt Änderungen der Stringwerte. Dabei kann die Groß- und Kleinschreibung beachtet werden oder nicht.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Case Sensitivity:** Wenn die Checkbox aktiviert ist, wird die Groß- und Kleinschreibung beachtet, anderenfalls nicht.

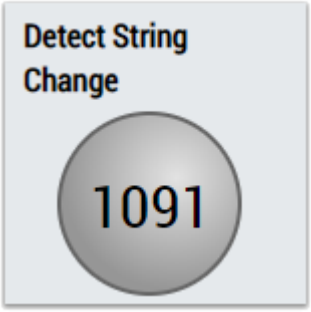
Ausgangswerte

- **String Changed:** *TRUE*, wenn eine Stringänderung erkannt wurde, ansonsten *FALSE*.
- **Count:** Zählt jedes Mal hoch, wenn Boolean Switch *TRUE* ist.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen

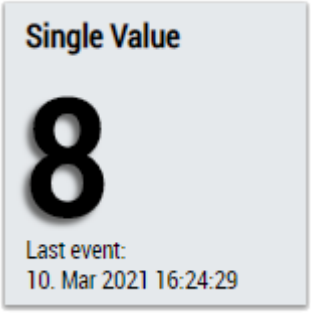
Standard HMI Controls

Für den Algorithmus Detect String Change 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Comparison Control visualisiert die Ausgangswerte Boolean Switch und Count.

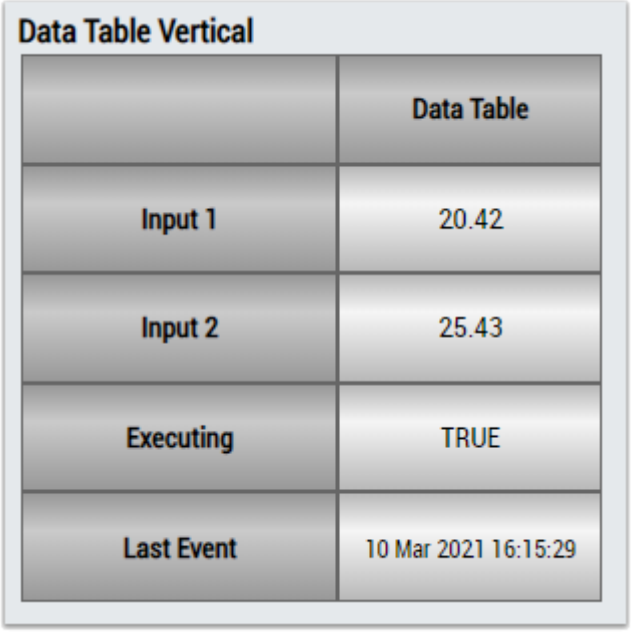


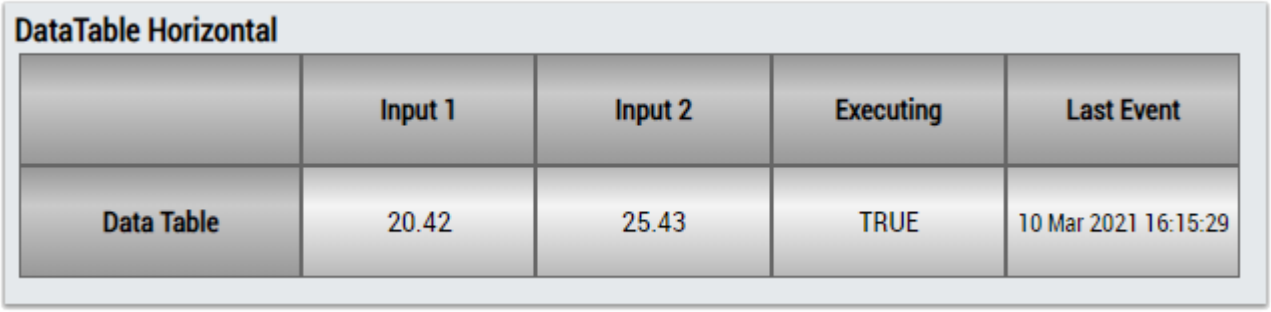
2. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.



3. Das BinaryState Control visualisiert den Ausgangswert Boolean Switch.

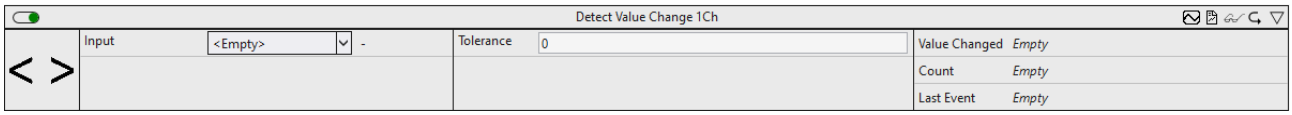
4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Operation Out, Count, Last Event.





Alternativ können im Algorithmus Detect String Change 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.3 Detect Value Change 1Ch



Der *Detect Value Change 1Ch* erkennt und zählt Änderungen der numerischen Eingangswerte.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Tolerance:** Die Toleranz bezieht sich auf den Eingangswert der zuletzt erkannten Wertänderung. Ist der Eingangswert außerhalb dieser Toleranz, wir eine Wertänderung erkannt.

Ausgangswerte

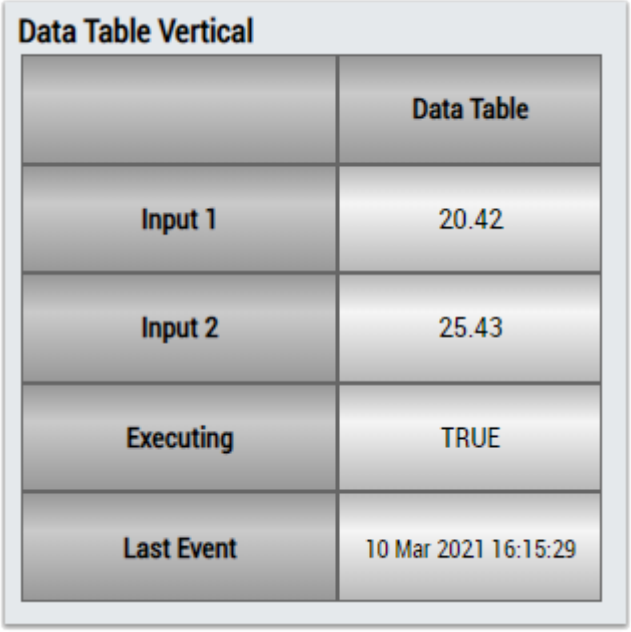
- **Value Changed:** *TRUE*, wenn eine Wertänderung erkannt wurde, ansonsten *FALSE*.
- **Count:** Zählt jedes Mal hoch, wenn Boolean Switch *TRUE* ist.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen

Standard HMI Controls

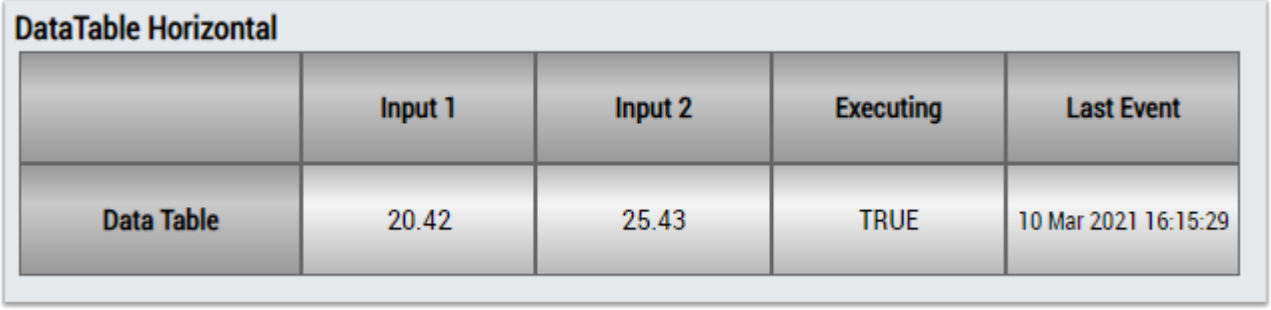
Für den Algorithmus *Detect Value Change 1Ch* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.

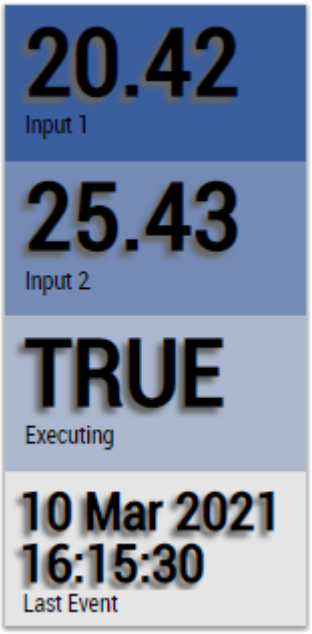
2. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte.



	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

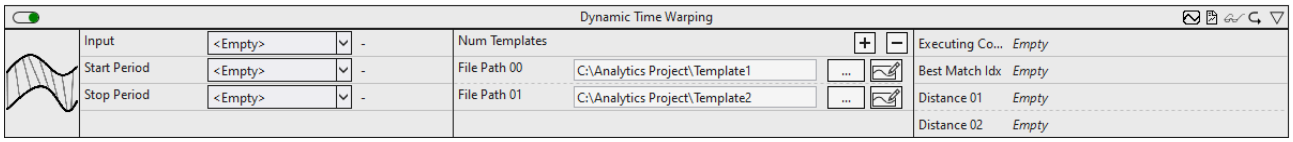


	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.4 Dynamic Time Warping



Der Algorithmus *Dynamic Time Warping* vergleicht Eingangsdaten mit zuvor aufgenommenen Templates. Das Besondere an dem Algorithmus ist, dass auch Signale mit unterschiedlicher Geschwindigkeit oder aber auch verschobene Signale verglichen werden können. Als Ergebnis wird die Distanz zwischen dem Eingangssignal und dem jeweiligen Template ausgegeben. Je geringer die Distanz, desto gleicher sind die verglichenen Signale. Ist die Distanz 0, so sind beide Signale identisch. Die Höhe der Distanz ist abhängig von der Gleichheit aber auch von der Länge der Signale.

Der Vergleich beginnt, wenn das Signal des Flags Start Period *TRUE* ist. Ein Ergebnis wird ausgegeben, wenn das Signal des Flags Stop Period TRUE oder das Flag Start Period erneut TRUE ist.

Es wird empfohlen, *Dynamic time Warping* aufgrund des konkurrierenden Dateizugriffs nicht gleichzeitig mit Time Based Teach Path 1Ch [▶ 259] zu verwenden. Stattdessen sollte zunächst ein Referenzsignal mit dem Time Based Teach Path 1Ch [▶ 259] eingelernt werden und erst im Anschluss die Auswertung mithilfe des *Dynamic time Warping* erfolgen. Die Templates enthalten Referenzsignale, die zuvor mit dem Time Based Teach Path 1Ch [▶ 259] aufgenommen wurden. In der Regel handelt es sich bei den Templates um wenige Hundert Stützpunkte. Daher ist eine Reduktion der Daten mit dem Baustein Downsampling 1Ch [▶ 88] oft sinnvoll.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Templates:** Fügt ein Template für den Vergleich hinzu oder entfernt es.
- **File Path 01 .. n:** Pfad zu dem zuvor aufgezeichneten Template.

Ausgangswerte

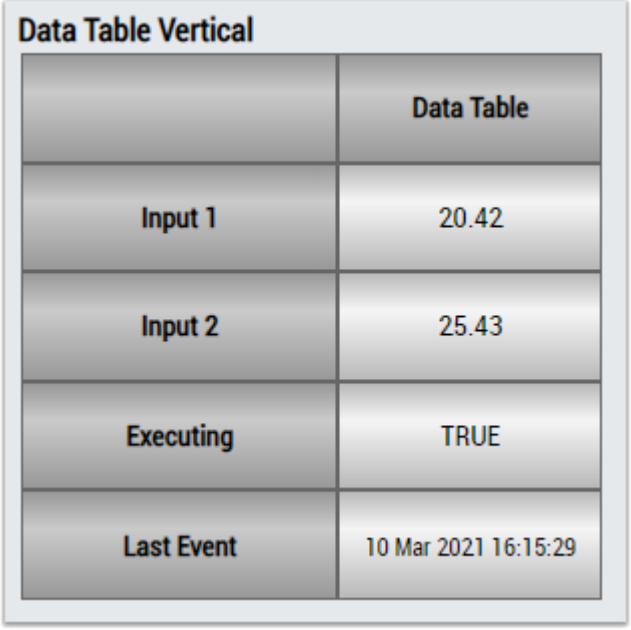
- **Executing Compare:** Zeigt *TRUE* an, wenn der Algorithmus die eingehenden Werte verarbeitet, ansonsten *FALSE*. Der Verarbeitungsprozess beginnt, wenn das Flag Start Period *TRUE* ist und endet, wenn Stop Period TRUE ist.
- **Best Match Idx:** Gibt den Index des Templates mit der geringsten Distanz zum Eingangskanal aus.

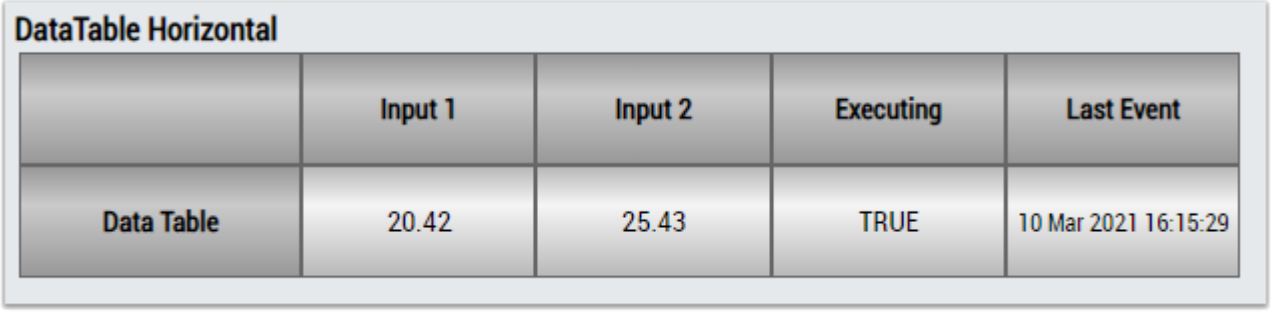
- **Distance 01 .. n:** Gibt die Distanz zwischen dem Eingangssignal und dem jeweiligen Template an. Je geringer die Distanz, desto gleicher sind die verglichenen Signale.

Standard HMI Controls

Für den Algorithmus *Dynamic Time Warping* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

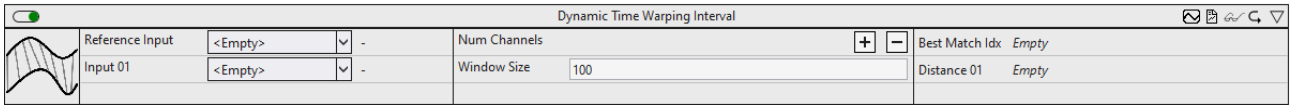
1. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte: *Executing Compare*, *Best Match Idx*, *Distance*.





Alternativ können im Algorithmus *Dynamic Time Warping* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.5 Dynamic Time Warping Interval



Der Algorithmus *Dynamic Time Warping Interval* vergleicht mehrere Eingangsdaten miteinander. Das Besondere an dem Algorithmus ist, dass auch Signale mit unterschiedlicher Geschwindigkeit oder aber auch verschobene Signale verglichen werden können. Für den Vergleich wird nur das Signalintervall eines konfigurierten Fensters berücksichtigt. Neue Ergebnisse werden nach dem Ablauf des Fensters ausgegeben. Als Ergebnis wird die Distanz zwischen dem Referenzsignal und dem jeweiligen Eingangssignal ausgegeben. Je geringer die Distanz, desto gleicher sind die verglichenen Signale. Ist die Distanz 0, so sind beide Signale identisch. Die Höhe der Distanz ist abhängig von der Gleichheit aber auch von der Länge der Signale.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Fügt einen Eingangskanal hinzu oder entfernt ihn.
- **Window Size:**Gibt die Anzahl an Zyklen an, über die eine Berechnung erfolgt. Der Speicherbedarf des Algorithmus ist proportional zu diesem Parameter.

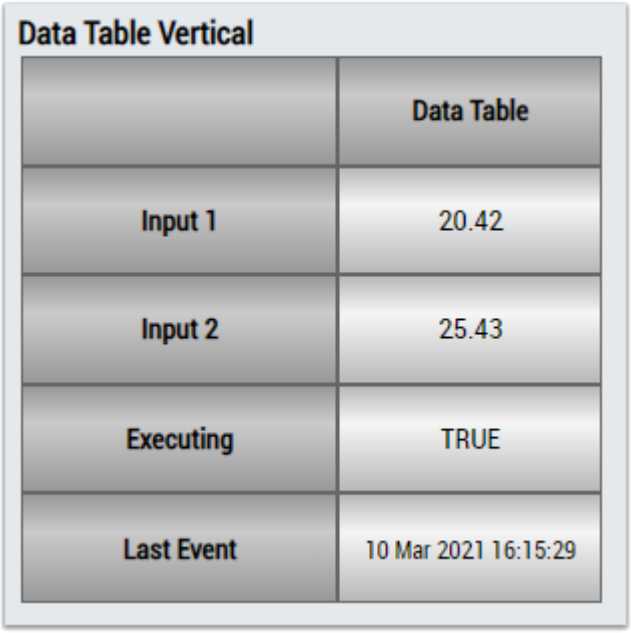
Ausgangswerte

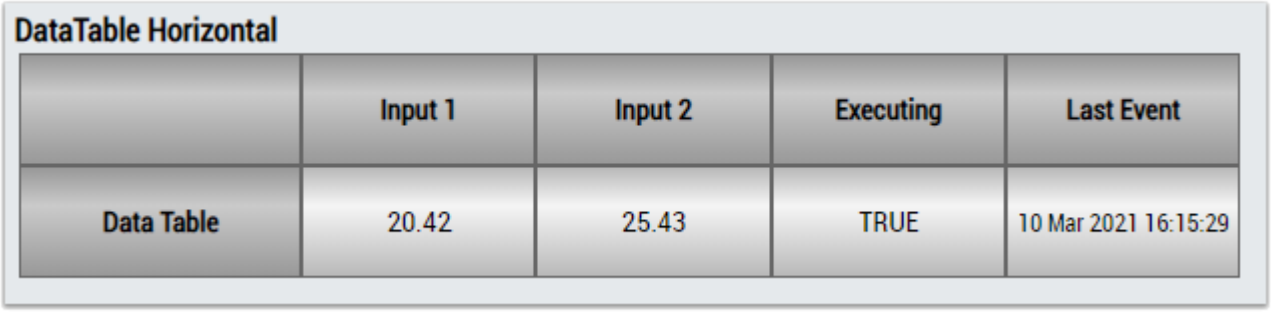
- **Best Match Idx:** Gibt den Index des Eingangskanals mit der geringsten Distanz zum Referenzkanal aus.
- **Distance 01 .. n:** Gibt die Distanz zwischen dem Referenzsignal und dem jeweiligen Eingangskanal an. Je geringer die Distanz, desto gleicher sind die verglichenen Signale.

Standard HMI Controls

Für den Algorithmus *Dynamic Time Warping Interval* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

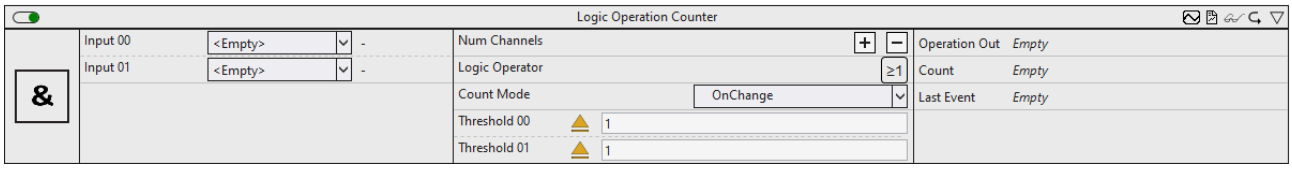
1. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte: *Best Match Idx*, *Distance*.





Alternativ können im Algorithmus *Dynamic Time Warping Interval* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.6 Logic Operation Counter



Der *Logic Operation Counter* führt eine logische Operation an den Werten von zwei oder mehr Kanälen aus und liefert das Ergebnis dieser logischen Operation. Dazu kann jeder Eingangswert mit einer Schwelle und einem Operator kombiniert werden. Des Weiteren können der logische Operator und der Zählmodus einzeln konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Fügt einen Eingangskanal hinzu oder entfernt ihn.
- **Threshold 00:** Schwelle für das Signal des ersten Kanals.
- **Threshold 01 ..n:** Schwelle für das Signal des zweiten bis n-ten Kanals.
- **Logic Operator:** Logischer Operator für die Operation:

  ≥1 Logisches ODER

Logisches XOR (EXKLUSIVES ODER)

Logisches UND

Logisches NAND (NICHT UND)

Logisches NOR (NICHT ODER)

- **Count Mode:** Modus des Ergebniszählers.
  OnChange: Der Zähler zählt jedes Mal, wenn sich das Ergebnis auf TRUE ändert.
  Cyclic: Der Zähler inkrementiert jeden Zyklus, wenn die Bedingung *TRUE* ist.
- **Use Absolute Values:** Wenn die Checkbox aktiviert ist, werden die absoluten Werte des Eingangssignals verwendet.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

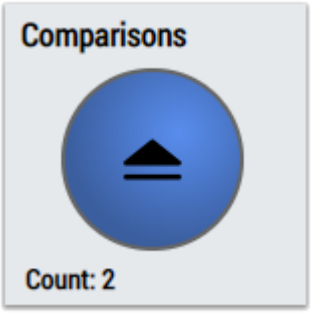
Ausgangswerte

- **Operation Out:** Ergebnis der logischen Operation.
- **Count:** Wird inkrementiert, wenn der Ausgangswert TRUE ist. Das Verhalten ist abhängig von dem Parameter *Count Mode*.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen

Standard HMI Controls

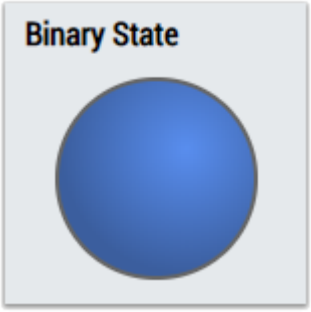
Für den Algorithmus Logic Operation Counter sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Comparison Control visualisiert die Ausgangswerte Operation Out und Count sowie die Konfigurationsoption Logic Operator.

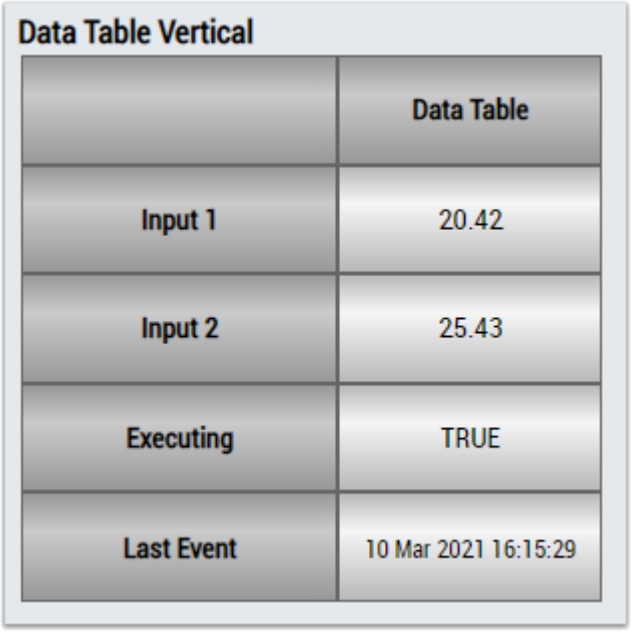


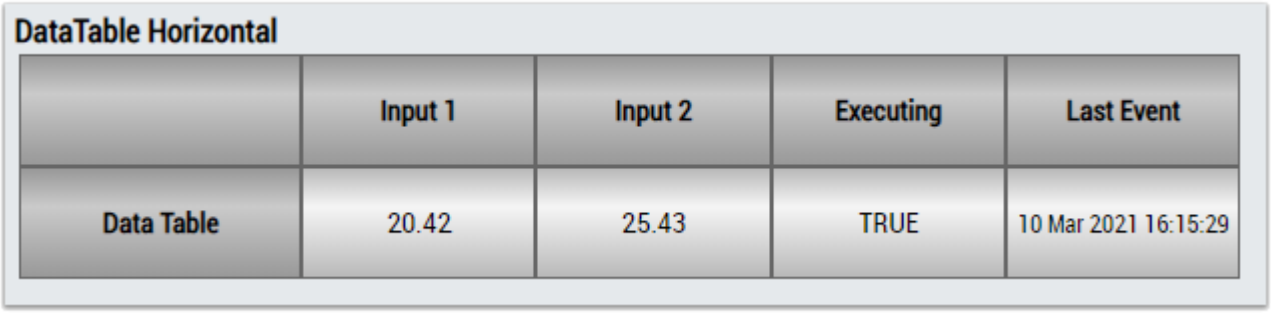
2. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.

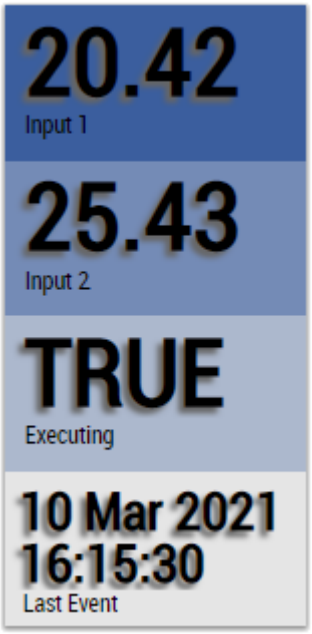
3. Das BinaryState Control visualisiert den Ausgangswert Operation Out.



4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Operation Out, Count, Last Event.

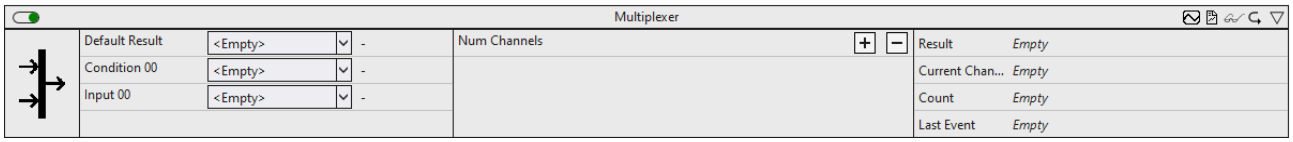






Alternativ können im Algorithmus Logic Operation Counter mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.7 Multiplexer



Der Multiplexer wählt einen Kanal aus einem oder mehreren Eingangskanälen aus. Für jeden Eingangskanal muss zusätzlich ein boolscher Eingang bereitgestellt werden. Der Ausgang entspricht dem ersten Eingangskanal, wenn der bedingte Eingang TRUE ist. Die Priorität der konfigurierten Kanäle ist die Reihenfolge ihrer Konfiguration. Wenn die Bedingung für keinen der Kanäle erfüllt ist, wird der bereitgestellte Standardkanal zurückgegeben.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Kanäle. Für jeden Kanal gibt es zwei Eingangsvariablen, eine ist eine Bedingung vom Typ Boolean (*Condition0n*) und ein Eingangswert eines beliebigen Datentyps (*Input0n*), der an *Result* weitergegeben werden kann, wenn die Bedingung erfüllt ist.

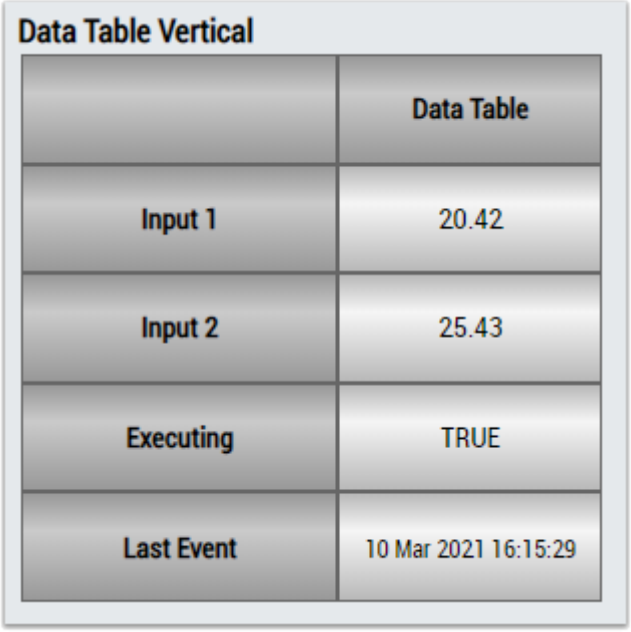
Ausgangswerte

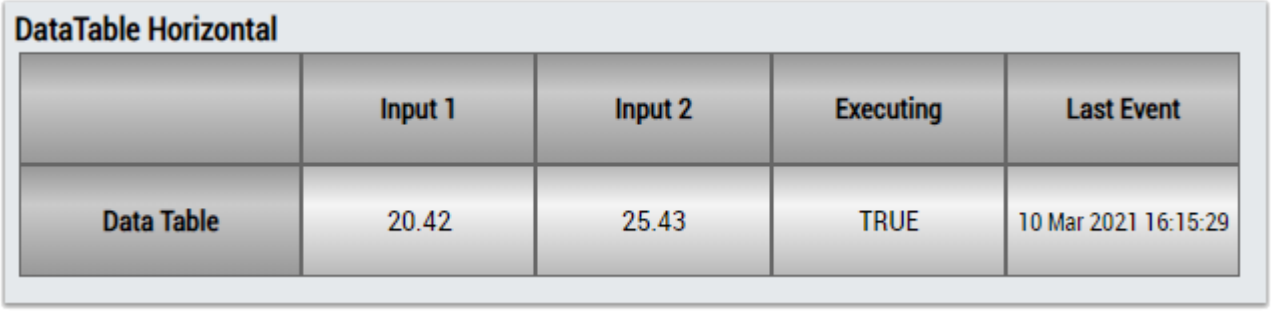
- **Result:** Liefert das Signal *Input0n* des ausgewählten Eingangskanals.
- **Current Channel:** Gibt die Nummer des ausgewählten Kanals an. Der Wert ist 0, wenn das Standardergebnis ausgewählt ist. Die Eingangskanäle werden in der Reihenfolge ihrer Konfiguration nummeriert.
- **Count:** Beginnt mit 1 für den Kanal, der bei Beginn der Analyse ausgewählt wird, und zählt jedes Mal hoch, wenn ein anderer Kanal ausgewählt wird.
- **Last Event:** Zeitstempel des letzten Kanalwechsels.

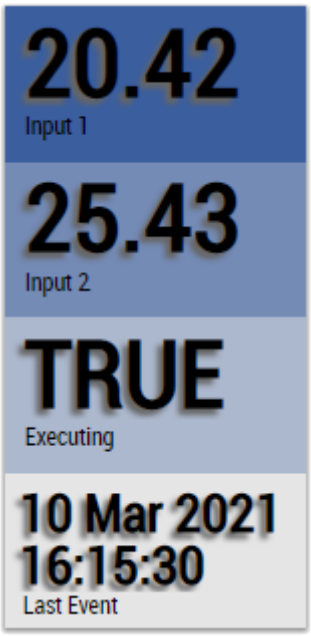
Standard HMI Controls

Für den Algorithmus Multiplexer sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Table Control oder Multivalue Control visualisiert die Ausgangswerte Count, Result, Current Channel und Last Event.

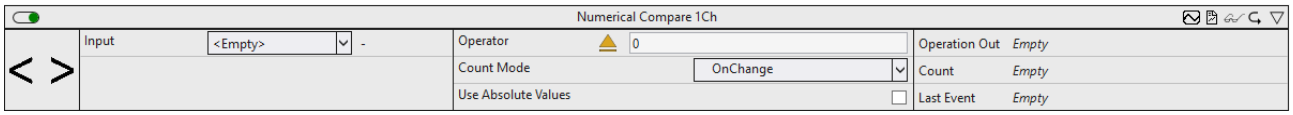






Alternativ können im Algorithmus Multiplexer mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.8 Numerical Compare 1Ch



Der *Numerical Compare 1Ch* vergleicht die Eingangswerte mit einem Referenzwert und liefert das Ergebnis dieser Vergleichsoperation. Der Operator, der Referenzwert und der Zählmodus können einzeln konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Operator:** Gibt an, ob der Eingangswert größer, größer oder gleich, gleich, kleiner oder gleich, kleiner oder ungleich dem Referenzwert sein soll.
- **Reference:** Referenzwert für die Vergleichsoperation.
- **Count Mode:** Modus des Ergebniszählers.
  OnChange: Der Zähler zählt jedes Mal, wenn sich das Ergebnis auf TRUE ändert.
  Cyclic: Der Zähler inkrementiert jeden Zyklus, wenn die Bedingung *TRUE* ist.
- **Use Absolute Values:** Wenn die Checkbox aktiviert ist, wird der absolute Wert des Eingangssignals verwendet.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

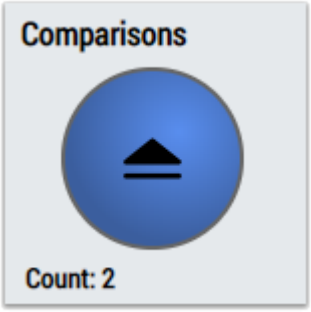
Ausgangswerte

- **Operation Out:** Ergebnis der Vergleichsoperation.
- **Count:** Wird inkrementiert, wenn der Ausgangswert TRUE ist. Das Verhalten ist abhängig von dem Parameter *Count Mode*.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen

Standard HMI Controls

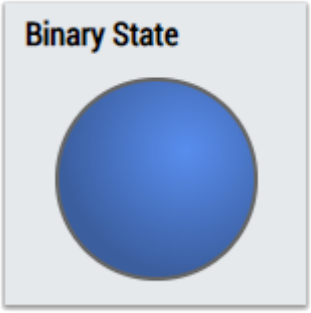
Für den Algorithmus Numerical Compare 1Ch stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das Comparison Control oder Multivalue Control visualisiert die Ausgangswerte Operation Out und Count sowie die Konfigurationsoption Operator.



2. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.

3. Das BinaryState Control visualisiert den Ausgangswert Operation Out.



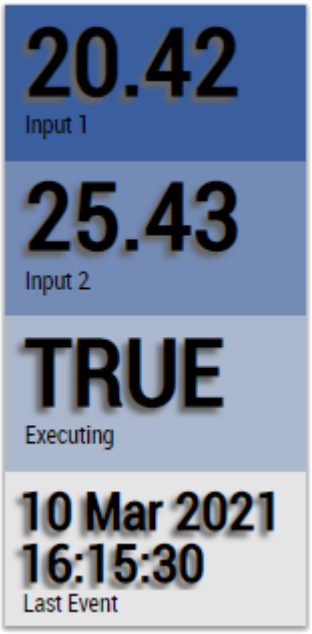
4. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte: Operation Out, Count, Last Event.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

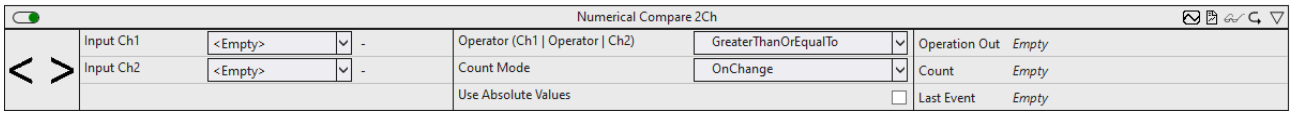
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus *Numerical Compare 1Ch* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.9 Numerical Compare 2Ch



Numerical Compare 2Ch vergleicht die Eingangswerte des ersten Kanals mit den Eingangswerten des zweiten Kanals und liefert das Ergebnis dieser Vergleichsoperation. Der Operator und der Zählmodus können einzeln konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Operator:** Gibt an, ob der Eingangswert größer, größer oder gleich, gleich, kleiner oder gleich, kleiner oder ungleich dem Referenzwert sein soll.
- **Count Mode:** Modus des Ergebniszählers.
  OnChange: Der Zähler zählt jedes Mal, wenn sich das Ergebnis auf TRUE ändert.
  Cyclic: Der Zähler inkrementiert jeden Zyklus, wenn die Bedingung *TRUE* ist.
- **Use absolute values:** Wenn die Checkbox aktiviert ist, werden die absoluten Werte des Eingangssignals verwendet.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.

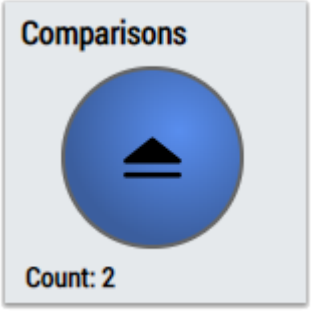
Ausgangswerte

- **Operation Out:** Ergebnis der Vergleichsoperation.
- **Count:** Wird inkrementiert, wenn der Ausgangswert TRUE ist. Das Verhalten ist abhängig von dem Parameter *Count Mode*.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen

Standard HMI Controls

Für den Algorithmus Numerical Compare 1Ch stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

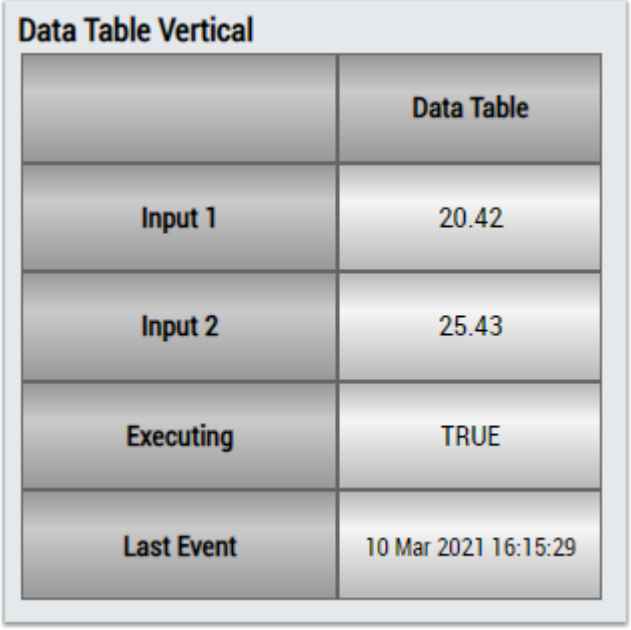
1. Das Comparison Control oder Multivalue Control visualisiert die Ausgangswerte Operation Out und Count sowie die Konfigurationsoption Operator.



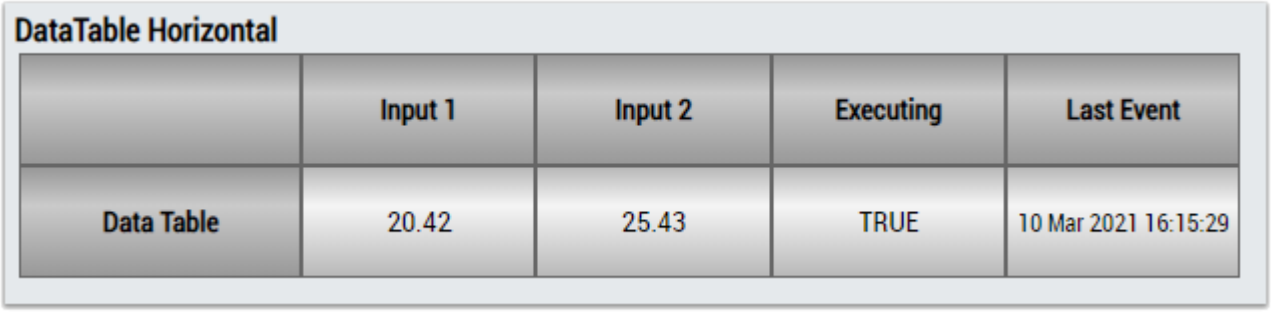
2. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.

3. Das BinaryState Control visualisiert den Ausgangswert Operation Out.

4. Das Table Control visualisiert alle Ausgangswerte: Operation Out, Count, Last Event.

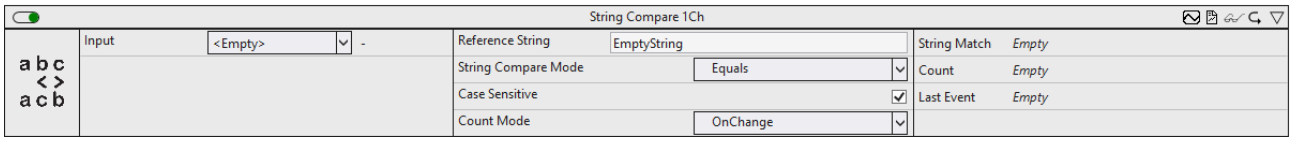


	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Numerical Compare 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.10 String Compare 1Ch



Der *String Compare 1Ch* vergleicht den Eingangsstring mit einem Referenzstring und zählt die Stringübereinstimmungen. Dabei kann die Groß- und Kleinschreibung beachtet werden oder nicht und der Zählmodus geändert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Reference String:** Referenzstring für die Vergleichsoperation.
- **String Compare Mode**: Aufzählung für verschiedene Stringvergleichsarten.
  Equals: Eingangsstring entspricht dem Referenzstring.
  BeginsWith: Eingangsstring beginnt mit dem Referenzstring.
  Contains: Eingangsstring enthält den Referenzstring.
- **Case Sensitivity:** Wenn die Checkbox aktiviert ist, wird die Groß- und Kleinschreibung beachtet, anderenfalls nicht.
- **Count Mode:** Modus des Ergebniszählers.
  OnChange: Der Zähler zählt jedes Mal, wenn sich das Ergebnis auf TRUE ändert.
  Cyclic: Der Zähler inkrementiert jeden Zyklus, wenn die Bedingung *TRUE* ist.

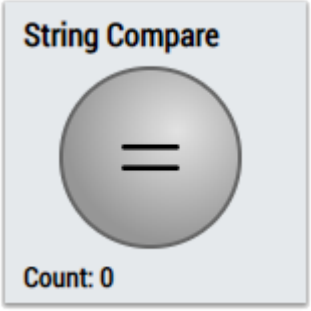
Ausgangswerte

- **String Match:** *TRUE*, wenn der Eingangsstring mit dem Referenzstring übereinstimmt, ansonsten *FALSE*.
- **Count:** Wird inkrementiert, wenn der Ausgangswert TRUE ist. Das Verhalten ist abhängig von dem Parameter *Count Mode*.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen

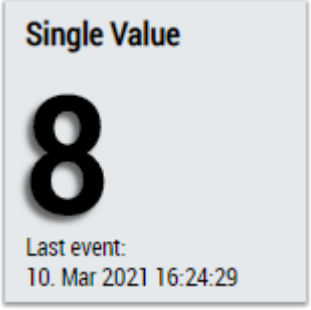
Standard HMI Controls

Für den Algorithmus String Compare 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

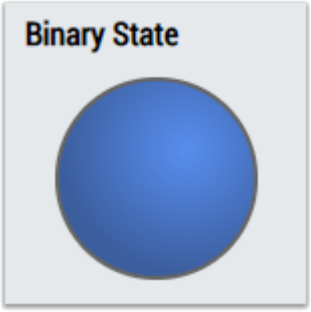
1. Das Comparison Control visualisiert die Ausgangswerte String Match und Count sowie die Konfigurationsoption String Compare Mode.



2. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.



3. Das BinaryState Control visualisiert den Ausgangswert String Match.



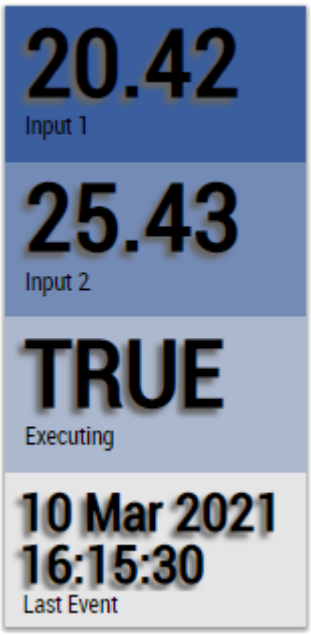
4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: String Match, Count, Last Event.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

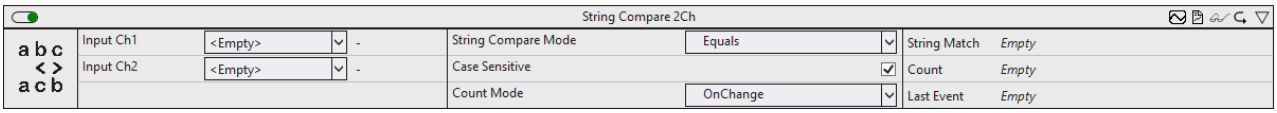
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus String Compare 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.11 String Compare 2Ch



Der *String Compare 2Ch* vergleicht die Werte des ersten Eingangsstrings mit den Werten des zweiten Strings und zählt die Stringübereinstimmungen. Dabei kann die Groß- und Kleinschreibung beachtet werden oder nicht und der Zählmodus geändert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **String Compare Mode**: Aufzählung für verschiedene Stringvergleichsarten.
  Equals: Eingangsstring entspricht dem Referenzstring.
  BeginsWith: Eingangsstring beginnt mit dem Referenzstring.
  Contains: Eingangsstring enthält den Referenzstring.
- **Case Sensitivity:** Wenn die Checkbox aktiviert ist, wird die Groß- und Kleinschreibung beachtet, anderenfalls nicht.
- **Count Mode:** Modus des Ergebniszählers.
  OnChange: Der Zähler zählt jedes Mal, wenn sich das Ergebnis auf TRUE ändert.
  Cyclic: Der Zähler inkrementiert jeden Zyklus, wenn die Bedingung *TRUE* ist.

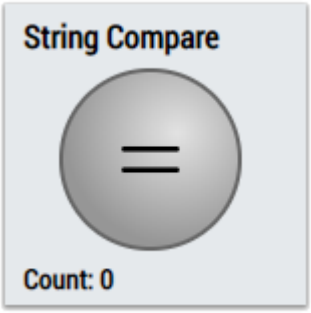
Ausgangswerte

- **String Match:** *TRUE*, wenn der Wert des ersten Eingangsstrings mit dem Wert des zweiten Eingangsstrings übereinstimmt, ansonsten *FALSE*.
- **Count:** Wird inkrementiert, wenn der Ausgangswert TRUE ist. Das Verhalten ist abhängig von dem Parameter *Count Mode*.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen

Standard HMI Controls

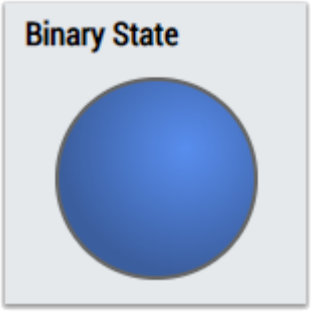
Für den Algorithmus String Compare 2Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Comparison Control visualisiert die Ausgangswerte String Match und Count sowie die Konfigurationsoption String Compare Mode.

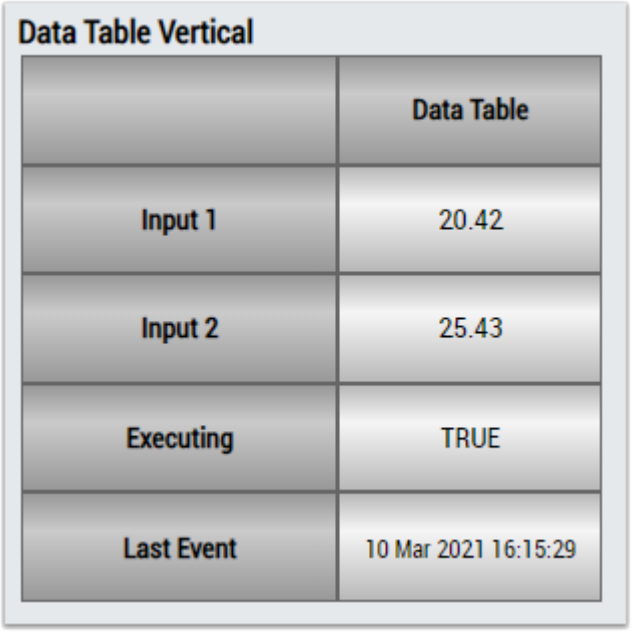


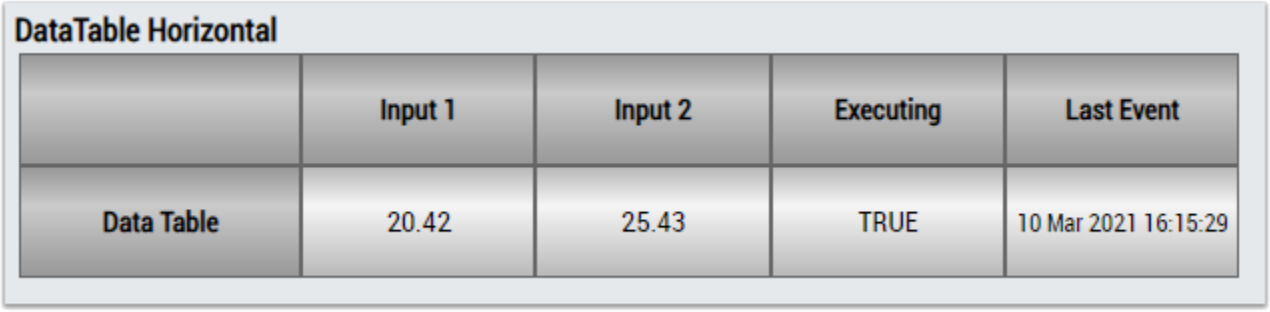
2. Das SingleValue Control visualisiert die Ausgangswerte Count und Last Event.

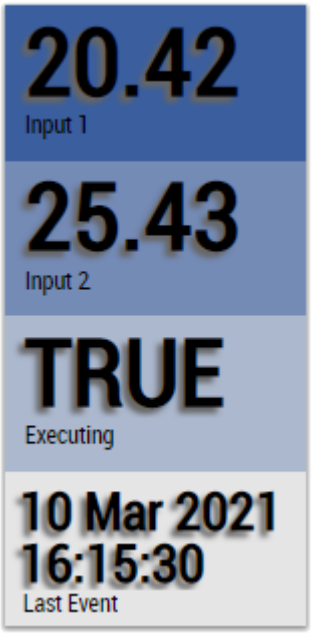
3. Das BinaryState Control visualisiert den Ausgangswert String Match.



4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: String Match, Count, Last Event.

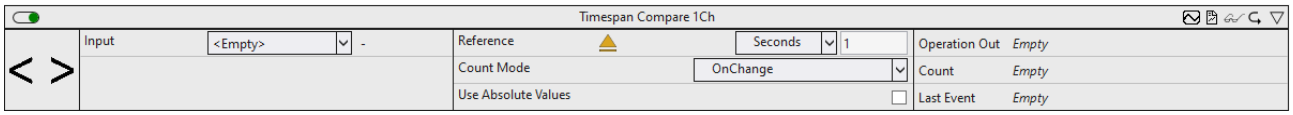






Alternativ können im Algorithmus String Compare 2Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.4.12 Timespan Compare 1Ch



Der *Timespan Compare 1Ch* vergleicht den Eingangswert mit einem Referenzwert und liefert das Ergebnis dieser Vergleichsoperation. Als Eingang wird eine Zeitspanne mit einer Auflösung von 1 ns unterstützt. Der Operator, der Referenzwert und der Zählmodus können einzeln konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Operator:** Gibt an, ob der Eingangswert größer, größer oder gleich, gleich, kleiner oder gleich, kleiner oder ungleich dem Referenzwert sein soll.
- **Reference:** Referenzzeitspanne für die Vergleichsoperation.
- **Count Mode:** Modus des Ergebniszählers.
  OnChange: Der Zähler zählt jedes Mal, wenn sich das Ergebnis auf *TRUE* ändert.
  Cyclic: Der Zähler inkrementiert jeden Zyklus, wenn die Bedingung *TRUE* ist.
- **Use Absolute Values:** Wenn die Checkbox aktiviert ist, wird der absolute Wert des Eingangssignals verwendet.

Ausgangswerte

- **Operation Out:** Ergebnis der Vergleichsoperation.
- **Count:** Wird inkrementiert, wenn der Ausgangswert *TRUE* ist. Das Verhalten ist abhängig von dem Parameter *Count Mode*.
- **Last Event:** Gibt den Zeitpunkt des letzten ausgelösten Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen

Standard HMI Controls

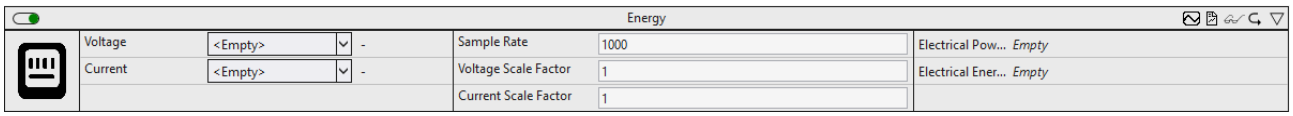
Für den Algorithmus Timespan Compare 1Ch stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das Table Control visualisiert alle Ausgangswerte: Operation Out, Count, Last Event.

Alternativ können im Algorithmus *Timespan Compare 1Ch* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.5 Analytics - Energy

6.4.5.1 Energy



Das *Energy*-Modul eignet sich für die Energiebetrachtung verschiedener elektrischer Komponenten. Die Eingangsdaten von Strom und Spannung werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Voltage Scale Factor:** Skalierungsfaktor der Eingangsspannung.
- **Current Scale Factor:** Skalierungsfaktor des Eingangsstroms.
- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

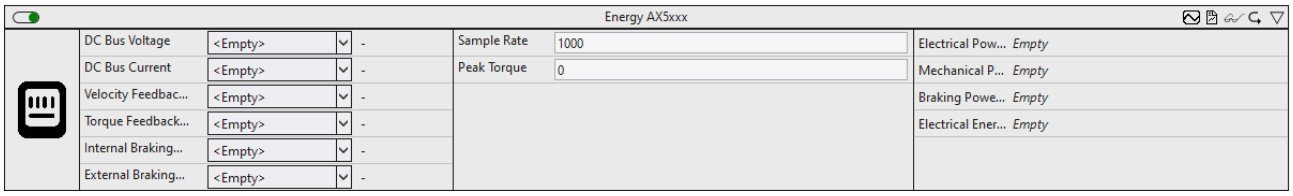
Eingangswerte

- **Voltage:** RMS-Wert der Spannung in V.
- **Current:** RMS-Wert des Stroms in A.

Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.5.2 Energy AX5xxx



Das *Energy AX5xxx*-Modul eignet sich für die Energiebetrachtung von Servoverstärkern der Baureihe AX5xxx. Die Eingangsdaten werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet. Damit die benötigten Eingangsdaten zur Verfügung stehen, müssen diese im Prozessabbild hinzugefügt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Peak Torque:** Spitzenwert des Drehmoments in Nm
  Prozessdaten: CoE: P-0-0094

- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

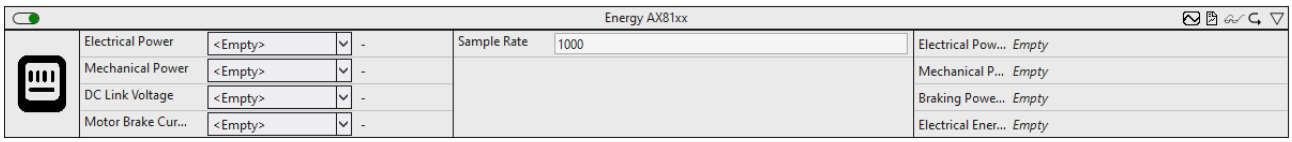
Eingangswerte

- **DC Bus Voltage:** Zwischenkreisspannung in 0,1 V.
  Prozessdaten: CoE: S-0-0016. Fügen Sie das CoE S-0-0380 "DC bus voltage" zum PDO-Inhalt hinzu.
- **DC Bus Current:** Eingangsstrom in den Zwischenkreis in mA.
  Prozessdaten: CoE: S-0-0016. Fügen Sie das CoE S-0-0381 "DC bus current" zum PDO-Inhalt hinzu.
- **Velocity Feedback Value:** Motordrezahl.
  Prozessdaten: CoE: S-0-0016. Fügen Sie das CoE S-0-0040 "Velocity feedback 1 value" to PDO content) zum PDO-Inhalt hinzu.
- **Torque Feedback Value:** Drehmomentbildender Strom.
  Prozessdaten CoE: S-0-0016. Fügen Sie das CoE S-0-0084 "Torque/force feedback value" zum PDO-Inhalt hinzu.
- **Internal Braking Resistor Power:** Über den internen Bremswiderstand abgegebene Bremsleistung in W.
  Prozessdaten: CoE: S-0-0016. Fügen Sie das CoE P-0-0209 "Internal braking resistor actual averaged power" zum PDO-Inhalt hinzu.
- **External Braking Resistor Power:** Über den externen Bremswiderstand abgegebene Bremsleistung in W.
  Prozessdaten: CoE: S-0-0016. Fügen Sie das CoE P-0-0210 "External braking resistor actual averaged power" zum PDO-Inhalt hinzu.

Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Mechanical Power [W]:** Mechanische Leistung in W.
- **Braking Power [W]:** Bremsleistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.5.3 Energy AX81xx



Das *Energy AX81xx*-Modul eignet sich für die Energiebetrachtung von Servoverstärkern der Baureihe AX81xx. Die Eingangsdaten werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet. Damit die benötigten Eingangsdaten zur Verfügung stehen, müssen diese im Prozessabbild hinzugefügt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

Eingangswerte

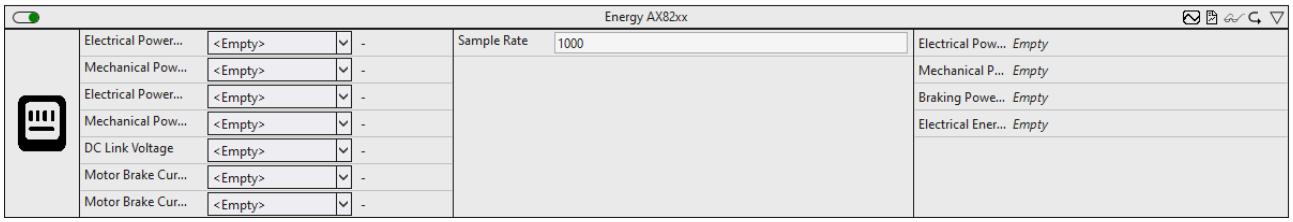
- **Electrical Power:** Elektrische Leistung in W.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A32 und fügen Sie den CoE 0x3242:12 "Electrical power" zum PDO-Inhalt hinzu.
- **Mechanical Power:** Mechanische Leistung in W.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A32 und fügen Sie den CoE 0x3242:13 " Mechanical power" zum PDO-Inhalt hinzu.

- **DC Link Voltage:** Zwischenkreisspannung in mV.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A00 und fügen Sie den CoE 0x6079 "DC link circuit voltage" zum PDO-Inhalt hinzu.
- **Motor Brake Current:** Bremsstrom des Motors in A.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A20 und fügen Sie den CoE 0x3001:01 "Actual motor brake current" zum PDO-Inhalt hinzu.

Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Mechanical Power [W]:** Mechanische Leistung in W.
- **Braking Power [W]:** Bremsleistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.5.4 Energy AX82xx



Das *Energy AX82xx*-Modul eignet sich für die Energiebetrachtung von Servoverstärkern der Baureihe AX82xx. Die Eingangsdaten werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet. Damit die benötigten Eingangsdaten zur Verfügung stehen, müssen diese im Prozessabbild hinzugefügt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

Eingangswerte

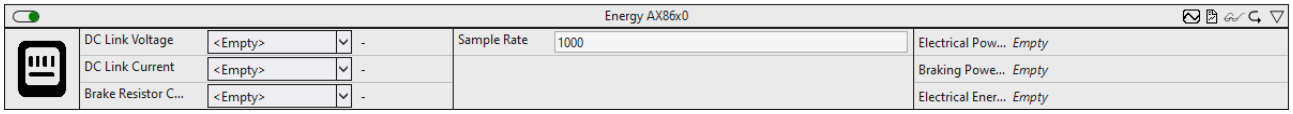
- **Electrical Power ChA:** Elektrische Leistung des ersten Kanals in W.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A32 und fügen Sie den CoE 0x3242:12 "Electrical power" zum PDO-Inhalt hinzu.
- **Mechanical Power ChA:** Mechanische Leistung des ersten Kanals in W.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A32 und fügen Sie den CoE 0x3242:13 " Mechanical power" zum PDO-Inhalt hinzu.
- **Electrical Power ChB:** Elektrische Leistung des zweiten Kanals in W.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A52 und fügen Sie den CoE 0x3642:12 "Electrical power" zum PDO-Inhalt hinzu.
- **Mechanical Power ChB:** Mechanische Leistung des zweiten Kanals in W.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A52 und fügen Sie den CoE 0x3642:13 " Mechanical power" zum PDO-Inhalt hinzu.
- **DC Link Voltage:** Zwischenkreisspannung in mV.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A00 und fügen Sie den CoE 0x6079 "DC link circuit voltage" zum PDO-Inhalt hinzu.
- **Motor Brake Current ChA:** Bremsstrom des Motors des ersten Kanals in A.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A20 und fügen Sie den CoE 0x3001:01 "Actual motor brake current" zum PDO-Inhalt hinzu.

- **Motor Brake Current ChB:** Bremsstrom des Motors des zweiten Kanals in A.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A40 und fügen Sie den CoE 0x3401:01 "Actual motor brake current" zum PDO-Inhalt hinzu.

Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Mechanical Power [W]:** Mechanische Leistung in W.
- **Braking Power [W]:** Bremsleistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.5.5 Energy AX86x0



Das *Energy AX86xx*-Modul eignet sich für die Energiebetrachtung von Einspeisemodulen der Baureihe AX86xx. Die Eingangsdaten werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet. Damit die benötigten Eingangsdaten zur Verfügung stehen, müssen diese im Prozessabbild hinzugefügt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

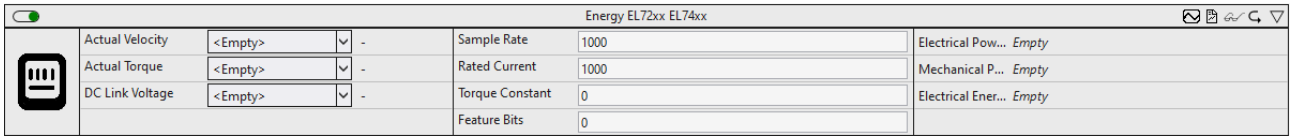
Eingangswerte

- **DC Link Voltage:** Zwischenkreisspannung in mV.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A05.
- **DC Link Current:** Eingangsstrom in den Zwischenkreis in 0.01 A.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A6.
- **Brake Resistor Continuous Power:** Über den Bremswiderstand abgegebene Leistung in W.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A07.

Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Braking Power [W]:** Bremsleistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.5.6 Energy EL72xx EL74xx



Das *Energy AX72xx EL74xx*-Modul eignet sich für die Energiebetrachtung von EtherCAT-Klemmen der Typen EL72xx und EL74xx. Die Eingangsdaten werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet. Damit die benötigten Eingangsdaten zur Verfügung stehen, müssen diese im Prozessabbild hinzugefügt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Rated Current:** Nennstrom in mA.
  Prozessdaten: CoE: 0x8011:12.
- **Torque Constant:** Drehmomentkonstante in mNm / A.
  Prozessdaten: CoE: 0x8011:16.
- **Feature Bits:** Feature-Bits zur Angabe des Stromtyps.
  Prozessdaten: CoE: 0x8010:53.
- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

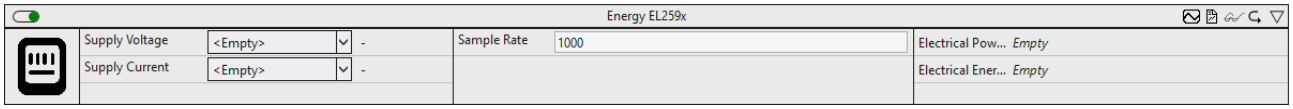
Eingangswerte

- **Actual Velocity:** Istwert der Geschwindigkeit.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A02.
- **Actual Torque:** Istwert des Drehmoments.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A03.
- **DC Link Voltage:** Zwischenkreisspannung in mV.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A04 und wählen Sie "DC link voltage (mV)" im CoE 0x8010:39 aus.

Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Mechanical Power [W]:** Mechanische Leistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.5.7 Energy EL259x



Das *Energy EL259x*-Modul eignet sich für die Energiebetrachtung von EtherCAT-Klemmen des Typs EL259x. Die Eingangsdaten werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet. Damit die benötigten Eingangsdaten zur Verfügung stehen, müssen diese im Prozessabbild hinzugefügt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

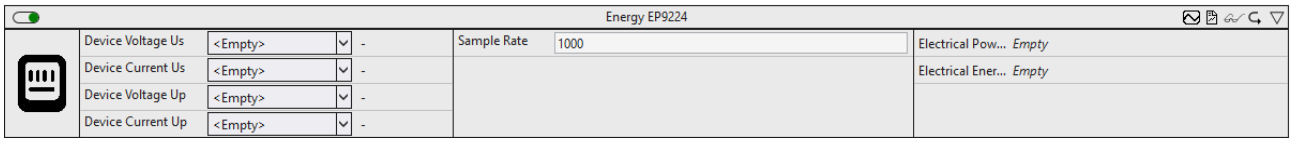
Eingangswerte

- **Supply Voltage:** Versorgungsspannung in 0,01 V.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A01 und wählen Sie CoE 0xF900:05 "Supply voltage" im CoE 0x8002:11 aus.
- **Supply Current:** Eingangsstrom in mA.
  Prozessdaten: Aktivieren Sie den PDO-Eingang 0x1A01 und wählen Sie CoE 0xF900:06 "Supply current" im CoE 0x8002:19 aus.

Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.5.8 Energy EP9224



Das *Energy EP9224*-Modul eignet sich für die Energiebetrachtung von EtherCAT-Boxen des Typs EP9224. Die Eingangsdaten werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet. Damit die benötigten Eingangsdaten zur Verfügung stehen, müssen diese im Prozessabbild hinzugefügt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

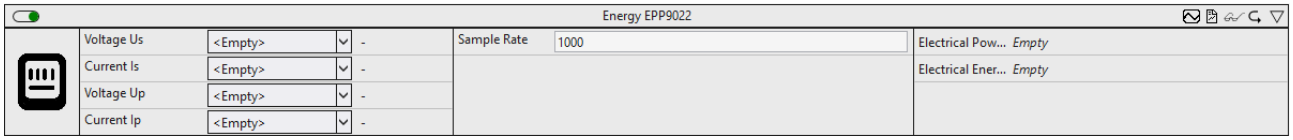
Eingangswerte

- **Device Voltage Us:** Steuerspannung in V.
  Prozessdaten: PDO Inputs Device
- **Device Current Us:** Steuerstrom in A.
  Prozessdaten: PDO Inputs Device
- **Device Voltage Up:** Lastspannung in V.
  Prozessdaten: PDO Inputs Device
- **Device Current Up:** Laststrom in A.
  Prozessdaten: PDO Inputs Device

Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.5.9 Energy EPP9022



Das *Energy EPP9022*-Modul eignet sich für die Energiebetrachtung von EtherCAT-Boxen des Typs EPP9022. Die Eingangsdaten werden über ein konfigurierbares Intervall gemittelt und anschließend verrechnet. Damit die benötigten Eingangsdaten zur Verfügung stehen, müssen diese im Prozessabbild hinzugefügt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.
- **Interval (optional):** Standardmäßig werden die Eingangsdaten über ein Intervall von einer Sekunde gemittelt.

Eingangswerte

- **Device Voltage Us:** Steuerspannung in V.
  Prozessdaten: PDO Inputs Device
- **Device Current Is:** Steuerstrom in A.
  Prozessdaten: PDO Inputs Device

- **Device Voltage Up:** Lastspannung in V.
  Prozessdaten: PDO Inputs Device
- **Device Current Ip:** Laststrom in A.
  Prozessdaten: PDO Inputs Device

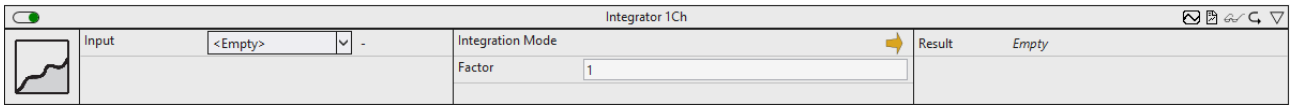
Ausgangswerte

- **Electrical Power [W]:** Elektrische Leistung in W.
- **Electrical Energy [Wh]:** Elektrische Energie in Wh.

6.4.6 Analytics - Math

Die Algorithmen der Kategorie *Analytics-Math* bieten Funktionen für mathematische Operationen, wie Grundrechenarten, Integration oder Steigungsanalyse.

6.4.6.1 Integrator 1Ch



Der *Integrator 1Ch* integriert den Eingangswert über die Zeit mit einer Basiseinheit von einer Sekunde und liefert das Ergebnis dieser Integrationsoperation. Für die Näherung dieses Integrals wird die Trapezregel angewendet. Das Trapez $T(t_n, t_{n+1})$ zwischen zwei aufeinanderfolgenden Zeitstempeln t_n und t_{n+1} mit den Werten y_n und y_{n+1} wird berechnet als

$$T(t_n, t_{n+1}) = (t_{n+1}[s] - t_n[s]) \cdot \frac{y_n + y_{n+1}}{2}.$$

Wenn der Integrationsmodus „absolut“ („|x|“) in der Konfiguration gewählt wird, werden y_n und y_{n+1} in der obigen Gleichung durch ihre absoluten Werte ersetzt.

In jedem Zyklus wird das Trapez zwischen dem aktuellen und dem letzten Zeitstempel berechnet und zu der Summe der Trapeze ab dem Beginn der Analyse addiert. Zusätzlich kann diese Summe um einen Faktor skaliert werden, der einzeln konfiguriert werden kann.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Integration Mode:** Sie können einen Integrationsmodus auswählen. „→“: der Eingangswert wird direkt integriert. „|x|“: Die absoluten Werte des Eingangssignals werden integriert.
- **Factor:** Mit diesem Faktor wird das Integral multipliziert.

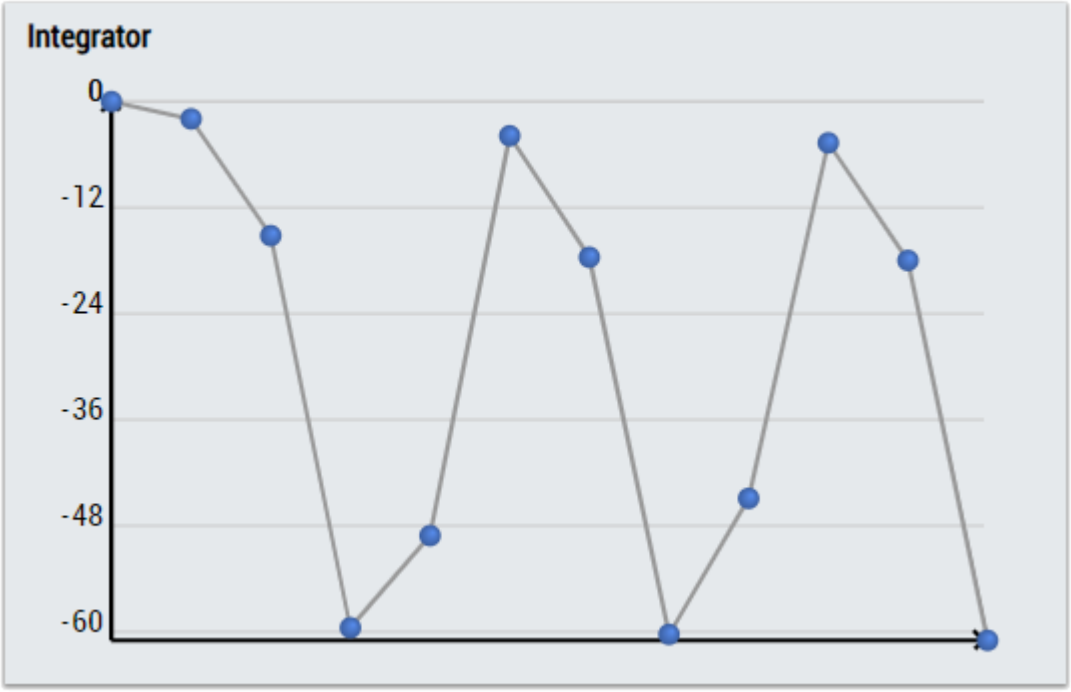
Ausgangswerte

- **Result:** Zeigt das Ergebnis des Integrationsvorgangs an.

Standard HMI Controls

Für den Algorithmus Integrator 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Integrator Control visualisiert die letzten x Ausgangswerte Result.



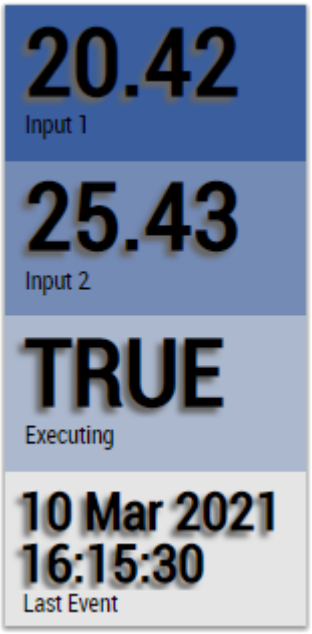
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Result.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

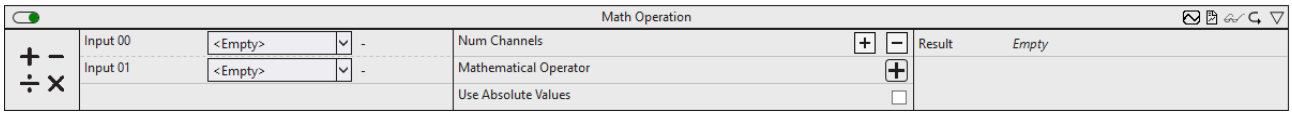
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Integrator 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.6.2 Math Operation



Der *Math Operation* führt eine mathematische Operation an zwei oder mehr verschiedenen Eingangskanälen aus und liefert das Ergebnis der mathematischen Operation. Der Operator ist für alle Operanden gleich.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Operanden, die die Eingänge für die Funktion sind.
- **Mathematical Operator:** Mathematischer Operator der Operation („+“, „-“, „x“, „/“, „x^y“, „%“).
- **Use Absolute Values:** Wenn die Checkbox aktiviert ist, werden die absoluten Werte des Eingangssignals verwendet.

Ausgangswerte

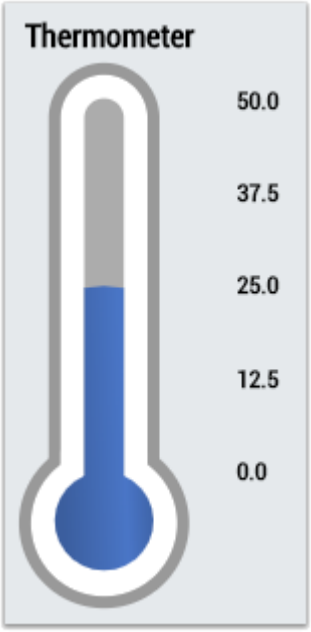
- **Result:** Ergebnis der mathematischen Operation.

Standard HMI Controls

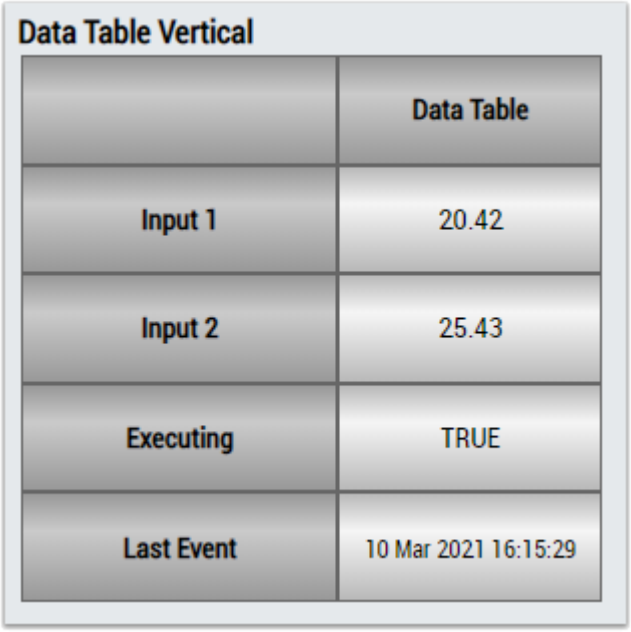
Für den Algorithmus Math Operation sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

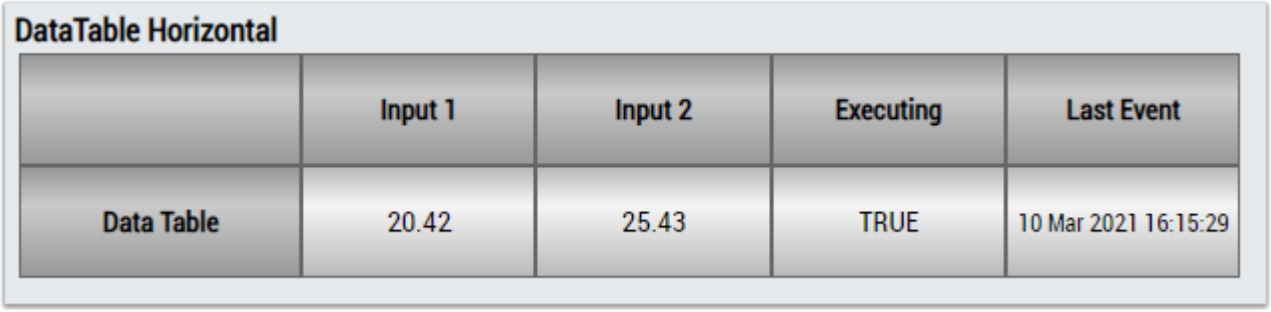
1. Das SingleValue Control visualisiert den Ausgangswert Result.

2. Das Thermometer Control visualisiert den Ausgangswert Result und kann für Temperaturanzeigen verwendet werden.



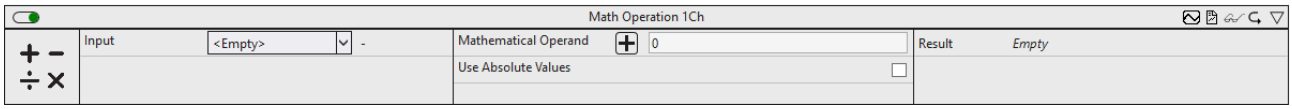
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Result.





Alternativ können im Algorithmus Math Operation mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.6.3 Math Operation 1Ch



Der Math Operation 1Ch führt eine mathematische Operation an dem Signal des Eingangskanals und einem Referenzwert aus. Der Algorithmus liefert das Ergebnis der mathematischen Operation und der Operator kann einzeln konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Mathematical Operator:** Mathematischer Operator der Operation („+“, „-“, „x“, „/“, „x^y“, „%“).
- **Mathematical Operand:** Mathematischer Operand für die Operation.
- **Use Absolute Values:** Wenn die Checkbox aktiviert ist, werden die absoluten Wert der beiden Operanden verwendet.

Ausgangswerte

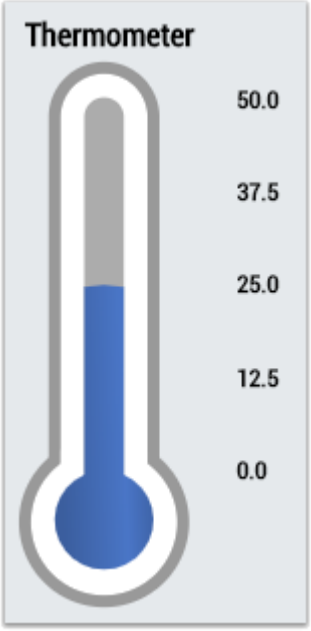
- **Result:** Ergebnis der mathematischen Operation.

Standard HMI Controls

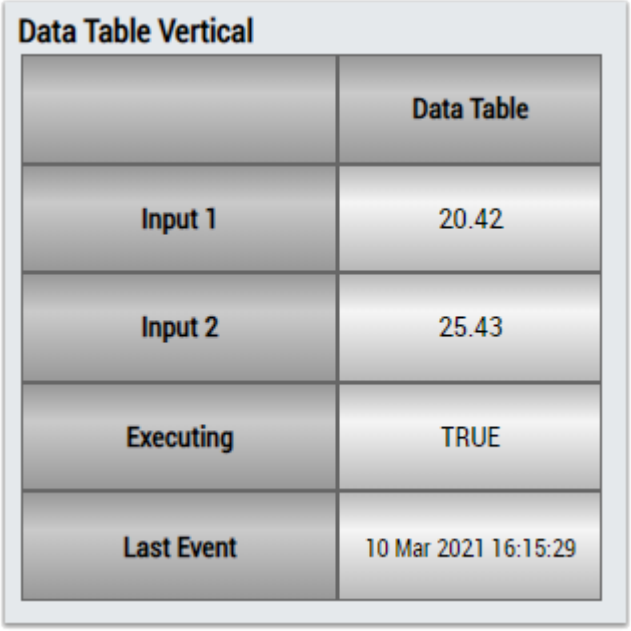
Für den Algorithmus Math Operation sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SingleValue Control visualisiert den Ausgangswert Result.

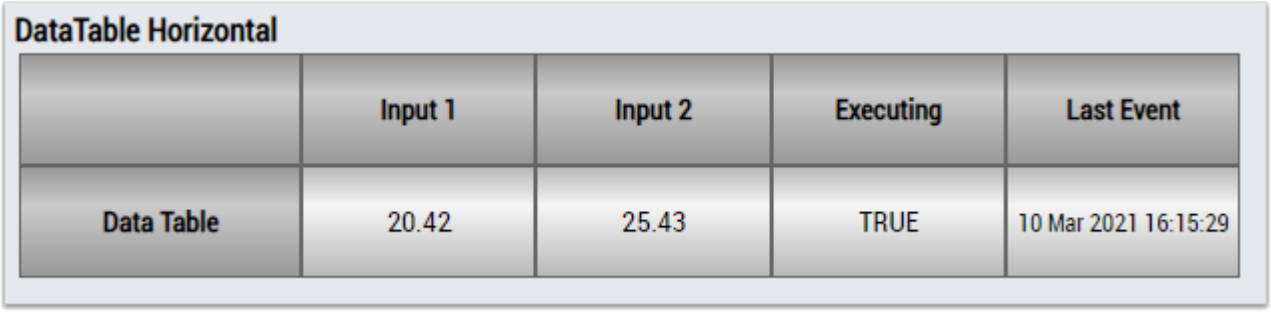
2. Das Thermometer Control visualisiert den Ausgangswert Result und kann für Temperaturanzeigen verwendet werden.



3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Result.

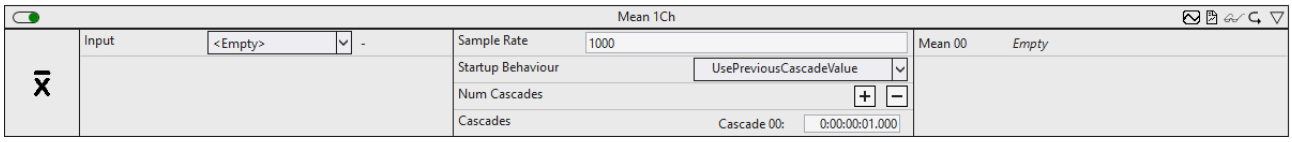


	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Math Operation mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.6.4 Mean 1Ch



Der *Mean 1Ch* berechnet den Mittelwert über die Eingangswerte nach der Formel

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x$$

Die Anzahl an Samples N, die mit in die Berechnung einfließen, kann über die Angabe eines Zeitintervalls konfiguriert werden. Ein kaskadierter Ausgang kann konfiguriert werden, um einen Langzeit-Mittelwert ressourcenschonend zu realisieren und Zwischenergebnisse abzugreifen. Das Zeitintervall der konfigurierten Kaskade muss einem ganzzahligen Vielfachen des Zeitintervalls der vorherigen Kaskade entsprechen.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Cascades:** Anzahl der Ausgangskaskaden
- **Cascades:** Konfiguration der Ausgangskaskaden. Das Zeitintervall der konfigurierten Kaskade muss einem ganzzahligen Vielfachen des Zeitintervalls der vorherigen Kaskade entsprechen.
- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz.

- **Startup Behaviour:**
  WaitUntilFilled wartet ab, bis die konfigurierte Zeitspanne der Kaskade abgelaufen ist. Erst nach Ablauf der Zeitspanne wird das Ergebnis als auch das boolesche Flag „NewResult“ erstmalig gesetzt.
  UsePreviousCascadeValue: Die Kaskaden, deren konfigurierte Zeitspanne noch nicht abgelaufen ist, verwenden das nächstkleinere bereits gesetzte Ergebnis.

Ausgangswerte

- **Mean:** Ausgangsarray in dem die Ergebnisse der Mittelwert-Berechnungen gespeichert werden. Die Dimension entspricht der Anzahl eingestellter Kaskaden. Das Verhalten beim Aufstarten kann über den Parameter *Startup Behaviour* eingestellt werden.

Standard HMI Controls

Für den Algorithmus Mean sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Mean Results.

Data Table Vertical

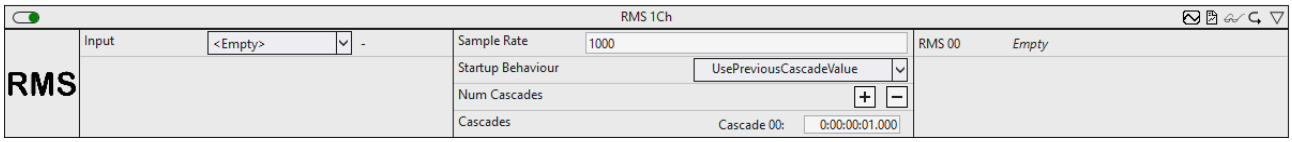
	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

Alternativ können im Algorithmus Mean mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.6.5 RMS 1Ch



Der *RMS 1Ch* berechnet das quadratische Mittel (root mean square) über die Eingangswerte nach der Formel

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2}$$

Die Anzahl an Samples N, die mit in die Berechnung einfließen, kann über die Angabe eines Zeitintervalls konfiguriert werden. Ein kaskadierter Ausgang kann konfiguriert werden, um einen Langzeit-RMS ressourcenschonend zu realisieren und Zwischenergebnisse abzugreifen. Das Zeitintervall der konfigurierten Kaskade muss einem ganzzahligen Vielfachen des Zeitintervalls der vorherigen Kaskade entsprechen.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Cascades:** Anzahl der Ausgangskaskaden
- **Cascades:** Konfiguration der Ausgangskaskaden. Das Zeitintervall der konfigurierten Kaskade muss einem ganzzahligen Vielfachen des Zeitintervalls der vorherigen Kaskade entsprechen.
- **Sample Rate:** Samplerate des zu analysierenden Systems in Hz
- **Startup Behaviour:**
  WaitUntilFilled wartet ab, bis die konfigurierte Zeitspanne der Kaskade abgelaufen ist. Erst nach Ablauf der Zeitspanne wird das RMS-Ergebnis als auch das boolesche Flag „NewResult“ erstmalig gesetzt.
  UsePreviousCascadeValue Die RMS-Kaskaden, deren konfigurierte Zeitspanne noch nicht abgelaufen ist, verwenden das nächstkleinere bereits gesetzte RMS-Ergebnis.

Ausgangswerte

- **RMS:** Ausgangsarray in dem die Ergebnisse der RMS-Berechnungen gespeichert werden. Die Dimension entspricht der Anzahl eingestellter Kaskaden. Das Verhalten beim Aufstarten kann über den Parameter *Startup Behaviour* eingestellt werden.

Standard HMI Controls

Für den Algorithmus RMS sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

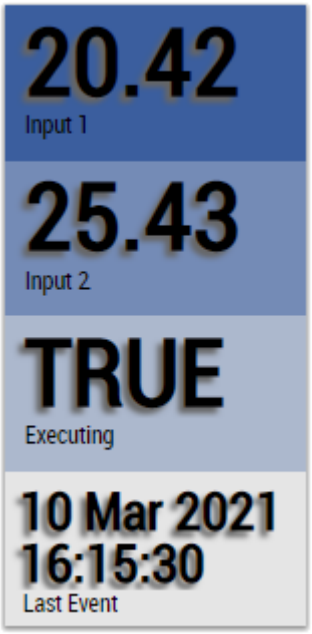
1. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: RMS Results.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

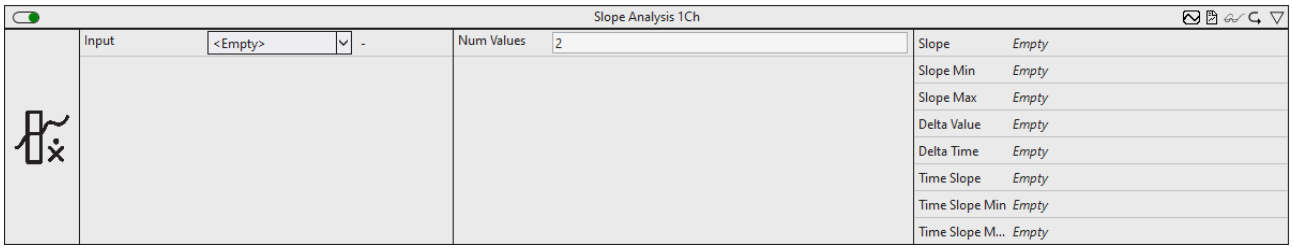
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus RMS mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.6.6 Slope Analysis 1Ch



Der *Slope Analysis 1Ch* berechnet die Steigung zwischen zwei Werten des Eingangsstroms. Einer dieser beiden Werte ist der aktuelle Eingangswert und der zweite Wert ist der Eingangswert, der eine festgelegte Anzahl (konfiguriert durch den Parameter *Num Values*) von Zyklen zuvor im Eingangsstrom aufgetreten ist. Die Differenz zwischen diesen beiden Werten wird als *Delta Value* zurückgegeben.

Der entsprechende Abstand auf der Zeitkoordinate wird als Differenz der Zeitstempel dieser beiden Werte berechnet und als Ausgangswert *Delta Time* bereitgestellt. Zu beachten ist, dass der Wert *Delta Time* in Nanosekunden angezeigt wird, für die Berechnung der Steigung jedoch auf eine Sekunde als Basiseinheit skaliert wird.

Der *Slope* wird dann als Bruch von *Delta Value* und *Delta Time* (skaliert auf Sekunden) berechnet und der Gradient für den Zeitstempel in der Mitte der beiden in der Berechnung von *Delta Time* verwendeten Zeitstempel geschätzt. Dies ist der Wert, der als *Time Slope* zurückgegeben wird, wenn er einem Zeitstempel des Eingangsstroms entspricht. Bei Konfigurationen, bei denen *Num Values* eine ungerade Zahl ist, gibt es keinen Eingangswert, der exakt mit dem Zeitstempel in der Mitte übereinstimmt. In diesem Fall wird der Zeitstempel des Werts, der direkt auf den berechneten Zeitstempel in der Mitte gefolgt ist, als *Time Slope* zurückgegeben.

Des Weiteren liefert der Algorithmus die minimale Steigung, die maximale Steigung und die Zeitwerte von Minimum und Maximum.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

Num Values: Anzahl der Zyklen, die zwischen den für die Berechnung der Steigung verwendeten Werten liegen.

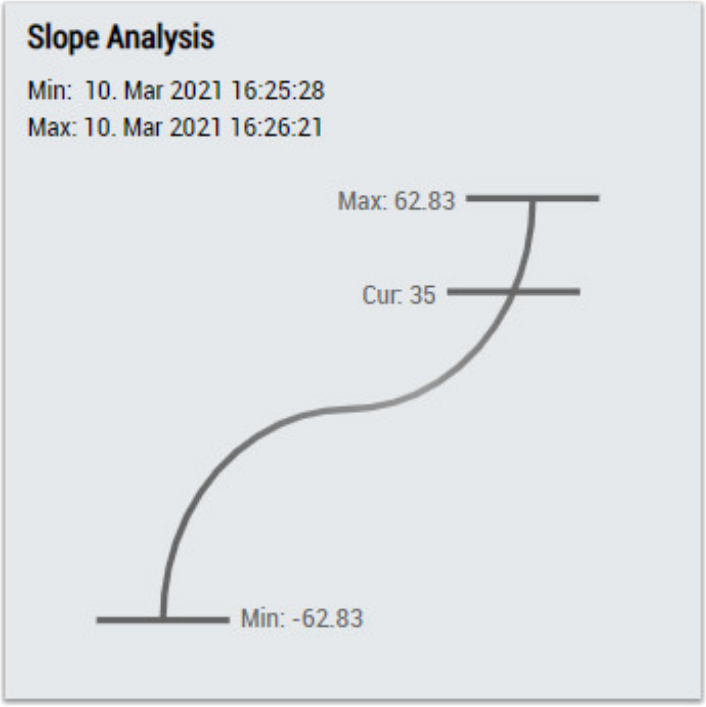
Ausgangswerte

- **Slope:** Zeigt den aktuellen Steigungswert an.
- **Slope Min:** Zeigt das Minimum der Steigungswerte an.
- **Slope Max:** Zeigt das Maximum der Steigungswerte an.
- **Delta Value:** Zeigt die Differenz zwischen den beiden Werten an, die für die Berechnung der neuesten Steigung verwendet werden.
- **Delta Time:** Zeigt den Zeitraum an, der für die Berechnung der neuesten Steigung verwendet wurde.
- **Time Slope:** Zeigt den Zeitwert des neuesten Steigungswerts an.
- **Time Slope Min:** Zeigt den Zeitwert des Minimums der Steigung an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **Time Slope Max:** Zeigt den Zeitwert des Maximums der Steigung an → dieses Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

Standard HMI Controls

Für den Algorithmus *Slope Analysis 1Ch* sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SlopeAnalysis Control visualisiert die Ausgangswerte Slope, Slope Min, Slope Max, Time Slope Min und Time Slope Max.



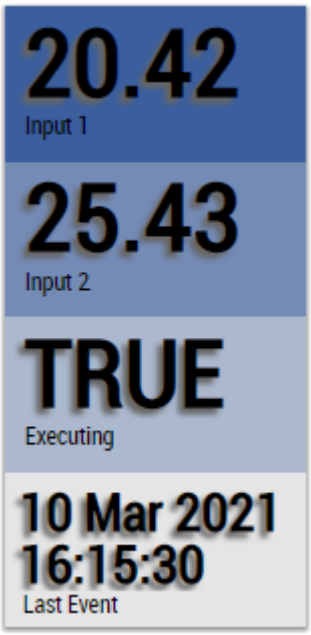
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Slope, Slope Min, Slope Max, Delta Value, Delta Time, Time Slope, Time Slope Min, Time Slope Max.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

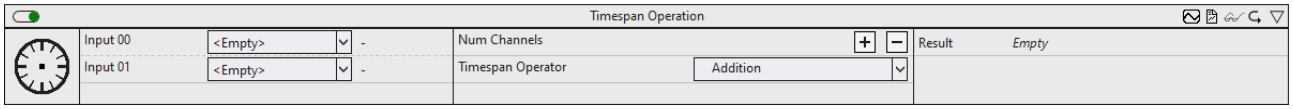
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.6.7 Timespan Operation



Die *Timespan Operation* führt eine mathematische Operation an zwei oder mehr verschiedenen Eingangskanälen aus und liefert das Ergebnis der mathematischen Operation. Als Eingänge werden Zeitspannen mit einer Auflösung von 1 ns unterstützt.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Operanden, die die Eingänge für die Funktion sind.
- **Timespan Operator:** Mathematischer Operator der Operation (*Addition* / *Subtraktion*).

Ausgangswerte

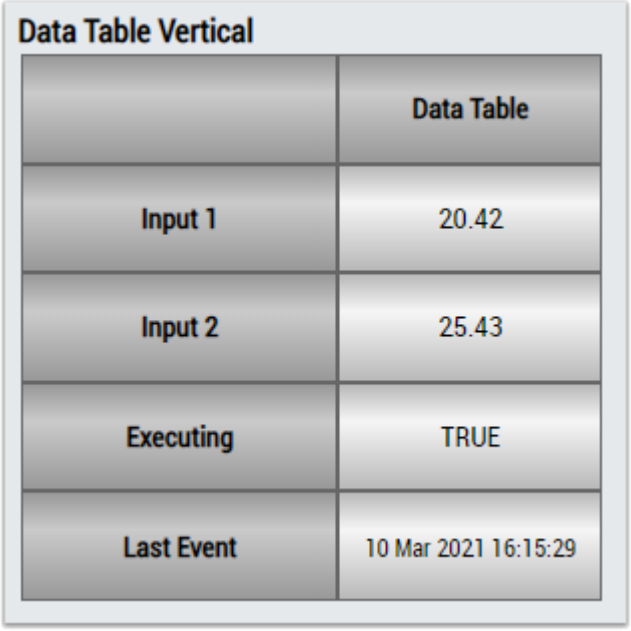
- **Result:** Ergebnis der mathematischen Operation als Zeitspanne.

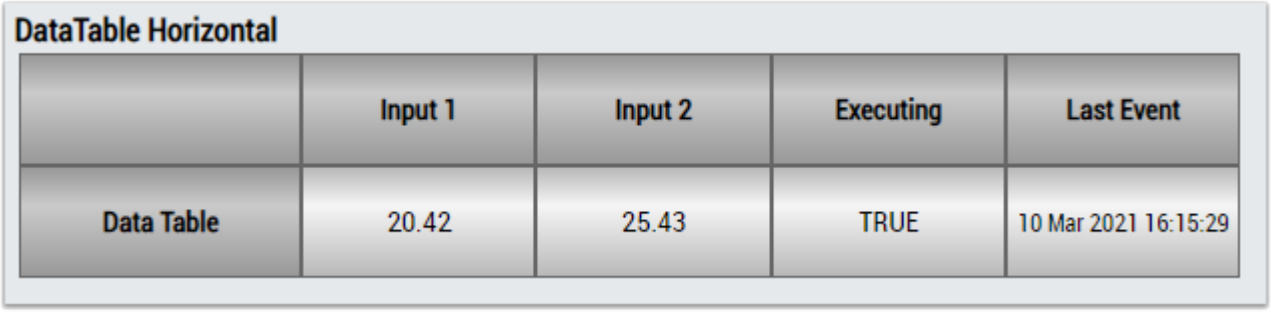
Standard HMI Controls

Für den Algorithmus Timespan Operation sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SingleValue Control visualisiert den Ausgangswert Result.

2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Result.

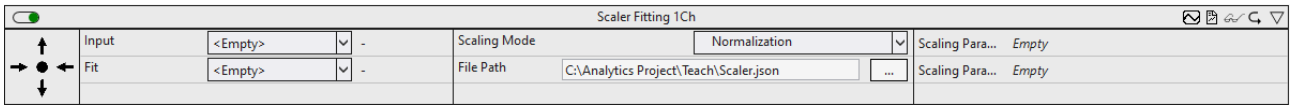




Alternativ können im Algorithmus Timespan Operation mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.7 Analytics - Preprocessing

6.4.7.1 Scaler Fitting



Das *Scaler Fitting*-Modul kann verwendet werden, um die Parameter für ein Feature-Scaling zu ermitteln und in einer Datei abzuspeichern. Anschließend können diese Parameter im *Scaler*-Modul verwendet werden.

Abhängig vom Betriebsmodus werden die für die Skalierung erforderlichen Parameter ermittelt.

Normalisierung:

$$x' = \frac{x - low(x)}{high(x) - low(x)}$$

Standardisierung:

$$x' = \frac{x - \bar{x}}{\sigma}$$

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

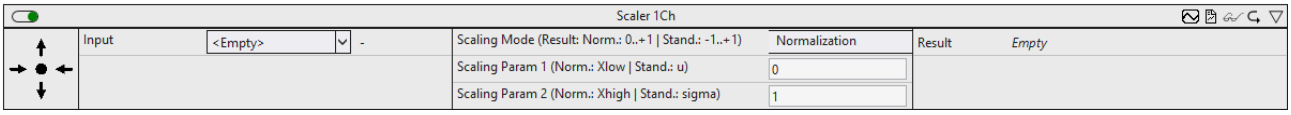
Konfigurationsoptionen

- **Scaling Mode:** Modus für die Skalierung.
  Normalization
  Standardization
- **File Path:** Gibt den Dateipfad für die Datei an, in der die Skalierungsparameter gespeichert werden. Dateityp: *json*.

Ausgangswerte

- **Scaling Param 1:** Der Wert des ersten Skalierungsparameters ist abhängig vom eingestellten Skalierungsmodus.
  Normalization: Kleinster Wert des Eingangssignals
  Standardization: Mittelwert des Eingangssignals
- **Scaling Param 2:** Der Wert des zweiten Skalierungsparameters ist abhängig vom eingestellten Skalierungsmodus.
  Normalization: Größter Wert des Eingangssignals
  Standardization: Standardabweichung des Eingangssignals

6.4.7.2 Scaler



Das *Scaler*-Modul kann verwendet werden, um ein Feature-Scaling durchzuführen.

Abhängig vom Betriebsmodus werden die Skalierungsparameter auf verschiedene Arten verwendet.

Normalisierung

$$x' = \frac{x - low(x)}{high(x) - low(x)}$$

Standardisierung:

$$x' = \frac{x - \bar{x}}{\sigma}$$

Die Skalierungsparameter können direkt konfiguriert werden. Alternativ können sie aus einer Datei importiert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Scaling Mode:** Modus für die Skalierung.
  Normalization: Ergebnis hat einen Wertebereich von 0..+1
  Standardization: Ergebnis hat einen Wertebereich von -1..+1
- **Scaling Param 1:** Der Wert des ersten Skalierungsparameters ist abhängig vom eingestellten Skalierungsmodus.
  Normalization: Kleinster Wert des Eingangssignals
  Standardization: Mittelwert des Eingangssignals
- **Scaling Param 2:** Der Wert des zweiten Skalierungsparameters ist abhängig vom eingestellten Skalierungsmodus.
  Normalization: Größter Wert des Eingangssignals
  Standardization: Standardabweichung des Eingangssignals
- **File Path (optional):** Dateipfad zu einer Datei mit zuvor ermittelten Parametern. Die Datei wird durch das Modul *Scaler Fitting* erstellt. Dateityp: *json*.

Ausgangswerte

- **Result:** Gibt den berechneten Wert aus. Dieser wird abhängig vom Skalierungsmodus aus den beiden Skalierungsparametern berechnet.
  Normalization: Ergebnis hat einen Wertebereich von 0..+1
  Standardization: Ergebnis hat einen Wertebereich von -1..+1

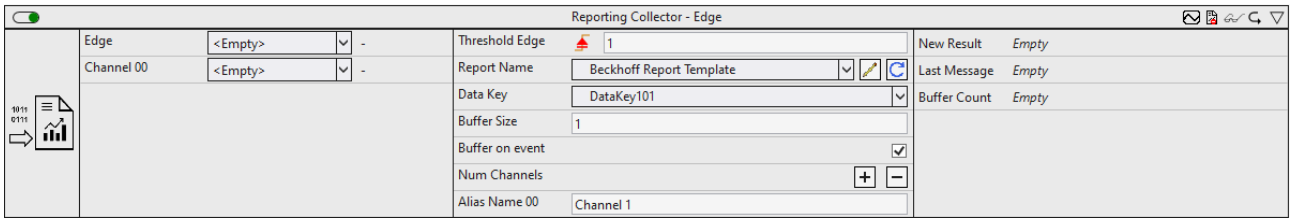
6.4.8 Analytics – Reporting

Das 24/7-Reporting kann in TwinCAT Analytics mit den Algorithmen der Kategorie *Reporting* umgesetzt werden. Die Reporting Collectoren sammeln die Daten und versenden diese an den Reporting-Server. Die Reporting Trigger lösen die Erstellung eines Reports aus.

6.4.8.1 Reporting Collector

Die Reporting Collectoren sammeln Daten und versenden diese nach einem Event in einer Daten-Nachricht an den Reporting Server.

6.4.8.1.1 Reporting Collector Edge



Der Reporting Collector Edge sammelt die Daten der Eingangskanäle und sendet die Daten je nach Konfiguration nach einem Ereignis oder nach dem Befüllen des Buffers an den Reporting Server. Ein Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten Schwelle passiert.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

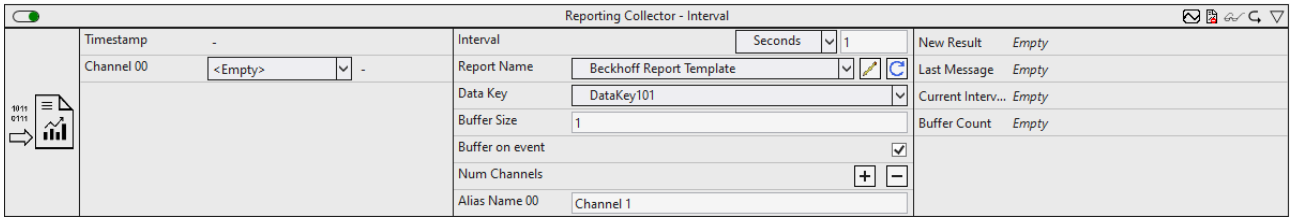
- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Ereignis ausgelöst.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.
- **Data Key:** Der Data Key sollte innerhalb eines Reports eindeutig sein. Über den Data Key lässt sich das Datenobjekt eindeutig in den Report einordnen.
- **Buffer Size:** Gibt die Größe der gepufferten Daten an. Ist die Buffer Size gleich eins, wird eine Key-Value-Struktur aufgebaut. Ist die Buffer Size größer als eins, werden die Daten in einer Tabelle dargestellt.
- **Buffer on event:** Gibt an, wie die Daten gesammelt und gepuffert werden sollen.
  Wenn der Parameter True ist, wird bei jedem Edge-Signal am Eingang die Daten der Inputs im Buffer zwischengespeichert. Sobald der Ausgang Buffer Count den gleichen Wert wie der Parameter Buffer Size hat, werden die Daten an den TcReportingServer versendet.
  Wenn der Parameter False ist, werden bei jedem Zyklus die Daten der Inputs im Buffer zwischengespeichert. Sobald die Buffer Size erreicht ist, ersetzen die neuen Daten die vorherigen Daten. Bei einem Edge-Signal am Eingang werden die Daten an den TcReportingServer versendet.

- **Num Channels:** Die Anzahl der Eingangskanäle, von denen die Daten gesammelt werden. Und die Anzahl der Parameter, um eine Beschreibung der Eingangskanäle zu ermöglichen.
- **Alias Name:** Dient als Beschreibung des Eingangskanals. BufferSize = 1: Der Alias Name dient als Key in der Key-Value-Struktur. BufferSize > 1: Der Alias Name dient als Überschrift der Tabellenspalte.
- **Include Timestamps (optional):** Fügt eine Spalte mit den Zeitstempeln ein.

Ausgangswerte

- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Buffer Count:** Gibt die Anzahl an Elementen im Buffer an.

6.4.8.1.2 Reporting Collector Interval



Der Reporting Collector Interval sammelt die Daten der Eingangskanäle und sendet die Daten je nach Konfiguration nach einem Ereignis oder nach dem Befüllen des Buffers an den Reporting Server. Ein Ereignis wird ausgelöst, wenn die Zeitspanne des konfigurierten Intervalls abgelaufen ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

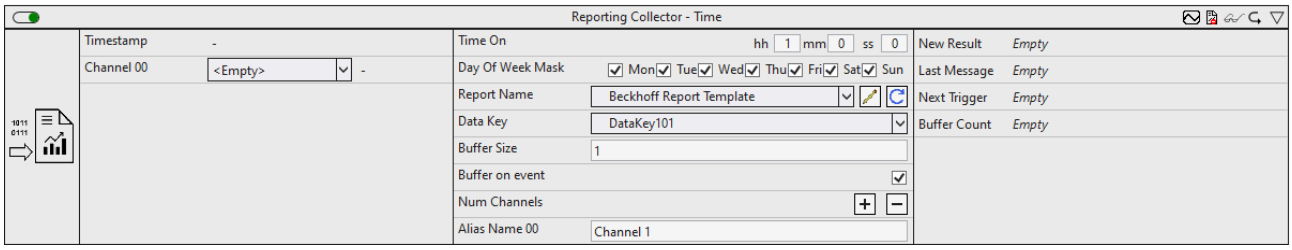
Konfigurationsoptionen

- **Interval:** Zeitintervall, in dem die Daten an den Reporting Server gesendet werden sollen.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.
- **Data Key:** Der Data Key sollte innerhalb eines Reports eindeutig sein. Über den Data Key lässt sich das Datenobjekt eindeutig in den Report einordnen.
- **Buffer Size:** Gibt die Größe der gepufferten Daten an. Ist die Buffer Size gleich eins, wird eine Key-Value-Struktur aufgebaut. Ist die Buffer Size größer als eins, werden die Daten in einer Tabelle dargestellt.
- **Buffer on event:** Gibt an, wie die Daten gesammelt und gepuffert werden sollen.
  Wenn der Parameter True ist, wird bei jedem Edge-Signal am Eingang die Daten der Inputs im Buffer zwischengespeichert. Sobald der Ausgang Buffer Count den gleichen Wert wie der Parameter Buffer Size hat, werden die Daten an den TcReportingServer versendet.
  Wenn der Parameter False ist, werden bei jedem Zyklus die Daten der Inputs im Buffer zwischengespeichert. Sobald die Buffer Size erreicht ist, ersetzen die neuen Daten die vorherigen Daten. Bei einem Edge-Signal am Eingang werden die Daten an den TcReportingServer versendet.
- **Num Channels:** Die Anzahl der Eingangskanäle, von denen die Daten gesammelt werden. Und die Anzahl der Parameter, um eine Beschreibung der Eingangskanäle zu ermöglichen.
- **Alias Name:** Dient als Beschreibung des Eingangskanals. BufferSize = 1: Der Alias Name dient als Key in der Key-Value-Struktur. BufferSize > 1: Der Alias Name dient als Überschrift der Tabellenspalte.
- **Include Timestamps (optional):** Fügt eine Spalte mit den Zeitstempeln ein.

Ausgangswerte

- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Current Interval Time:** Zeigt die Zeitspanne an, die vom aktuellen Intervall bereits verstrichen ist.
- **Buffer Count:** Gibt die Anzahl an Elementen im Buffer an.

6.4.8.1.3 Reporting Collector Time



Der Reporting Collector Time sammelt die Daten der Eingangskanäle und sendet die Daten je nach Konfiguration nach einem Ereignis oder nach dem Befüllen des Buffers an den Reporting Server. Ein Ereignis wird ausgelöst, wenn die konfigurierte Einschaltzeit erreicht wird. Die Einschaltzeit und die Tage der Wochen lassen sich konfigurieren.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Time On:** Einschaltzeit.
- **Day of Week Mask:** Wochentage, an denen die Zeitschaltung aktiv sein soll.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.
- **Data Key:** Der Data Key sollte innerhalb eines Reports eindeutig sein. Über den Data Key lässt sich das Datenobjekt eindeutig in den Report einordnen.
- **Buffer Size:** Gibt die Größe der gepufferten Daten an. Ist die Buffer Size gleich eins, wird eine Key-Value-Struktur aufgebaut. Ist die Buffer Size größer als eins, werden die Daten in einer Tabelle dargestellt.
- **Buffer on event:** Gibt an, wie die Daten gesammelt und gepuffert werden sollen.
  Wenn der Parameter True ist, wird bei jedem Edge-Signal am Eingang die Daten der Inputs im Buffer zwischengespeichert. Sobald der Ausgang Buffer Count den gleichen Wert wie der Parameter Buffer Size hat, werden die Daten an den TcReportingServer versendet.
  Wenn der Parameter False ist, werden bei jedem Zyklus die Daten der Inputs im Buffer zwischengespeichert. Sobald die Buffer Size erreicht ist, ersetzen die neuen Daten die vorherigen Daten. Bei einem Edge-Signal am Eingang werden die Daten an den TcReportingServer versendet.
- **Num Channels:** Die Anzahl der Eingangskanäle, von denen die Daten gesammelt werden. Und die Anzahl der Parameter, um eine Beschreibung der Eingangskanäle zu ermöglichen.
- **Alias Name:** Dient als Beschreibung des Eingangskanals. BufferSize = 1: Der Alias Name dient als Key in der Key-Value-Struktur. BufferSize > 1: Der Alias Name dient als Überschrift der Tabellenspalte.
- **Include Timestamps (optional):** Fügt eine Spalte mit den Zeitstempeln ein.

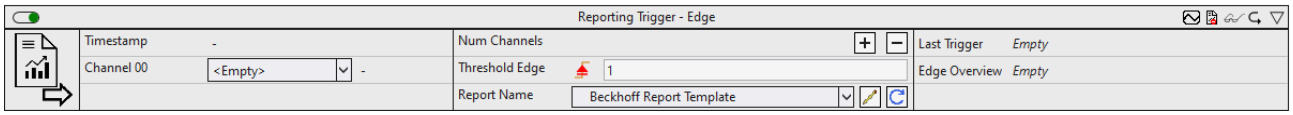
Ausgangswerte

- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Next Message:** Zeigt die verbleibende Zeit bis zur nächsten Nachricht an.
- **Buffer Count:** Gibt die Anzahl an Elementen im Buffer an.

6.4.8.2 Reporting Trigger

Die Reporting Trigger versenden nach einem Event eine Trigger-Nachricht an den Reporting Server und lösen damit die Erstellung eines Reports aus.

6.4.8.2.1 Reporting Trigger Edge



Der Reporting Trigger Edge löst die Erstellung eines Reports aus, nachdem ein Ereignis ausgelöst wurde. Ein Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einem bestimmten Schwellenwert überschreitet. Intern bleiben die Eingänge, die einmal True waren auf True. Die Eingänge werden lediglich auf False zurückgesetzt, sobald alle Eingänge mindestens einmal True waren. Dadurch kann der Ausgang bNewResult von mehreren Reporting Collectoren als ein Eingang genutzt werden und sobald alle Reporting Collectoren eine Daten-Nachricht gesendet haben, wird eine Trigger-Nachricht versendet.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

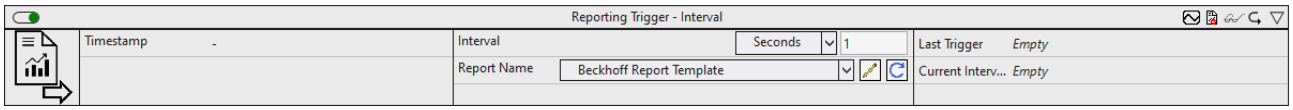
Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Ereignis ausgelöst.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.

Ausgangswerte

- **Last Trigger:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Edge Overview:** Zeigt an, welche Eingangskanäle mindestens einmal True waren.

6.4.8.2.2 Reporting Trigger Interval



Der Reporting Trigger Interval löst die Erstellung eines Reports aus, nachdem ein Ereignis ausgelöst wurde. Ein Ereignis wird ausgelöst, wenn die Zeitspanne des konfigurierten Intervalls abgelaufen ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

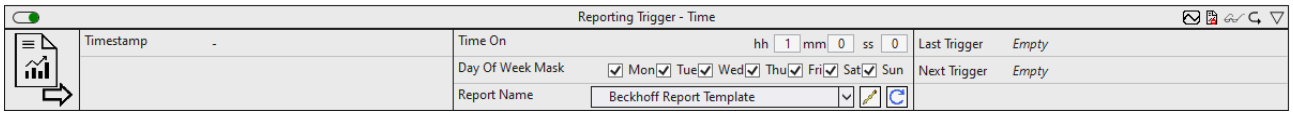
Konfigurationsoptionen

- **Interval:** Zeitintervall, in dem die Daten an den Reporting Server gesendet werden sollen.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.

Ausgangswerte

- **Last Trigger:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Current Interval Time:** Zeigt die Zeitspanne an, die vom aktuellen Intervall bereits verstrichen ist.

6.4.8.2.3 Reporting Trigger Time



Der Reporting Trigger Time löst die Erstellung eines Reports aus, nachdem ein Ereignis ausgelöst wurde. Ein Ereignis wird ausgelöst, wenn die konfigurierte Einschaltzeit erreicht wird. Die Einschaltzeit und die Tage der Wochen lassen sich konfigurieren.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Time On:** Einschaltzeit.
- **Day of Week Mask:** Wochentage, an denen die Zeitschaltung aktiv sein soll.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.

Ausgangswerte

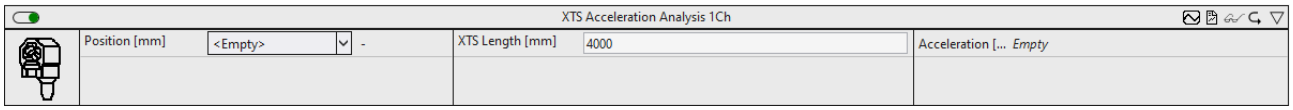
- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Next Trigger:** Zeigt die verbleibende Zeit bis zur nächsten Nachricht an.

6.4.9 Analytics - Specific

6.4.9.1 XTS

Die Algorithmen der Kategorie *Analytics-XTS* bieten spezielle Funktionen für das Beckhoff XTS-System, zum Beispiel die Analyse von Entfernung, Geschwindigkeit und Beschleunigung.

6.4.9.1.1 XTS Acceleration Analysis 1Ch



Der *XTS Acceleration Analysis 1Ch* berechnet die aktuelle Beschleunigung eines XTS-Movers. Zu diesem Zweck muss die Länge des XTS in Millimetern angegeben werden, und als Eingangssignal wird die Position des Movers benötigt.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **XTS Length [mm]:** Länge des gegebenen XTS-Systems in Millimetern.

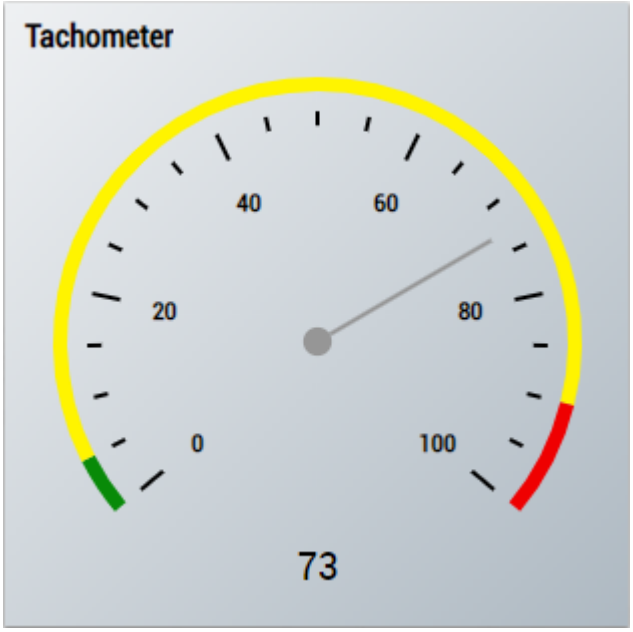
Ausgangswerte

- **Acceleration:** Aktuelle Beschleunigung des XTS-Movers.

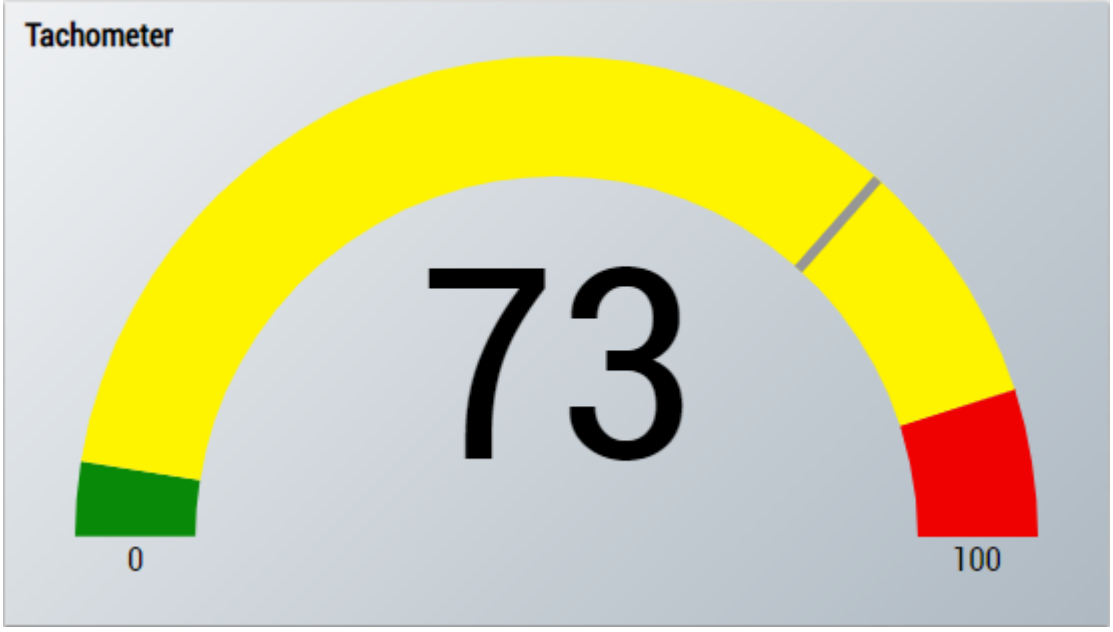
Standard HMI Controls

Für den Algorithmus XTS Acceleration Analysis 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Tachometer Control visualisiert den Ausgangswert Acceleration.



2. Das Radial Gauge Control visualisiert den Ausgangswert Acceleration.



3. Das SingleValue Control visualisiert den Ausgangswert Acceleration.

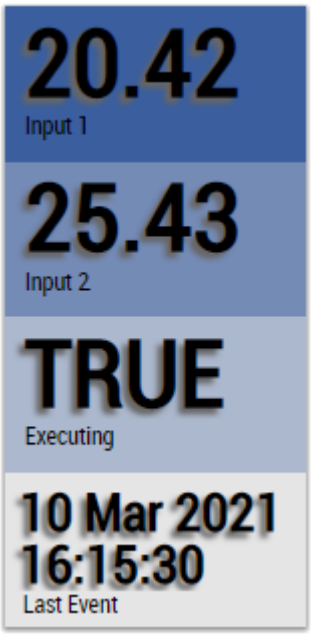
4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Acceleration.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

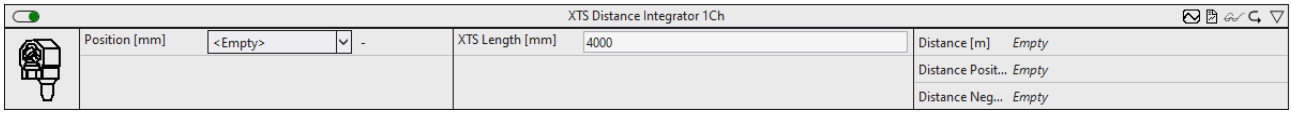
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus XTS Acceleration Analysis 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.9.1.2 XTS Distance Integrator 1Ch



Der *XTS Distance Integrator 1Ch* berechnet die zurückgelegte Entfernung eines XTS-Movers. Der Algorithmus liefert die Gesamtentfernung, die positive Entfernung und die negative Entfernung. Zu diesem Zweck muss die Länge des XTS in Millimetern angegeben werden, und als Eingangssignal wird die Position des Movers benötigt.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **XTS Length [mm]:** Länge des gegebenen XTS-Systems in Millimetern.

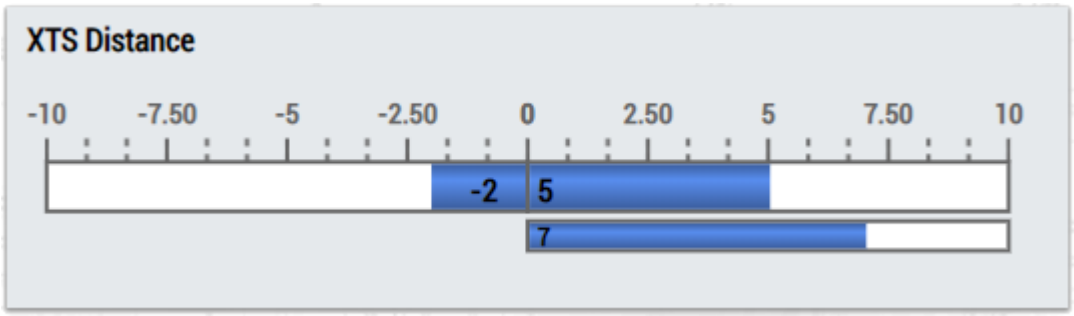
Ausgangswerte

- **Distance:** Gesamtentfernung, die der XTS-Mover zurückgelegt hat.
- **Distance Positive:** Positive Entfernung, die der XTS-Mover zurückgelegt hat (Richtung: vorwärts).
- **Distance Negative:** Negative Entfernung, die der XTS-Mover zurückgelegt hat (Richtung: rückwärts).

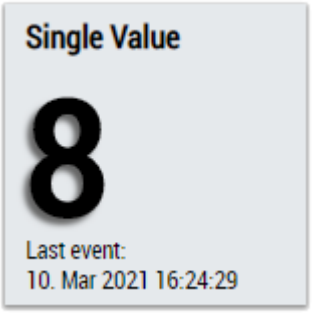
Standard HMI Controls

Für den Algorithmus XTS Distance Integrator 1Ch stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das XTSDistance Control visualisiert die Ausgangswerte Distance, Distance Positive und Distance Negative.



2. Das SingleValue Control visualisiert den Ausgangswert Distance.



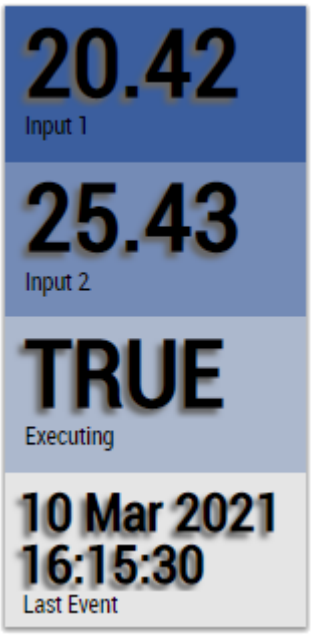
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Ausgangswerte Distance, Distance Positive, Distance Negative.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

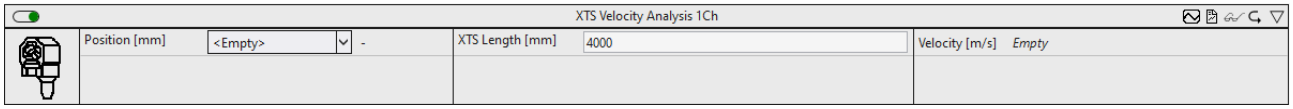
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus XTS Distance Integrator 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.9.1.3 XTS Velocity Analysis 1Ch



XTS Velocity Analysis 1Ch berechnet die aktuelle Geschwindigkeit eines XTS-Movers. Zu diesem Zweck muss die Länge des XTS in Millimetern angegeben werden, und als Eingangssignal wird die Position des Movers benötigt.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **XTS Length [mm]:** Länge des gegebenen XTS-Systems in Millimetern.

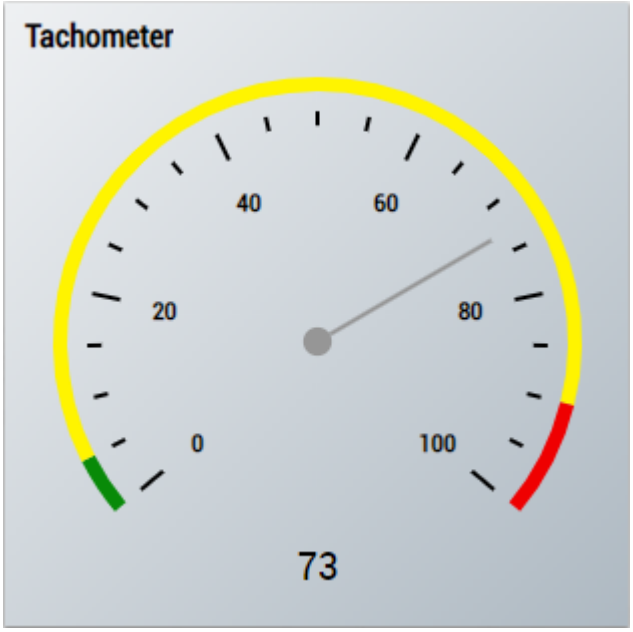
Ausgangswerte

- **Velocity:** Aktuelle Geschwindigkeit des XTS-Movers.

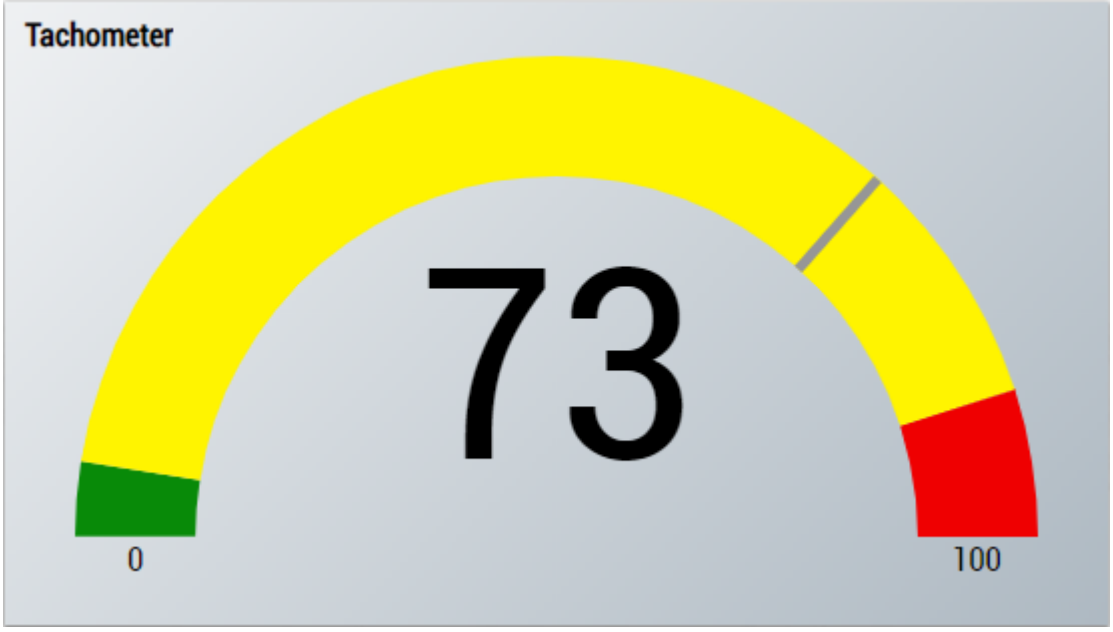
Standard HMI Controls

Für den Algorithmus XTS Velocity Analysis 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Tachometer Control visualisiert den Ausgangswert Velocity.

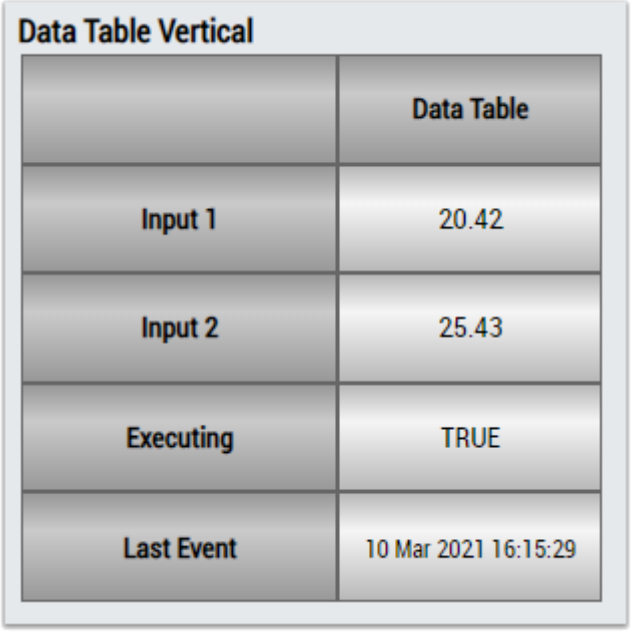


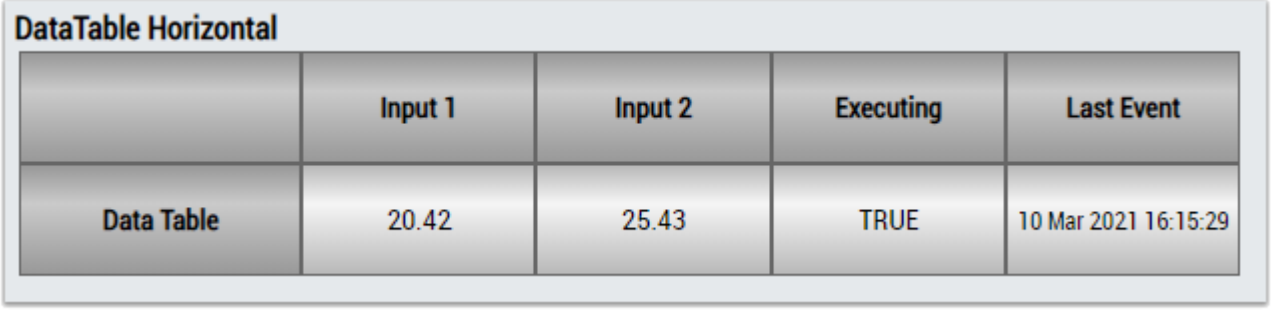
2. Das Radial Gauge Control visualisiert den Ausgangswert Velocity.

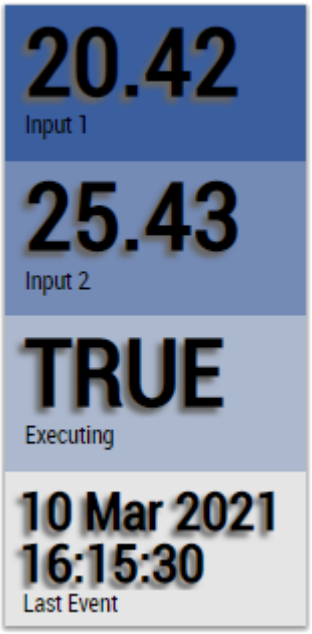


3. Das SingleValue Control visualisiert den Ausgangswert Velocity.

4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Velocity.





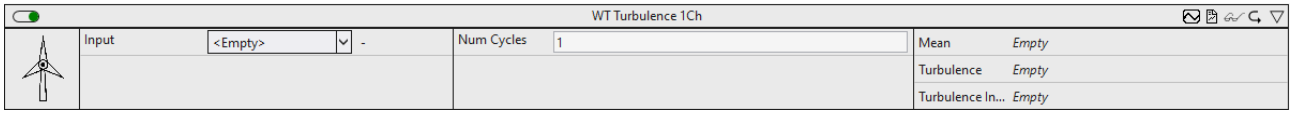


Alternativ können im Algorithmus XTS Velocity Analysis 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.9.2 Wind Turbine

Die Algorithmen der Kategorie *Analytics-WT* bieten spezielle Funktionen für die Windbranche, zum Beispiel die Analyse von mittlerer Windgeschwindigkeit, Turbulenz und Turbulenzintensität.

6.4.9.2.1 WT Turbulence 1Ch



WT Turbulence 1Ch berechnet den Mittelwert der Windgeschwindigkeit, die Turbulenz und die Turbulenzintensität nach der Norm *EN 61400-1*. Als Eingangssignal wird die Windgeschwindigkeit benötigt. Die Ausgangswerte werden in einem Zyklus von 10 Minuten aktualisiert.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Cycles 10Min:** Gibt die Anzahl der Zyklen an, die in das Zeitintervall für die Berechnung passen, nach EN 61400-1 ist dies ein Intervall von zehn Minuten.

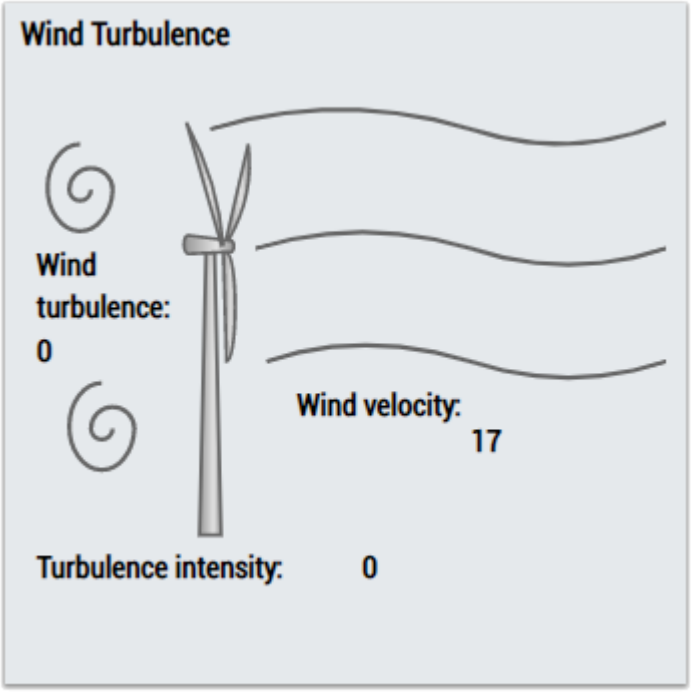
Ausgangswerte

- **Mean:** Mittelwert der Windgeschwindigkeit.
- **Turbulence:** Turbulenz des Windes. Nach der EN-Norm ist dies die Standardabweichung der Windgeschwindigkeit während eines Zeitintervalls von 10 Minuten.
- **Turbulence Intensity:** Intensität der Windturbulenz.

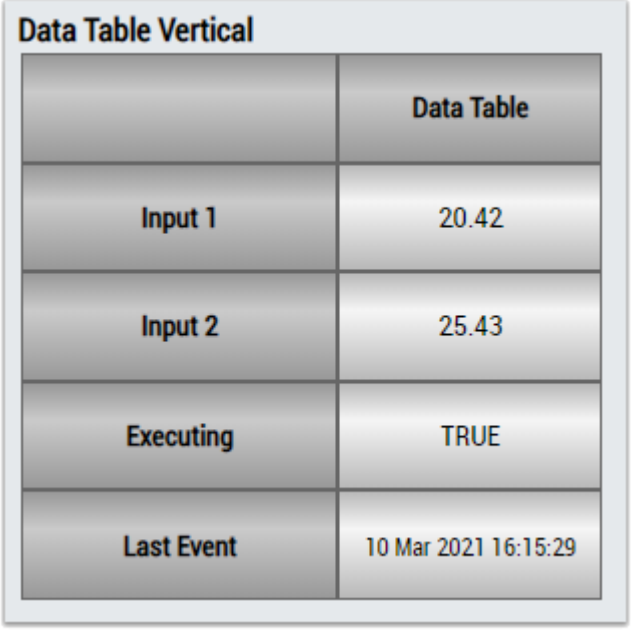
Standard HMI Controls

Für den Algorithmus WT Turbulence 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

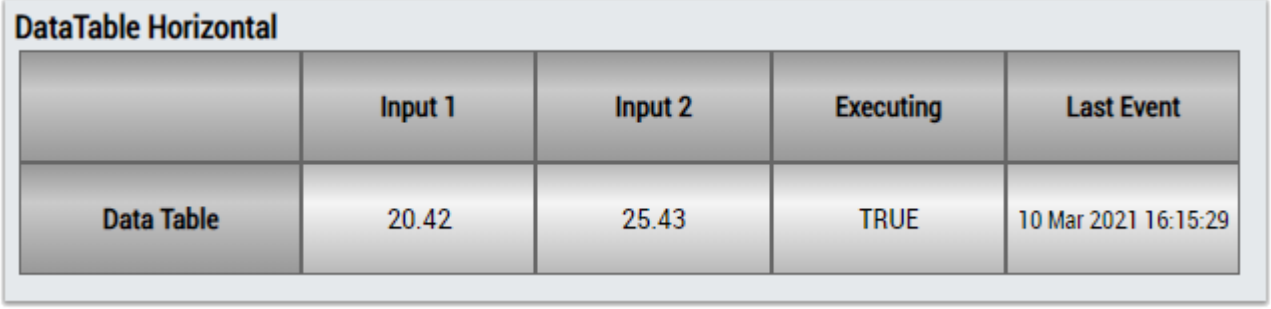
1. Das WindTurbulence Control visualisiert die Ausgangswerte Mean, Turbulence und Turbulence Intensity.



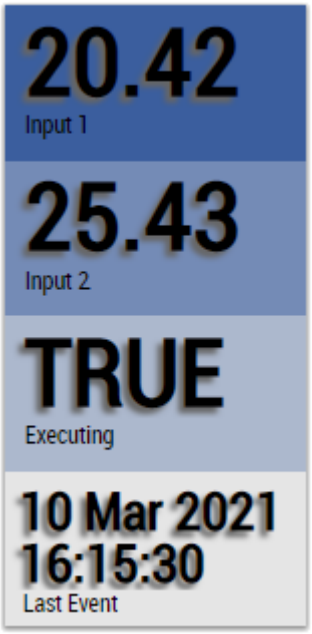
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Mean, Turbulence, Turbulence Intensity.



	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29



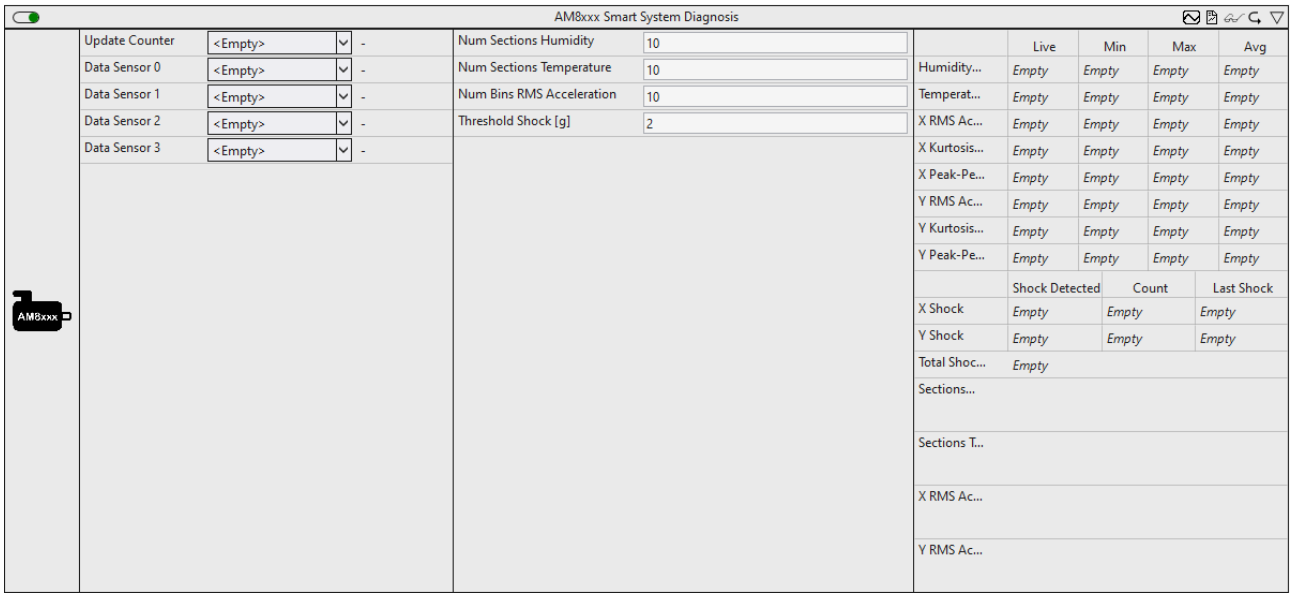
	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus WT Turbulence 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.9.3 AM

6.4.9.3.1 AM8xxx Smart System Diagnosis



Der *AM8xxxSSD* ermöglicht die Analyse eines Motors, der mit der Smart System Diagnosis-Funktionalität ausgestattet ist. Der verbaute Encoder nimmt Messwerte verschiedener, intern verbauter Sensoren auf. Dazu zählen die Temperatur und Luftfeuchtigkeit im Encoder, als auch die Beschleunigung in X- und Y-Richtung. Auf Basis dieser Messwerte werden verschiedene statistische Kenngrößen errechnet und eine Schwellwertüberwachung zur Detektion von Schocks durchgeführt.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Sections Humidity:** Konfigurieren der Anzahl der Sektionen für die Analyse der Luftfeuchte im Encoder.

- **Num Sections Temperature:** Konfigurieren der Anzahl der Sektionen für die Analyse der Temperatur im Encoder.
- **Num Bins RMS Acceleration:** Konfigurieren der Anzahl der Histogramm-Bins für die Analyse der Beschleunigung.
- **Threshold Shock:** Schwellwert für den RMS der Beschleunigung zur Detektion von Schocks.

Ausgangswerte

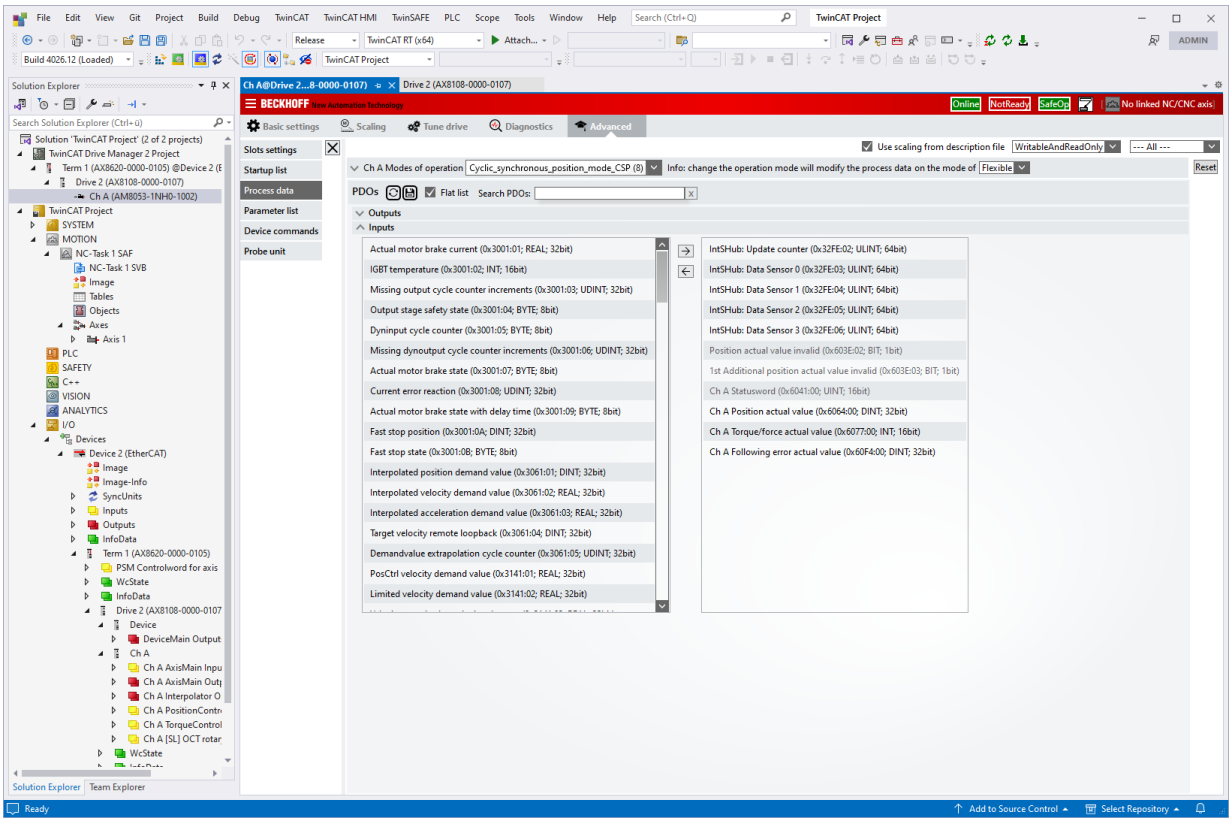
- **Humidity:** Minimal, Maximal und Mittelwert der Luftfeuchte innerhalb des Encoders in %.
- **Temperature:** Minimal, Maximal und Mittelwert der Temperatur innerhalb des Encoders in °C.
- **X RMS Acceleration:** Minimal, Maximal und Mittelwert des RMS der Beschleunigung in X-Richtung in g.
- **X Kurtosis Acceleration:** Minimal, Maximal und Mittelwert der Kurtosis der Beschleunigung in X-Richtung.
- **X Peak-Peak Acceleration:** Minimal, Maximal und Mittelwert des Spitze-Spitze-Werts der Beschleunigung in X-Richtung in g.
- **YRms Acceleration:** Minimal, Maximal und Mittelwert des RMS der Beschleunigung in Y-Richtung in g.
- **Y Kurtosis Acceleration:** Minimal, Maximal und Mittelwert der Kurtosis der Beschleunigung in Y-Richtung.
- **Y Peak-Peak Acceleration:** Minimal, Maximal und Mittelwert des Spitze-Spitze-Werts der Beschleunigung in Y-Richtung in g.
- **X Shock | Shock Detected:** `TRUE`, wenn der ein Schock in X-Richtung detektiert wird. Hierfür wird der Schwellwert des RMS der Beschleunigung überwacht.
- **X Shock | Count:** Anzahl der Schocks X-Richtung.
- **X Shock | Last Shock:** Zeitstempel des zuletzt detektierten Schocks in X-Richtung.
- **Y Shock | Shock Detected:** `TRUE`, wenn der ein Schock in Y-Richtung detektiert wird. Hierfür wird der Schwellwert des RMS der Beschleunigung überwacht.
- **Y Shock | Count:** Anzahl der Schocks Y-Richtung.
- **Y Shock | Last Shock:** Zeitstempel des zuletzt detektierten Schocks in Y-Richtung.
- **Total Shocks:** Anzahl der Schocks in X- und Y-Richtung
- **Sections Humidity:** Zeitliche Verteilung der Luftfeuchte innerhalb des Encoders. (Section Timer)
- **Sections Temperature:** Zeitliche Verteilung der Temperatur innerhalb des Encoders. (Section Timer)
- **X Rms Acceleration:** Verteilung der Beschleunigung in X-Richtung. (Histogramm)
- **Y Rms Acceleration:** Verteilung der Beschleunigung in Y-Richtung. (Histogramm)

Vorbereitung

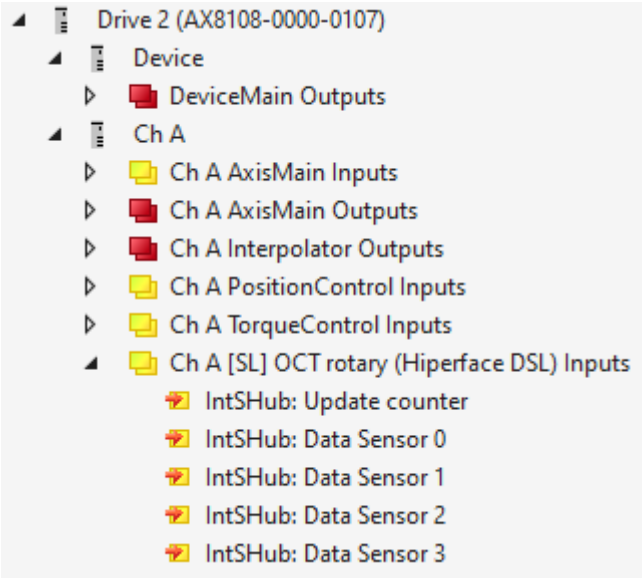
Damit die Sensorwerte für eine Analyse zur Verfügung stehen, müssen diese zum Prozessabbild des Antriebs hinzugefügt werden. Dieser Vorgang wird im Folgenden aufgezeigt.

1. Erstellen Sie ein TwinCAT-Projekt.
2. Scannen Sie die I/Os des TwinCAT-Projektes.
3. Fügen Sie zu der Solution ein Drive Manager-Projekt hinzu.
4. Wählen Sie den entsprechenden Kanal des Antriebs aus, an dem der Motor mit dem B/SSD-Fähigen Encoder angeschlossen ist.
5. Wählen Sie den Reiter **Advanced** an.
6. Im Unterpunkt **Process data** können weitere Prozessdaten hinzugefügt werden.
7. Fügen Sie folgende Prozessdaten hinzu:
   IntSHub: Update counter
   IntSHub: Data Sensor 0
   IntSHub: Data Sensor 1

IntSHub: Data Sensor 2
IntSHub: Data Sensor 3



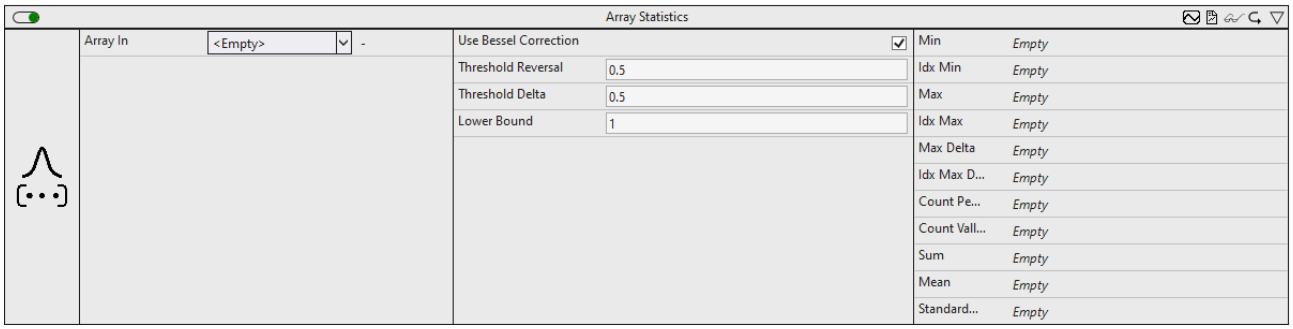
⇨ Nach dem Hinzufügen der Prozessdaten werden diese im Knoten I/O angezeigt und stehen für die Analyse zur Verfügung.



6.4.10 Analytics - Statistics

Die Algorithmen der Kategorie *Statistics* bieten Funktionen für die Datenanalyse auf Basis statistischer Methoden. Dazu gehören beispielsweise die Berechnung von Signal-Korrelationen und Regressionsanalysen.

6.4.10.1 Array Statistics



Der Algorithmus *Array Statistics* berechnet verschiedene statistische Größen auf Basis des Eingangs-Arrays.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Use Bessel Correction:** Wenn die Checkbox aktiviert ist, wird die Bessel‘sche Korrektur angewendet. Um bei Stichproben ein erwartungstreues Ergebnis zu erhalten, ist dieser Parameter zu aktivieren. Der Parameter ist nur für die Berechnung der Standardabweichung relevant.

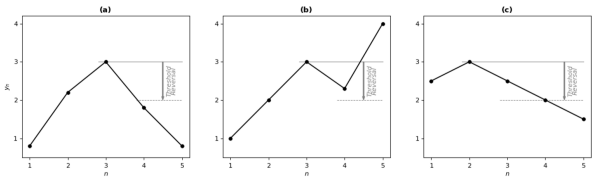
Die empirische Standardabweichung, ohne Bessel’sche Korrektur

$$s' = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}(x[n]-\overline{x})^2}$$

Die empirische Standardabweichung, mit Bessel’scher Korrektur

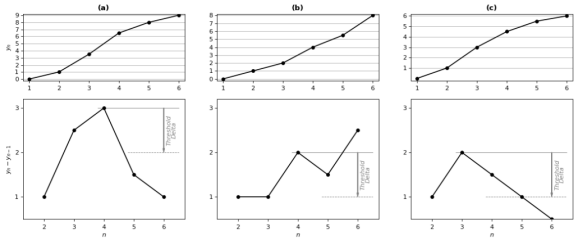
$$s = \sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}(x[n]-\overline{x})^2}$$

- **Threshold Reversal:** Schwellenwert für die Identifizierung von Umkehrungen. Umkehrungen werden nur erkannt, wenn ihre Differenz zur nächsten Umkehrung den Wert von Threshold Reversal überschreitet.
  Nachfolgend sind drei Beispiele für die Identifizierung von Peaks mit dem Parameter *Threshold Reversal* aufgeführt.
  (a) Der Wert y_3 wird direkt nach der Verarbeitung des Werts y_4 als Peak identifiziert, da die Differenz zwischen y_3 und y_4 größer ist als *Threshold Reversal*.
  (b) Der Wert y_3 wird nicht als Peak identifiziert, da die Differenz zwischen y_3 und y_4 kleiner ist als *Threshold Reversal* und die Kurve nach y_4 wieder ansteigt.
  (c) Der Wert y_2 wird nach der Verarbeitung des Werts y_5 als Peak identifiziert, da die Differenz zwischen y_2 und y_5 *Threshold Reversal* überschreitet. Der Wert y_2 kann vorher nicht als Peak identifiziert werden, da die Differenz zwischen y_2 und y_3 (y_4) kleiner als/gleich *Threshold Reversal* ist und nicht bekannt ist, ob die Werte weiter sinken.



- **Threshold Delta:** Schwellenwert für die Identifizierung der Delta-Maxima. Maxima der absoluten Differenz von zwei aufeinanderfolgenden Werten (Delta) werden nur erkannt, wenn die Differenz zwischen aufeinanderfolgenden Deltas *Threshold Delta* überschreitet.
  Nachfolgend sind drei Beispiele für die Identifizierung der Delta-Maxima mit dem Parameter *Threshold Delta* aufgeführt. Die oberen Diagramme zeigen die ursprünglichen Eingangssignale, die unteren das zugehörige Delta.
  (a) Der Wert y_4 wird nach der Verarbeitung des Werts y_5 als Maximum identifiziert, da die Differenz der zwei Deltas *Threshold Delta* überschreitet.

(b) Es wird kein Maximum identifiziert, da die Differenz zwischen den Deltas kleiner ist als *Threshold Delta*.
(c) Der Wert y_3 wird nach der Verarbeitung des Werts y_6 als Maximum identifiziert.



Unabhängig von *Threshold Delta* wird mindestens ein Maximum des Deltas zwischen zwei Umkehrungen erkannt.

- **Lower Bound:** Untere Grenze des Arrays

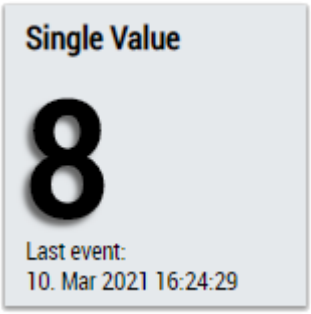
Ausgangswerte

- **Min:** Kleinster Wert im Eingangsarray.
- **Idx Min:** Array-Index von *Min*. Die Indexierung beginnt bei 1.
- **Max:** Größter Wert im Eingangsarray.
- **Idx Max:** Array-Index von *Max*. Die Indexierung beginnt bei 1.
- **Max Delta:** Maximum der absoluten Differenz zwischen zwei aufeinanderfolgenden Werten im Eingangsarray.
- **Idx Max Delta:** Array-Index von *Max Delta*. Die Indexierung beginnt bei 1.
- **Count Peaks:** Gesamtanzahl der identifizierten Peaks.
- **Count Valleys:** Gesamtanzahl der identifizierten Täler.
- **Sum:** Summe über das gesamte Eingangsarray.
- **Mean:** Mittelwert über das gesamte Eingangsarray.
- **Standard Deviation:** Standardabweichung über das gesamte Eingangsarray.

Standard HMI Controls

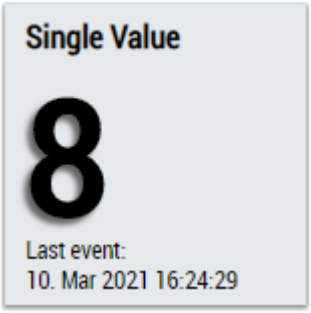
Für den Algorithmus Array Statistics sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SingleValue Control visualisiert den Ausgang Sum.

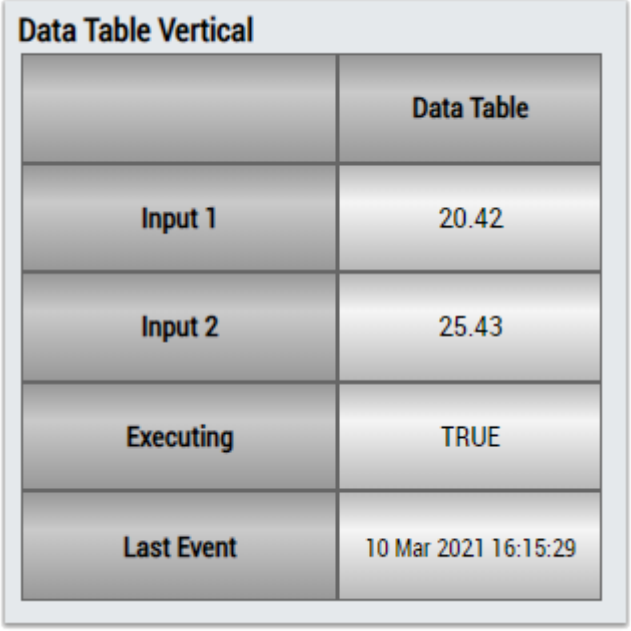


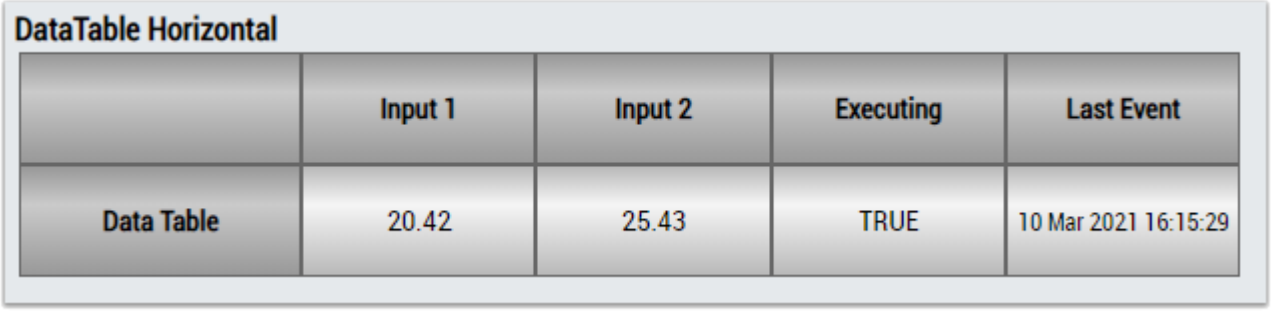
2. Das SingleValue Control visualisiert den Ausgang Mean.

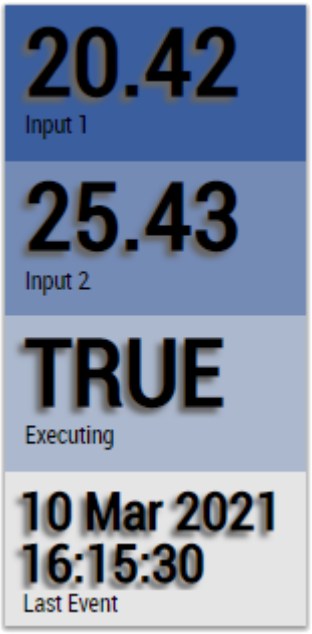
3. Das SingleValue Control visualisiert den Ausgang Standard Deviation.



4. Das Table Control oder Multivalue Control visualisiert die Ausgangswerte: Min, Max, Max Delta, CountPeaks Count Valleys, Sum, Mean, Standard Deviation.

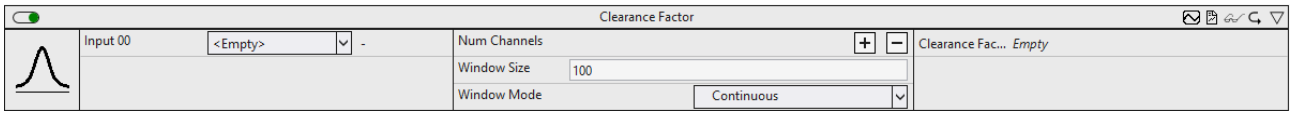






Alternativ können im Algorithmus Standard Deviation mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.10.2 Clearance Factor



Der Algorithmus *Clearance Factor* berechnet das gleichnamige Signal-Feature aus den Eingangswerten. Der Ausgangswert berechnet sich aus dem Verhältnis des Spitzenwertes des Eingangssignals zu dem quadrierten Mittelwert der Quadratwurzeln des absoluten Eingangssignals.

$$\text{Clearance Factor} = \frac{\max(|x|)}{\left(\frac{1}{N}\sum_{n=1}^{N}\sqrt{|x[n]|}\right)^2}$$

Die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogenen werden und die Art der Berechnung können konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Konfiguration der Anzahl unabhängiger Eingangs- und Ausgangskanäle.
- **Window Mode:** Verwendeter Fenstermodus. Beeinflusst die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogenen werden sowie den Zeitpunkt der Berechnungen.
  Continuous: Alle Eingangswerte seit dem Start des Algorithmus werden mit in die Berechnung einbezogen. Die Berechnung erfolgt zyklisch.
  Fix Window: Nur die letzten *N* Eingangswerte werden mit in die Berechnung einbezogen. Die Berechnung erfolgt alle *N* Aufrufe.
  Sliding Window: Nur die letzten *N* Eingangswerte werden mit in die Berechnung eingbezogen. Die Berechnung erfolgt zyklisch.
- **Window Size:** Abhängig von dem verwendeten Fenstermodus kann die Anzahl der Werte *N* konfiguriert werden, die mit in die Berechnung einbezogen werden. Beim Fenstermodus *Continuous* wird dieser Parameter ignoriert.

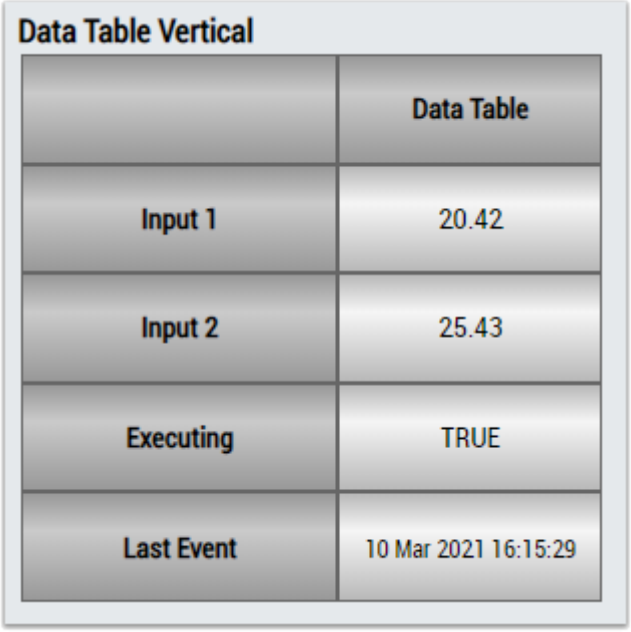
Ausgangswerte

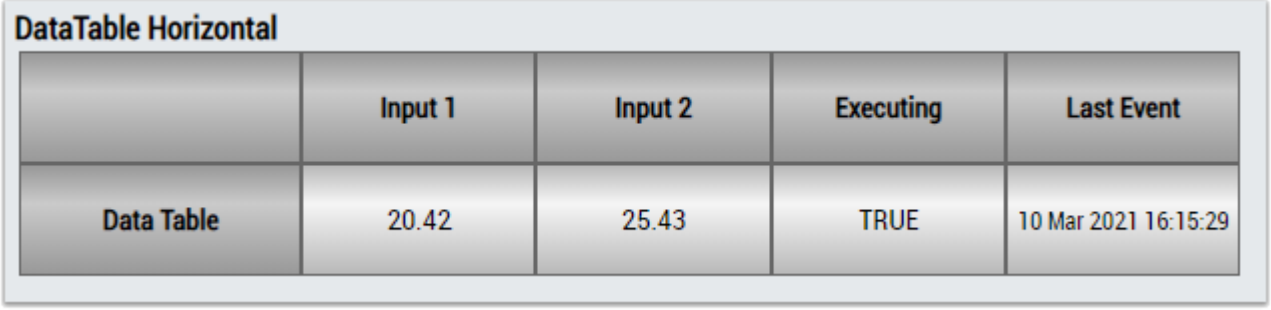
- **Clearance Factor 00..n:** Clearance-Faktor der Eingangskanäle.

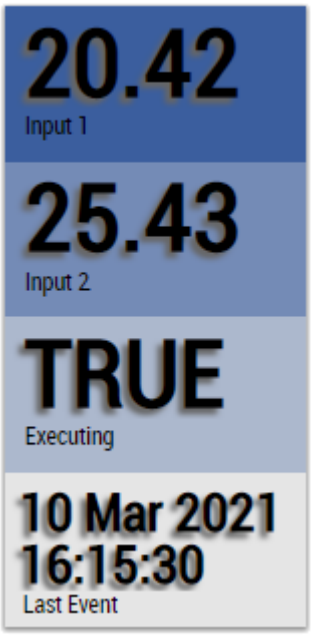
Standard HMI Controls

Für den Algorithmus *Clearance Factor* sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das Table Control oder Multivalue Control visualisieren alle Ausgangswerte.

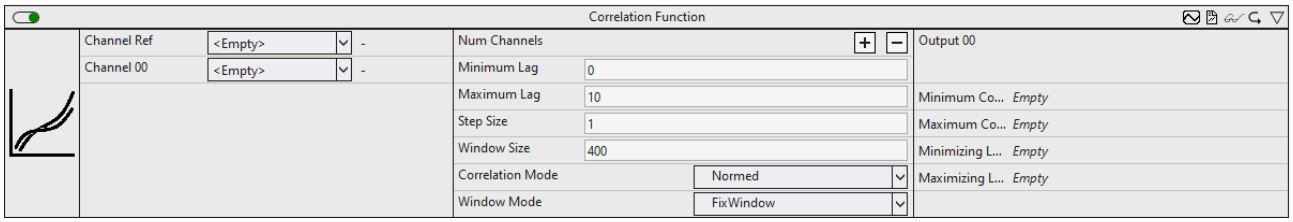






Alternativ können mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.10.3 Correlation Function



Der *Correlation Function* Baustein berechnet die diskrete Korrelationsfunktion zwischen einem Referenzsignal (Channel Ref) und einem oder mehreren weiteren Signalen (Channel 00, …, Channel 0n). Die Korrelationskoeffizienten werden für Zeitverschiebungen von *m* Zyklen zwischen den beiden Signalen berechnet, wobei der maximale und der minimale Wert für *m* durch die Parameter *Minimum Lag* (negative ganze Zahl) und *Maximum Lag* (positive ganze Zahl) begrenzt ist.

Durch den Parameter *StepSize* wird bestimmt, um wie viele Zyklen die Signale für die Berechnung zweier aufeinanderfolgender Korrelationskoeffizienten verschoben werden. D.h. *m* ist immer ein Vielfaches der *StepSize*. Dementsprechend sind für *Minimum Lag* und *Maximum Lag* auch nur Vielfache der *StepSize* zulässig. Ist die *StepSize* auf eins gesetzt und beispielsweise *Minimum Lag* auf -6 sowie *Maximum Lag* auf +4, so werden Korrelationskoeffizienten für Verschiebungen um -6, -5, -4, -3, -2, -1, 0, +1, +2, +3 und +4 Zyklen berechnet. Setzt man die *StepSize* auf zwei, werden Koeffizienten für Verschiebungen um -6, -4, -2, 0, +2 und +4 Zyklen berechnet.

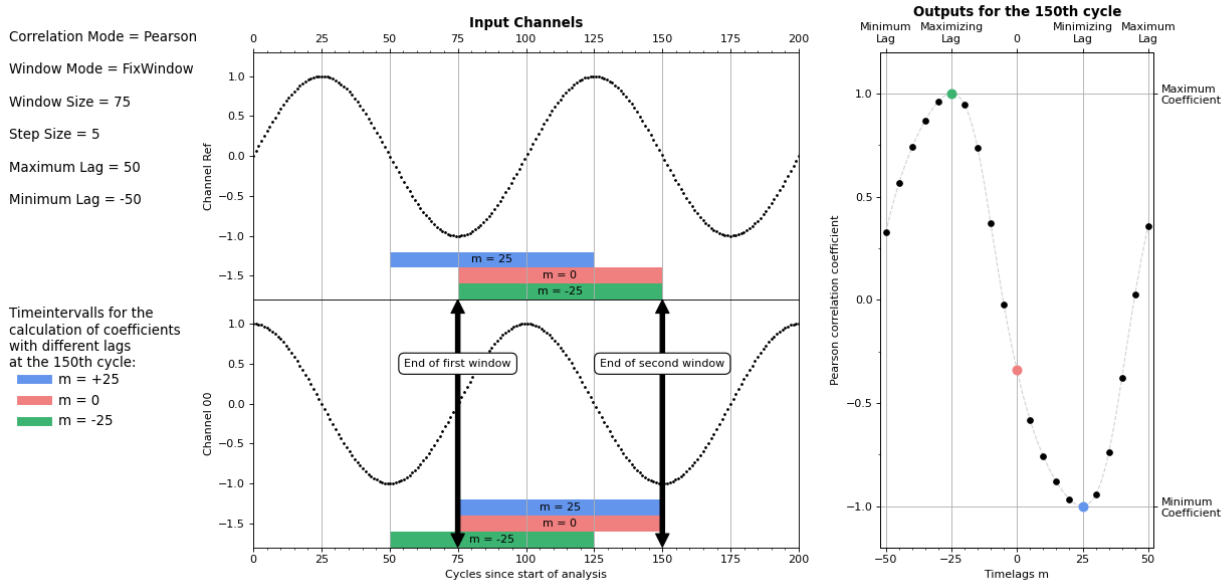
Die Koeffizienten können über verschiedene Zeitfenster berechnet werden. Diese werden durch den Parameter *Window Mode* gesetzt. Im *Continuous* Modus werden alle Werte seit Beginn der Analyse in die Berechnungen miteinbezogen. Im *SlidingWindow* Modus läuft die Berechnung kontinuierlich über die letzten, durch die *Window Size* gesetzte Anzahl von Zyklen. Im *FixWindow* Modus erfolgt die Berechnung auch über die durch die *Window Size* gesetzte Anzahl von Zyklen. Jedoch startet die Berechnung nach jeden *WindowSize* Zyklen neu und die Ausgangswerte werden nur beim Durchlaufen des letzten Zyklus eines Fensters aktualisiert.

Für *m* ungleich Null verschiebt sich das Fenster für das entsprechende Signal um *m* Zyklen. Damit bleibt die Anzahl von Werten, welche in die Berechnung der Koeffizienten mit einbezogen werden, für alle Werte von *m* gleich. Eine Ausnahme hierzu bilden nur die Ergebnisse aus den ersten Zyklen nach Start der Analyse. Ist die Anzahl der verstrichenen Zyklen kleiner als $|m|$, werden entsprechend weniger Werte in die Berechnung mit einbezogen.

Für positive Werte von *m* werden die Werte des Referenzsignals (Channel Ref) in einem Ringspeicher gespeichert, damit jeweils der vor *m* Zyklen eingegangene Wert des Referenzsignals mit den aktuellen Werten von Channel 00 bis Channel 0n verglichen werden kann. Dies entspricht einer Verschiebung des Referenzsignals in die Vergangenheit. Für negative Werte von *m* müsste das Referenzsignal dementsprechend in die Zukunft verschoben werden. Da dies offensichtlich nicht möglich ist, wird stattdessen das zweite Signal (Channel 00,..Channel0n) gespeichert und nach hinten verschoben.

Die Korrelationskoeffizienten können nach unterschiedlichen Berechnungsvorschriften berechnet werden. Diese wird durch den *Correlation Mode* bestimmt. In den Modi *Base* und *Normed* werden die Koeffizienten analog zu der Definition aus der Signalverarbeitung berechnet, welche die Korrelation über die Faltung berechnet. Im *Normed* – Modus werden die Koeffizienten zudem durch die Anzahl der Summanden geteilt. *Covariance* und *CovarianceBessel* berechnen die Kovarianz ohne und mit Bessel-Korrektur. Der Modus *Pearson* nutzt die Definition des in der Statistik üblichen Pearson Korrelationskoeffizienten. In den Konfigurationsoptionen sind die genauen Berechnungsvorschriften mathematisch aufgeführt. Hier bezeichnet x_n den Wert des Referenzsignals und y_n den Wert des zweiten Signals (jeweils Channel00, …, Channel0n) zum Zeitstempel t_n (entsprechend dem n-ten Zyklus seit Start der Analyse oder seit Reset, ausgenommen der Zyklen in denen *Enable Execution* = FALSE). Der Wert von *N* hängt von dem gewählten *WindowMode* ab. Für den *SlidingWindow*-Mode und *FixWindow*-Mode ist *N* gleich der *Window Size*, vorausgesetzt seit Start der Analyse sind bereits entsprechend viele Zyklen vergangen, sodass x_{n-N-m} (bzw. y_{n-N+m}) aufgenommen wurde, andernfalls reduziert sich *N* zu $n-m+1$ bzw. $n+m+1$. Zu beachten ist, dass im *FixWindow*-Mode die Ausgangswerte nur alle *Window Size* Zyklen aktualisiert werden. Im *Continuous*-Mode gilt stets $N = n+1$.

In der Abbildung sind beispielhaft für eine Konfiguration (*Correlation Mode* = *Pearson*, *Window Mode* = *FixWindow*, *Window Size* = 75, *Step Size* = 5, *Maximum Lag* = 50, *Minimum Lag* = -50) die Ausgangswerte des Bausteins für zwei Signale (Channel Ref und Channel 00) für einen bestimmten Zyklus (n=150) dargestellt. In den beiden linken Plots sind die Eingangssignale Channel Ref und Channel 00 im Verlauf der Zeit dargestellt. Im rechten Plot ist die diskrete Korrelationsfunktion (die Pearson-Korrelationskoeffizienten in Abhängigkeit von m) zu sehen. Es sind Koeffizienten für die Verschiebungen m= -50, -45, -40, -35, -30, -25, -20, -15, -10, -5, 0, +5, +10, +15, +20, +25, +30, +35, +40, +45, +50 dargestellt. Diese werden im Baustein als Array ausgegeben. Neben den beiden Parametern Minimum und Maximum Lag, sind auf der Abszisse die Ausgangswerte Minimizing und Maximizing Lag gekennzeichnet. Die dazugehörigen Koeffizienten Minimum und Maximum Coefficient, welche ebenfalls Ausgänge des Bausteins darstellen, sind auf der Ordinate markiert. Exemplarisch sind für die Verschiebungen m = -25, m = 0 und m =+25 in den Plots der Eingangskanäle (links) die Zeitbereiche farblich hervorgehoben, welche in die Berechnung des jeweiligen Koeffizienten eingehen. Im rechten Plot sind die entsprechenden Punkte jeweils eingefärbt.



Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Kanäle, die mit dem Referenzsignal korreliert werden. Diese kann über Add/Remove Channel eingestellt werden
- **Maximum Lag:** Gibt die maximale Anzahl an Zyklen an, um die die beiden Signale für die Berechnung der Korrelation zueinander verschoben werden. Dies ist eine positive ganze Zahl.
- **Minimum Lag:** Gibt die minimale Anzahl an Zyklen an, um die die beiden Signale für die Berechnung der Korrelation zueinander verschoben werden. Dies ist eine negative ganze Zahl.
- **Step Size:** Gibt an, um wie viele Zyklen die Signale für die Berechnung zweier aufeinanderfolgender Korrelationskoeffizienten verschoben werden.
- **Window Size:** Gibt für die Window Mode SlidingWindow und FixWindow die Anzahl der Zyklen an, über die die Koeffizienten berechnet werden. Für den Window Mode *Continuous* hat die Fensterung keine Auswirkung und kann auch nicht gesetzt werden. Im *SlidingWindow* Mode, werden zusätzlich zu Maximum Lag Werten vom Referenz Channel und Minimum Lag Werten für Channel 00 bis Channel 0n, Window Size Werte für alle Channel zwischengespeichert. Die Größe des Router Speichers sollte beim Setzen dieser Parameter also berücksichtigt werden.
- **Correlation Mode:** Die Korrelationskoeffizienten werden nach einer der folgenden Definitionen berechnet:
- *Base*:

$$C_{xy}[m,t_n] = \begin{cases} \sum_{i=0}^{N-1} x_{n-i-m} \cdot y_{n-i}, & m \geq 0 \\ \sum_{i=0}^{N-1} x_{n-i} \cdot y_{n-i+m}, & m < 0 \end{cases}$$

- *Normed*:

$$\hat{C}_{xy}[m,t_n] = \begin{cases} \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i-m} \cdot y_{n-i}, & m \geq 0 \\ \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i} \cdot y_{n-i+m}, & m < 0 \end{cases}$$

- *Covariance*:

$$cov_{xy}[m,t_n] = \begin{cases} \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i-m} \cdot y_{n-i} - \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i-m} \cdot \frac{1}{N}\sum_{i=0}^{N-1} y_{n-i}, & m \geq 0 \\ \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i} \cdot y_{n-i+m} - \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i} \cdot \frac{1}{N}\sum_{i=0}^{N-1} y_{n-i+m}, & m < 0 \end{cases}$$

- *CovarianceBessel*:

$$\widetilde{cov}_{xy}[m,t_n] = \frac{N}{N-1} cov_{xy}[m,t_n]$$

- *Pearson:*

$$\rho_{xy}[m,t_n] = \frac{cov(x,y)[m,t_n]}{\sigma_x[m,t_n]\sigma_y[m,t_n]}, where \;\; \sigma_x^2[m,t_n] = cov_{xx}[m,t_n]$$

- **Window Mode:** Gibt an über welche Art von Fenster die Koeffizienten berechnet werden:

Continuous: Alle Werte, seit Start der Analyse werden mit gleicher Gewichtung in die Analyse miteinbezogen.

SlidingWindow: Die Berechnung wird über ein Fenster der Größe Window Size berechnet. Dabei fließen immer die aktuellen Werte in die Analyse ein und es werden mit jedem Zyklus Ausgänge aktualisiert.

FixWindow: Die Ausgänge werden alle Window Size Zyklen aktualisiert und über ein Fenster der Länge Window Size berechnet.

Ausgangswerte

- **Output00 .. Output0n:** Zeigt für jeden Eingang (Channel00, …, Channel0n) die Koeffizienten für die Verschiebungen *m* = *Minimum Lag*, *m* = *Minimum Lag* + *Step Size*, …, *m* = - *Step Size*, *m* = 0, *m* =+*Step Size*,…, m = *Maximum Lag* als Array.
- **Minimizing Lag00..Minimizing Lag0n:** Gibt für jeden Kanal die Verschiebung an, für die der Koeffizient minimal wird.
- **Minimum Coefficient00..Minimum Coefficient0n:** Gibt für jeden Kanal den minimalen Koeffizienten an.

- **Maximizing Lag00..Maximizing Lag0n:**Gibt für jeden Kanal die Verschiebung an, für die der Koeffizient maximal wird.
- **Maximum Coefficient 00..00.. Maximum Coefficient0n:** Gibt für jeden Kanal den maximalen Koeffizienten an.

Standard HMI Controls

Für den Algorithmus Correlation Function sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

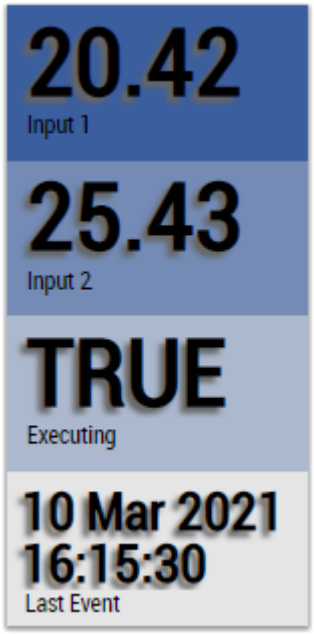
1. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Minimaler Koeffizient, maximaler Koeffizient, minimaler lag, maximaler lag und den Koeffizienten Array.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

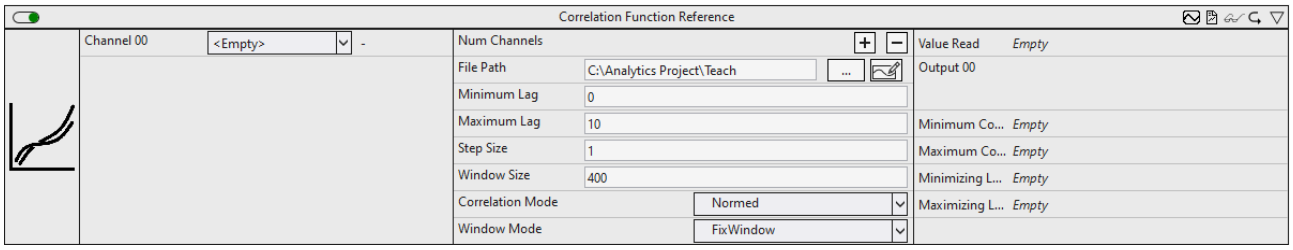
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Correlation Function mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.10.4 Correlation Function Reference



Der *Correlation Function Reference* Baustein berechnet die diskrete Korrelationsfunktion zwischen einem aufgenommenen Signal (im Folgenden Referenzsignal), welches aus einer tcab-Datei eingelesen wird, und einem oder mehreren Eingangssignalen (Channel 00, …, Channel 0n).

Die Korrelationskoeffizienten werden für Zeitverschiebungen von *m* Zyklen zwischen den beiden Signalen berechnet, wobei der maximale und der minimale Wert für *m* durch die Parameter *Minimum Lag* (negative ganze Zahl) und *Maximum Lag* (positive ganze Zahl) begrenzt ist.

Durch den Parameter *Step Size* wird bestimmt, um wie viele Zyklen die Signale für die Berechnung zweier aufeinanderfolgender Korrelationskoeffizienten verschoben werden. D.h. *m* ist immer ein Vielfaches der *Step Size*. Dementsprechend sind für *Minimum Lag* und *Maximum Lag* auch nur Vielfache der *Step Size* zulässig. Ist die *Step Size* auf eins gesetzt und beispielsweise *Minimum Lag* auf -6 sowie *Maximum Lag* auf +4, so werden Koeffizienten für Verschiebungen um -6, -5, -4, -3, -2, -1, 0, +1, +2, +3 und +4 Zyklen berechnet. Setzt man die *Step Size* auf zwei, werden Koeffizienten für Verschiebungen um -6, -4, -2, 0, +2 und +4 Zyklen berechnet.

Ab dem Start der Analyse wird von dem eingelesenen Signal pro Zyklus ein Wert verarbeitet. Ist das Ende der Datei erreicht, wird wieder mit dem ersten Wert der Datei begonnen. Das eingelesene Signal wird also als periodisch angenommen. Möchte man nur bestimmte Zeitabschnitte des Eingangssignals mit dem Referenzsignal korrelieren, kann man dies über die Eingänge *Enable Execution* und *Reset* sowie über den Ausgang *New Result* steuern.

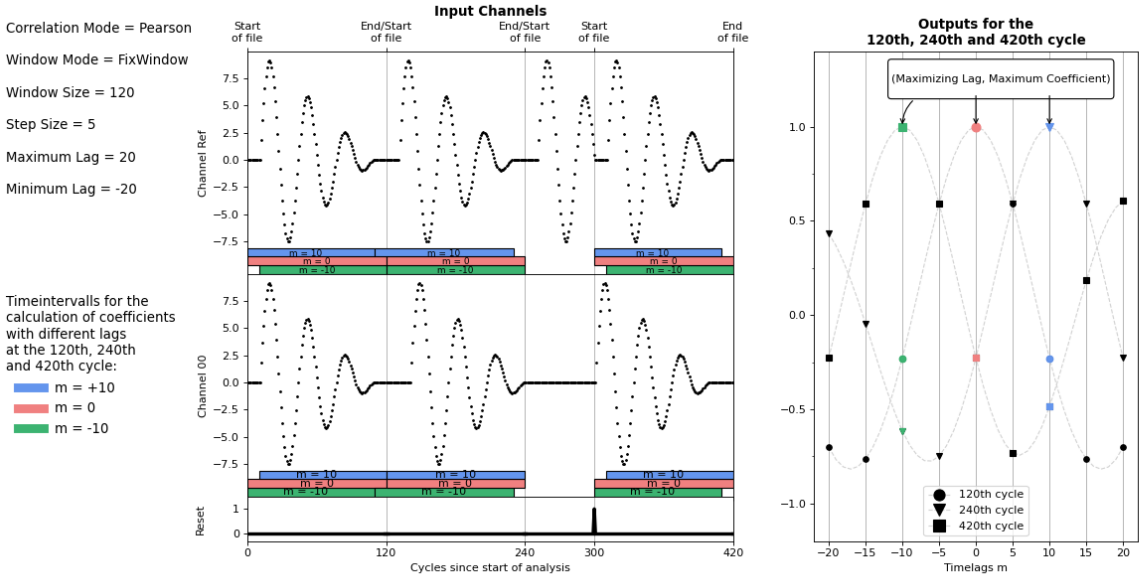
Die Korrelationskoeffizienten können über verschiedene Zeitfenster berechnet werden. Diese werden durch den Parameter *Window Mode* gesetzt. Im *Continuous* Modus werden alle Werte seit Beginn der Analyse in die Berechnungen miteinbezogen. Im *SlidingWindow* Modus läuft die Berechnung kontinuierlich über die letzten, durch die *Window Size* gesetzte Anzahl von Zyklen. Im *FixWindow* Modus erfolgt die Berechnung über die Länge des Referenzsignals und die Ausgangswerte werden immer zum Ende der aufgenommenen Sequenz aktualisiert.

Für *m* ungleich Null verschiebt sich das Fenster für das entsprechende Signal um *m* Zyklen. Damit bleibt die Anzahl von Werten, welche in die Berechnung der Koeffizienten mit einbezogen werden, für alle Werte von *m* gleich. Eine Ausnahme hierzu bilden nur die Ergebnisse aus den ersten Zyklen nach Start der Analyse. Ist die Anzahl der verstrichenen Zyklen kleiner als |*m*|, werden entsprechend weniger Werte in die Berechnung mit einbezogen.

Für positive Werte von *m* werden die Werte des Referenzsignals (Channel Ref) in einem Ringspeicher gespeichert, damit jeweils der vor *m* Zyklen eingegangene Wert des Referenzsignals mit den aktuellen Werten von Channel 00 bis Channel 0n verglichen werden kann. Dies entspricht einer Verschiebung des Referenzsignals in die Vergangenheit. Für negative Werte von *m* müsste das Referenzsignal dementsprechend in die Zukunft verschoben werden. Da dies offensichtlich nicht möglich ist, wird stattdessen das zweite Signal (Channel 00,..Channel0n) gespeichert und nach hinten verschoben.

Die Korrelationskoeffizienten können nach unterschiedlichen Berechnungsvorschriften berechnet werden. Diese wird durch den *Correlation Mode* bestimmt. In den Modi *Base* und *Normed* werden die Koeffizienten analog zu der Definition aus der Signalverarbeitung berechnet, welche die Korrelation über die Faltung berechnet. Im *Normed* – Modus werden die Koeffizienten zudem durch die Anzahl der Summanden geteilt. *Covariance* und *CovarianceBessel* berechnen die Kovarianz ohne und mit Bessel-Korrektur. Der Modus *Pearson* nutzt die Definition des in der Statistik üblichen Pearson Korrelationskoeffizienten. In den Konfigurationsoptionen sind die genauen Berechnungsvorschriften mathematisch aufgeführt. Hier bezeichnet x_n den Wert des Referenzsignals und y_n den Wert des Eingangssignals (jeweils Channel00, …, Channel0n) zum Zeitstempel t_n (entsprechend dem n-ten Zyklus seit Start der Analyse oder seit Reset, ausgenommen der Zyklen in denen *Enable Execution* = FALSE). Der Wert von *N* hängt von dem gewählten *WindowMode* ab. Für den *SlidingWindow*-Mode und *FixWindow*-Mode ist *N* gleich der *Window Size*, vorausgesetzt seit Start der Analyse sind bereits entsprechend viele Zyklen vergangen, sodass x_{n-N-m} (bzw. y_{n-N+m}) aufgenommen wurde, andernfalls reduziert sich *N* zu *n-m+1* bzw. *n+m+1*. Im FixWindow-Mode ist die Window Size nicht manuell zu setzen, sondern entspricht der Länge des eingelesenen Signalabschnitts (Referenzsignal). und die Ausgangswerte werden zum Ende des Signalabschnitts aktualisiert. Im *Continuous*-Mode gilt stets *N = n+1*.

In der Abbildung sind beispielhaft für eine Konfiguration (*Correlation Mode* = *Pearson*, *Window Mode* = *FixWindow*, *Window Size* = 120 (= Anzahl an Werten in der Datei), *Step Size* = 5, *Maximum Lag* = 20, *Minimum Lag* = -20) und einen Eingangskanal die verschiedenen Eingangs- und Ausgangswerte des Bausteins dargestellt. Auf der linken Seite sind in den beiden unteren Plots die Eingangssignale Channel 00 und *Reset* abgebildet, darüber ist auf der gleichen Zeitachse die eingelesene Sequenz, gemäß ihrer Verarbeitung, abgebildet. Zu Beginn der Analyse wird begonnen das Signal einzulesen. Ist der letzte Wert aus der Datei im 120ten (bzw. 240ten) Zyklus verarbeitet, wird erneut mit dem ersten begonnen. Im 300ten Zyklus wird ein *Reset* durchgeführt und es wird erneut mit dem ersten Wert der Datei begonnen. Zudem werden alle Werte aus den internen Speichern gelöscht und die Berechnung der Koeffizienten beginnt erneut. Beispielsweise wurde hier erkannt, dass Channel 00 in den vorherigen Zyklen keine gültigen Werte enthielt und jetzt die Schwingung erneut angeregt wurde. In diesem Bereich könnte auch durch *Enable Execution* = FALSE die Analyse unterbrochen werden. Da *WindowMode* = *FixWindow*, werden die Ausgänge nur im 120ten, 240ten und 420ten Zyklus aktualisiert. Die entsprechenden Koeffizienten (von Ausgang Output00) sind im rechten Plot dargestellt. An den Wertepaaren (*Maximizing Lag*, *Maximum Coefficient*) kann man jeweils ablesen, um wie viel das Eingangssignal zum Referenzsignal verschoben ist. Beispielsweise ist im 420ten Zyklus (*Maximizing Lag*, *Maximum Coefficient*) = (-10,1). Das heißt, hätte der *Reset* 10 Zyklen früher stattgefunden, hätten das Referenzsignal und Channel 00 in diesem Fenster exakt übereinander gepasst.



Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Kanäle, die mit dem Referenzsignal korreliert werden. Diese kann über Add/Remove Channel eingestellt werden
- **File Path:** Pfad zur Datendatei.
- **Maximum Lag:** Gibt die maximale Anzahl an Zyklen an, um die die beiden Signale für die Berechnung der Korrelation zueinander verschoben werden. Dies ist eine positive ganze Zahl.
- **Minimum Lag:** Gibt die minimale Anzahl an Zyklen an, um die die beiden Signale für die Berechnung der Korrelation zueinander verschoben werden. Dies ist eine negative ganze Zahl.
- **Step Size:** Gibt an, um wie viele Zyklen die Signale für die Berechnung zweier aufeinanderfolgender Korrelationskoeffizienten verschoben werden.
- **Window Size:** Gibt für die Window Mode *SlidingWindow* und *FixWindow* die Anzahl der Zyklen an, über die die Koeffizienten berechnet werden. Für den Window Mode *Continuous* hat die Fensterung keine Auswirkung und kann auch nicht gesetzt werden. Im *SlidingWindow* Mode, werden zusätzlich zu Maximum Lag Werten vom Referenz Channel und Minimum Lag Werten für Channel 00 bis Channel 0n, Window Size Werte für alle Channel zwischengespeichert. Die Größe des Routerspeichers sollte beim Setzen dieser Parameter also berücksichtigt werden.
- **Correlation Mode:** Die Korrelationskoeffizienten werden nach einer der folgenden Definitionen berechnet:
- *Base*:

$$C_{xy}[m, t_n] = \begin{cases} \sum_{i=0}^{N-1} x_{n-i-m} \cdot y_{n-i}, & m \geq 0 \\ \sum_{i=0}^{N-1} x_{n-i} \cdot y_{n-i+m}, & m < 0 \end{cases}$$

- *Normed:*

$$\hat{C}_{xy}[m,t_n] = \begin{cases} \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i-m} \cdot y_{n-i}, & m \geq 0 \\ \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i} \cdot y_{n-i+m}, & m < 0 \end{cases}$$

- *Covariance:*

$$cov_{xy}[m,t_n] = \begin{cases} \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i-m} \cdot y_{n-i} - \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i-m} \cdot \frac{1}{N}\sum_{i=0}^{N-1} y_{n-i}, & m \geq 0 \\ \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i} \cdot y_{n-i+m} - \frac{1}{N}\sum_{i=0}^{N-1} x_{n-i} \cdot \frac{1}{N}\sum_{i=0}^{N-1} y_{n-i+m}, & m < 0 \end{cases}$$

- *CovarianceBessel:*

$$\widetilde{cov}_{xy}[m,t_n] = \frac{N}{N-1} cov_{xy}[m,t_n]$$

- *Pearson:*

$$\rho_{xy}[m,t_n] = \frac{cov(x,y)[m,t_n]}{\sigma_x[m,t_n]\sigma_y[m,t_n]}, where \;\; \sigma_x^2[m,t_n] = cov_{xx}[m,t_n]$$

- **Window Mode:** Gibt an über welche Art von Fenster die Koeffizienten berechnet werden:

Continuous: Alle Werte, seit Start der Analyse werden mit gleicher Gewichtung in die Analyse miteinbezogen.

SlidingWindow: Die Berechnung wird über ein Fenster der Größe Window Size berechnet. Dabei fließen immer die aktuellen Werte in die Analyse ein und es werden mit jedem Zyklus Ausgänge aktualisiert.

FixWindow: Die Koeffizienten werden über die gesamte Länge des eingelesenen Signalabschnitts berechnet und die Ausgänge jeweils nach verarbeiten des letzten Wertes aktualisiert.

Ausgangswerte

- **Output00 .. Output0n:** Zeigt für jeden Eingang (Channel00, …, Channel0n) die Koeffizienten für die Verschiebungen *m* = *Minimum Lag*, *m* = *Minimum Lag* + *Step Size*, …, *m* = - *Step Size*, *m* = 0, *m* =+*Step Size*,…, m = *Maximum Lag* als Array.
- **Minimizing Lag00..Minimizing Lag0n:** Gibt für jeden Kanal die Verschiebung an, für die der Koeffizient minimal wird.
- **Minimum Coefficient00..Minimum Coefficient0n:** Gibt für jeden Kanal den minimalen Koeffizienten an.
- **Maximizing Lag00..Maximizing Lag0n:**Gibt für jeden Kanal die Verschiebung an, für die der Koeffizient maximal wird.
- **Maximum Coefficient 00..00.. Maximum Coefficient0n:** Gibt für jeden Kanal den maximalen Koeffizienten an.

Standard HMI Controls

Für den Algorithmus Correlation Function Reference sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

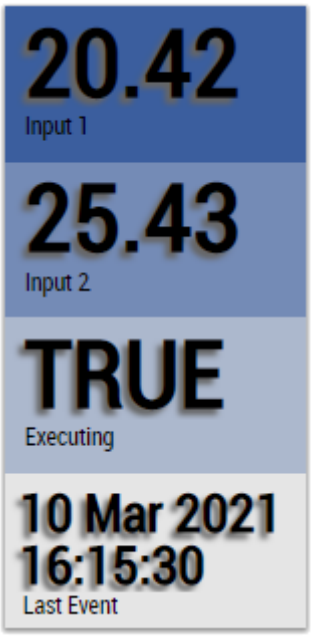
1. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Gelesener Wert, minimaler Koeffizient, maximaler Koeffizient, minimaler lag, maximaler lag und den Koeffizientenarray.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

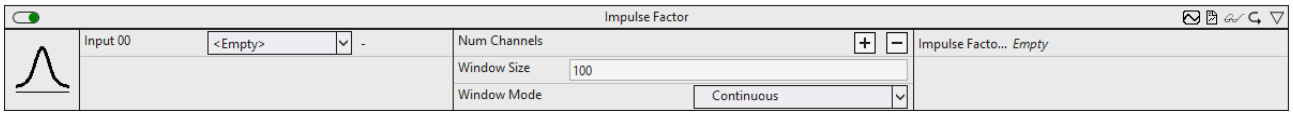
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Correlation Function Reference mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.10.5 Impulse Factor



Der Algorithmus *Impulse Factor* berechnet das gleichnamige Signal-Feature aus den Eingangswerten. Der Ausgangswert berechnet sich aus dem Verhältnis des Spitzenwertes des Eingangssignals zu dem Mittelwert des Eingangssignals.

$$\text{Impulse Factor} = \frac{\max(|x|)}{\frac{1}{N}\sum_{n=1}^{N}|x[n]|}$$

Die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogenen werden und die Art der Berechnung können konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Konfiguration der Anzahl unabhängiger Eingangs- und Ausgangskanäle.
- **Window Mode:** Verwendeter Fenstermodus. Beeinflusst die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogenen werden sowie den Zeitpunkt der Berechnungen.
  Continuous: Alle Eingangswerte seit dem Start des Algorithmus werden mit in die Berechnung einbezogen. Die Berechnung erfolgt zyklisch.
  Fix Window: Nur die letzten *N* Eingangswerte werden mit in die Berechnung einbezogen. Die Berechnung erfolgt alle *N* Aufrufe.
  Sliding Window: Nur die letzten *N* Eingangswerte werden mit in die Berechnung eingbezogen. Die Berechnung erfolgt zyklisch.
- **Window Size:** Abhängig von dem verwendeten Fenstermodus kann die Anzahl der Werte *N* konfiguriert werden, die mit in die Berechnung einbezogen werden. Beim Fenstermodus *Continuous* wird dieser Parameter ignoriert.

Ausgangswerte

- **Impulse Factor 00..n:** Impuls-Faktor der Eingangskanäle.

Standard HMI Controls

Für den Algorithmus *Impulse Factor* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

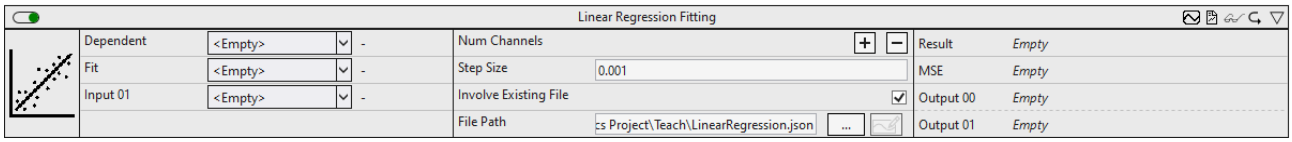
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

Abb. 3:

Alternativ können mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.10.6 Linear Regression Fitting



Der Baustein Linear Regression Fitting nähert eine Variable (den Input Dependent) durch die Linearkombination mehrerer anderer Variablen (Input 01 … Input 0n) an. Dies geschieht durch das inkrementelle stochastische Gradientenverfahren. Zum Ende der Analyse werden die berechneten Koeffizienten in eine Datei geschrieben.

Die Linearkombination ist gegeben durch folgende Gleichung:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i \times \text{Input 0i}$$

In jedem Zyklus werden die Werte für β_0 bis β_n mit folgender Vorschrift neu berechnet:

$$\beta_i = \begin{cases} \beta_i - \gamma \times \text{Input 0i} \times (y - \text{Dependent}), & i = 1, 2, \ldots, n \\ \beta_i - \gamma \times (y - \text{Dependent}), & i = 0 \end{cases}$$

Dies entspricht der Minimierung der quadratischen Abweichung der berechneten Werte y (vom Baustein als

Result ausgegeben) von dem dazugehörigen Eingangswert Dependent. Der Parameter γ entspricht der Step Size und gibt an, wie stark die Parameter angeglichen werden. Je größer der Wert, desto schneller nähern sich die Koeffizienten einem lokalen Optimum an. Wird der Wert jedoch zu groß gewählt, kann es sein, dass der Algorithmus nicht konvergiert.

Typischerweise werden zunächst mit dem Baustein Linear Regression Fitting die Gewichte für die Regression einer Zielvariable angepasst. Anschließend kann mithilfe des Bausteins Linear Regression Inference und den angepassten Gewichten die Zielvariable anhand der Eingangsvariablen vorhergesagt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Kanäle, die mit dem Referenzsignal korreliert werden. Diese kann über Add/Remove Channel eingestellt werden
- **File Path:** Gibt den Dateipfad für die Datei an, in der die Koeffizienten nach Ende des Annäherungsprozesses gespeichert werden. Dateityp: *json* oder *tas*.
- **Involve Existing File:** Wenn TRUE, werden beim Start der Analyse die Werte für die Koeffizienten aus der Datei eingelesen und anschließend weiter angeglichen. Wenn FALSE, werden alle Koeffizienten zu Beginn auf Null gesetzt.
- **Step Size:** Gibt an, wie stark die Koeffizienten nach jeder neuen Berechnung angepasst werden.
- **Bias (optional):** Wenn FALSE, wird der Bias Output 00 auf Null gesetzt und nicht weiter angenähert.
- **Mini Batch Size (optional):** Gibt an über wie viele Zyklen der MSE berechnet werden soll, bevor die Koeffizienten anhand diesem angepasst werden.

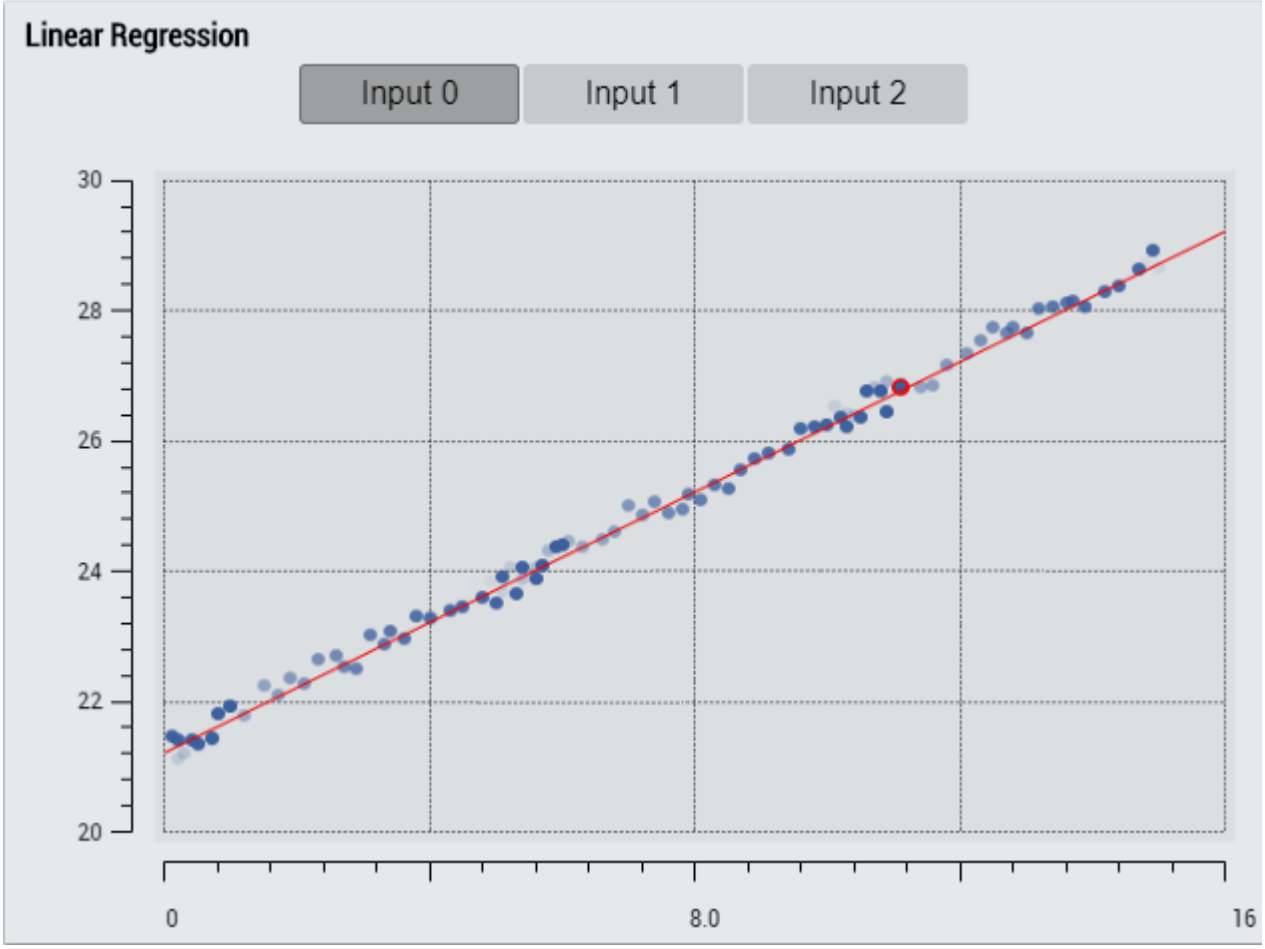
Ausgangswerte

- **MSE:** Gibt den MSE (mean squared error) zwischen errechnetem Wert *Result* und Eingangswert *Dependent* an.
- **Result:** Gibt den angenäherten Wert für die Inputs des aktuellen Zyklus mit den daraus aktualisierten Koeffizienten aus.
- **Output00 .. Output0n:** Zeigt die berechneten Koeffizienten.

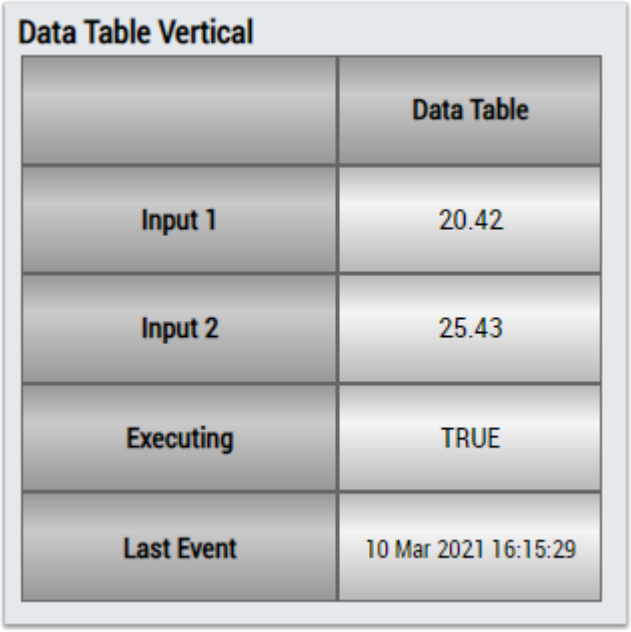
Standard HMI Controls

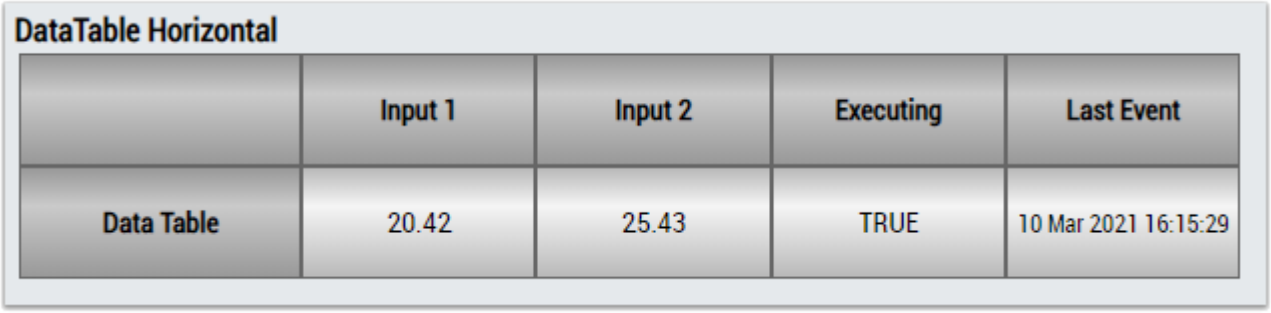
Für den Algorithmus Linear Regression Fitting sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

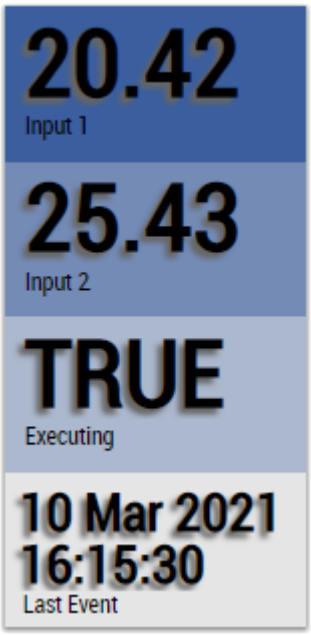
1. Das Linear Regression Control visualisiert die Inputs und die berechnete Regressionsgerade. Über die Buttons kann der Eingangskanal ausgewählt werden, der auf der x-Achse angezeigt werden soll. Die y-Achse zeigt die Zielwerte der Regression. Ein neuer Punkt wird rot umrandet, nach und nach verblassen alte Punkte.



2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: MSE, Ergebnis, Output-Koeffizienten.

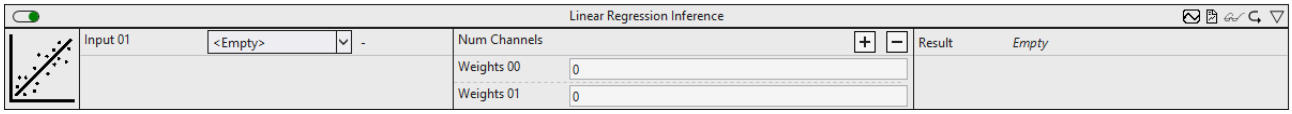






Alternativ können im Algorithmus Linear Regression Curve Fitting mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.10.7 Linear Regression Inference



Der Baustein Linear Regression Inference berechnet die Linearkombination der Inputs (Input 01 .. Input 0n) mit den Koeffizienten (Weights 00.. Weights 0n).

$$\text{Result} = \text{Weights 00} + \sum_{i=1}^{n} \text{Weights 0i} \times \text{Input 0i}$$

Die Parameter Weights 00 bis Weights 0n können entweder manuell gesetzt werden, oder über eine Datei, wie sie vom Baustein Linear Regression Fitting erzeugt wird über Drag / Drop auf das Parameterfeld automatisch gesetzt werden. Typischerweise werden zunächst mit dem Baustein Linear Regression Fitting die Gewichte für die Regression einer Zielvariable angepasst. Anschließend kann mithilfe des Bausteins Linear Regression Inference und den angepassten Gewichten die Zielvariable anhand der Eingangsvariablen vorhergesagt werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Kanäle, die mit dem Referenzsignal korreliert werden. Diese kann über Add/Remove Channel eingestellt werden
- **Weights00 .. Weights 0n:** Gibt die Koeffizienten an, welche die zu berechnende Linearkombination bestimmen.
- **File Path (optional):** Dateipfad zu einer Datei mit zuvor ermittelten Parametern. Die Datei wird durch das Modul *Linear Regression Fitting* erstellt. Dateityp: *json oder tas*.

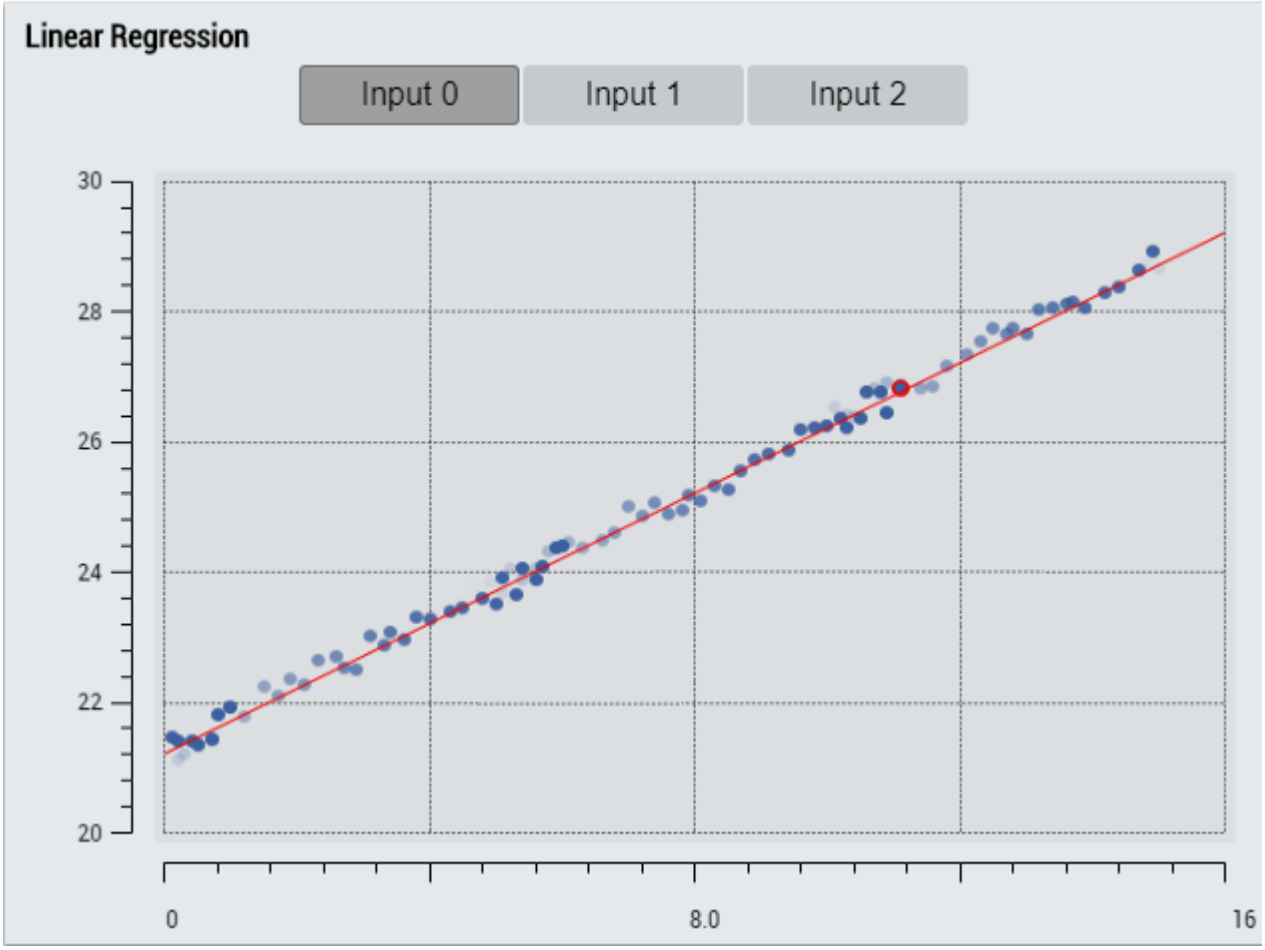
Ausgangswerte

- **Result:** Gibt den aus der Linearkombination berechneten Wert an.

Standard HMI Controls

Für den Algorithmus Linear Regression Inference sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

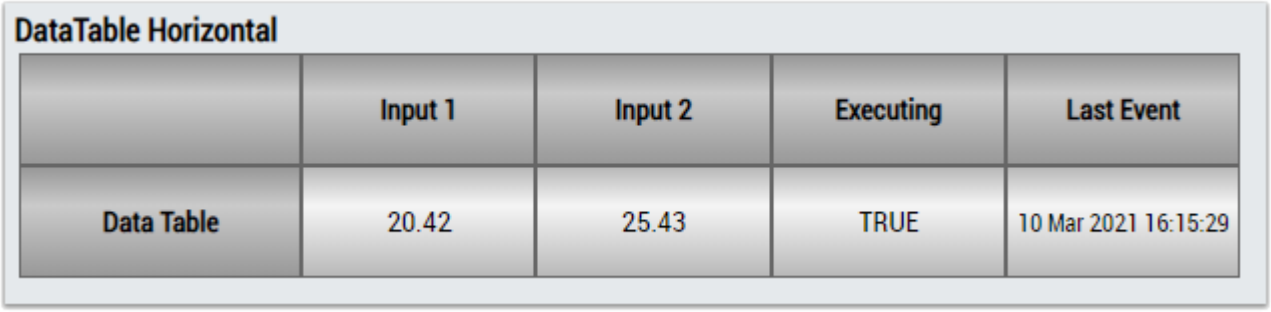
1. Das Linear Regression Control visualisiert die Inputs und die berechnete Regressionsgerade. Über die Buttons kann der Eingangskanal ausgewählt werden, der auf der x-Achse angezeigt werden soll. Die y-Achse zeigt die Zielwerte der Regression. Ein neuer Punkt wird rot umrandet, nach und nach verblassen alte Punkte.



2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: MSE, Ergebnis, Output-Koeffizienten.

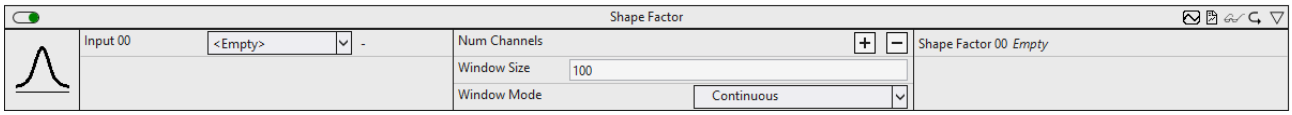
Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29



Alternativ können im Algorithmus Linear Regression Inference mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.10.8 Shape Factor



Der Algorithmus *Shape Factor* berechnet das gleichnamige Signal-Feature aus den Eingangswerten. Der Ausgangswert berechnet sich aus dem Verhältnis des quadratischen Mittels (RMS) des Eingangssignals zu dem Mittelwert der Absolutwerte des Eingangssignals.

$$\text{Shape Factor} = \frac{x_{rms}}{\frac{1}{N}\sum_{n=1}^{N}|x[n]|}$$

Die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogenen werden und die Art der Berechnung können konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Konfiguration der Anzahl unabhängiger Eingangs- und Ausgangskanäle.
- **Window Mode:** Verwendeter Fenstermodus. Beeinflusst die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogenen werden sowie den Zeitpunkt der Berechnungen.
  Continuous: Alle Eingangswerte seit dem Start des Algorithmus werden mit in die Berechnung

einbezogen. Die Berechnung erfolgt zyklisch.
Fix Window: Nur die letzten *N* Eingangswerte werden mit in die Berechnung einbezogen. Die Berechnung erfolgt alle *N* Aufrufe.
Sliding Window: Nur die letzten *N* Eingangswerte werden mit in die Berechnung eingbezogen. Die Berechnung erfolgt zyklisch.

- **Window Size:** Abhängig von dem verwendeten Fenstermodus kann die Anzahl der Werte *N* konfiguriert werden, die mit in die Berechnung einbezogen werden. Beim Fenstermodus *Continuous* wird dieser Parameter ignoriert.

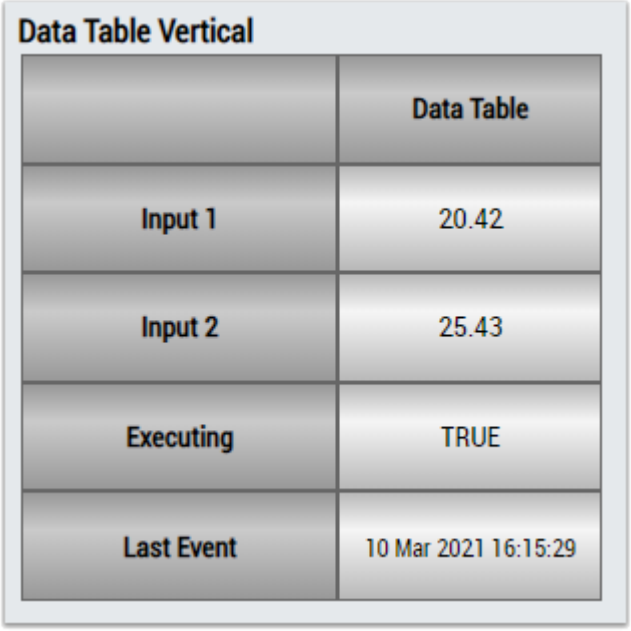
Ausgangswerte

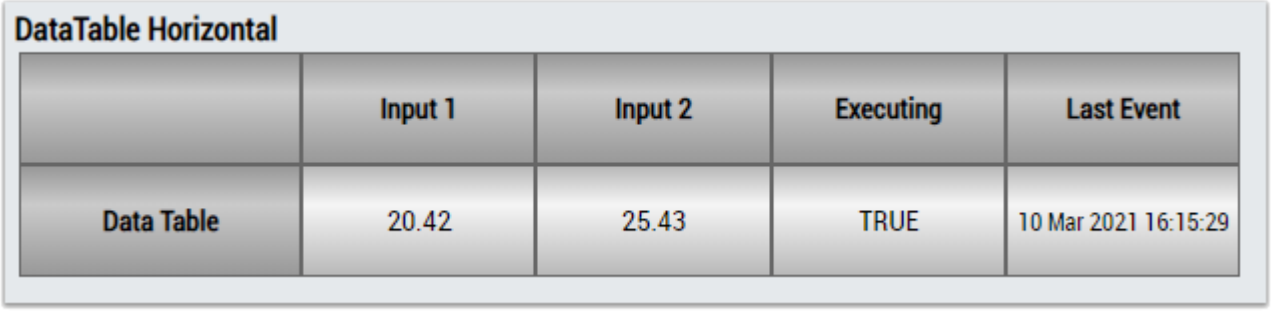
- **Shape Factor 00..n:** Shape-Factor der Eingangskanäle.

Standard HMI Controls

Für den Algorithmus *Shape Factor* stehen die folgenden HMI Controls für die Generierung eines Analytics Dashboards zur Verfügung:

1. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte.





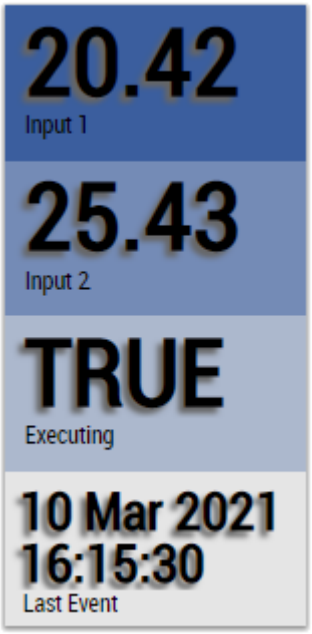
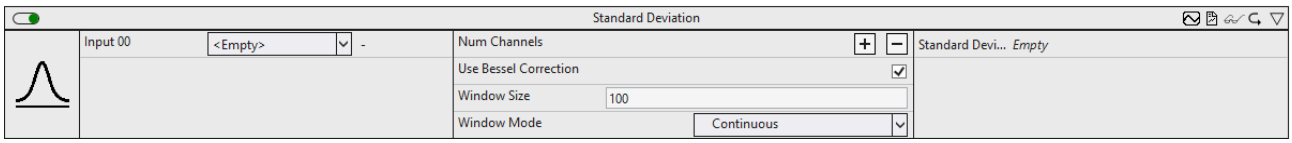


Abb. 4:

Alternativ können mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.10.9 Standard Deviation



Der Algorithmus *Standard Deviation* berechnet die empirische Standardabweichung für eine konfigurierbare Anzahl von Eingangskanälen. Die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogenen werden und die Art der Berechnung können konfiguriert werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Num Channels:** Konfiguration der Anzahl unabhängiger Eingangs- und Ausgangskanäle.
- **Use Bessel Correction:** Wenn die Checkbox aktiviert ist, wird die Bessel‘sche Korrektur angewendet. Um bei Stichproben ein erwartungstreues Ergebnis zu erhalten, ist dieser Parameter zu aktivieren.

Die empirische Standardabweichung, ohne Bessel’sche Korrektur

$$s' = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}(x[n]-\overline{x})^2}$$

Die empirische Standardabweichung, mit Bessel’scher Korrektur

$$s = \sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}(x[n]-\overline{x})^2}$$

- **Window Mode:** Verwendeter Fenstermodus. Beeinflusst die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogenen werden sowie den Zeitpunkt der Berechnungen.
  Continuous: Alle Eingangswerte seit dem Start des Algorithmus werden mit in die Berechnung einbezogen. Die Berechnung erfolgt zyklisch.
  Fix Window: Nur die letzten *N* Eingangswerte werden mit in die Berechnung einbezogen. Die

Berechnung erfolgt alle *N* Aufrufe.
Sliding Window: Nur die letzten *N* Eingangswerte werden mit in die Berechnung eingbezogen. Die Berechnung erfolgt zyklisch.

- **Window Size:** Abhängig von dem verwendeten Fenstermodus kann die Anzahl der Werte *N* konfiguriert werden, die mit in die Berechnung einbezogen werden. Beim Fenstermodus *Continuous* wird dieser Parameter ignoriert.

Ausgangswerte

- **Standard Deviation 00..n:** Empirische Standardabweichung der Eingangskanäle.

Standard HMI Controls

Für den Algorithmus Standard Deviation sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

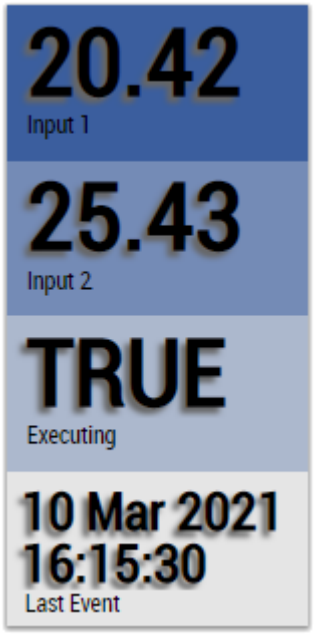
1. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Standard Deviation Results.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29

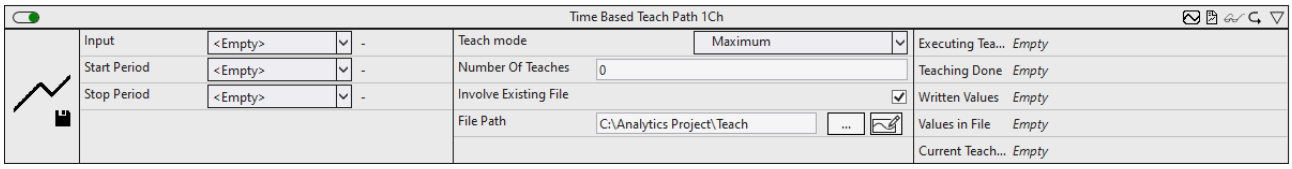


Alternativ können im Algorithmus Standard Deviation mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.11 Analytics - Training Base

Die Algorithmen der Kategorie *Analytics – Training Base* bieten Funktionen für das Einlernen periodischer Signale und schreiben diese Daten in eine Datei. Auf diese Weise können sie später mit einem anderen Eingangssignal verglichen werden, um Abweichungen vom optimalen Verhalten zu analysieren.

6.4.11.1 Time Based Teach Path 1Ch



Der *Time Based Teach Path 1Ch* schreibt die Eingangsdaten periodisch, entsprechend der konfigurierten Anzahl von Einlernvorgängen, in eine Datei. Dadurch werden die Werte nicht für jede Periode nacheinander geschrieben, sondern die Werte einer neuen Periode mit den vorhandenen Werten verglichen. Die Periode kann durch die Eingangswerte *Start Period* und *Stop Period* festgelegt werden (boolsche Signale sind erforderlich). Gemäß dem Teach -Modus wird jeder Wert überschrieben oder beibehalten, so dass das Ergebnis ein eingelerntes Eingangssignal ist, das später beispielsweise als Referenzsignal für den Algorithmus Time Based Envelope 1Ch [▶ 159] verwendet werden kann.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Es wird empfohlen, Time Based Teach Path 1Ch aufgrund des konkurrierenden Dateizugriffs nicht gleichzeitig mit Time Based Envelope 1Ch [▶ 159] zu verwenden. Stattdessen sollte zunächst das Referenzsignal erlernt werden und erst im Anschluss eine Auswertung mit dem Time Based Envelope 1Ch [▶ 159] vorgenommen werden.

Konfigurationsoptionen

- **Teach mode:** Modus für die Einlernung. Definiert, nach welchen Kriterien die Werte verglichen werden (*Minimum*, *Maximum* oder *Mean*). Im Fall von *Mean* wird ein gewichteter Mittelwert berechnet, um sicherzustellen, dass die Werte einer späteren Periode kein zunehmendes Gewicht im Hinblick auf das Gesamtergebnis haben.

- **Number of Teaches:** Anzahl der Zyklen, nach denen der Einlernprozess automatisch gestoppt werden soll. Wenn dies auf 0 gesetzt wird, schreitet die Einlernung kontinuierlich fort.
- **Involve Existing File:** Wenn die Checkbox aktiviert ist und bereits eine Datei mit Daten vorhanden ist, werden die Werte der vorhandenen Datei in den Einlernprozess einbezogen. Anderenfalls wird die vorhandene Datei ignoriert und überschrieben.
- **File Path:** Pfad zur Datendatei.
- **Negate Start Period:** Wenn die Checkbox aktiviert ist, wird das boolsche Eingangssignal von *Start Period* negiert.
- **Negate Stop Period:** Wenn die Checkbox aktiviert ist, wird das boolsche Eingangssignal von *Stop Period* negiert.

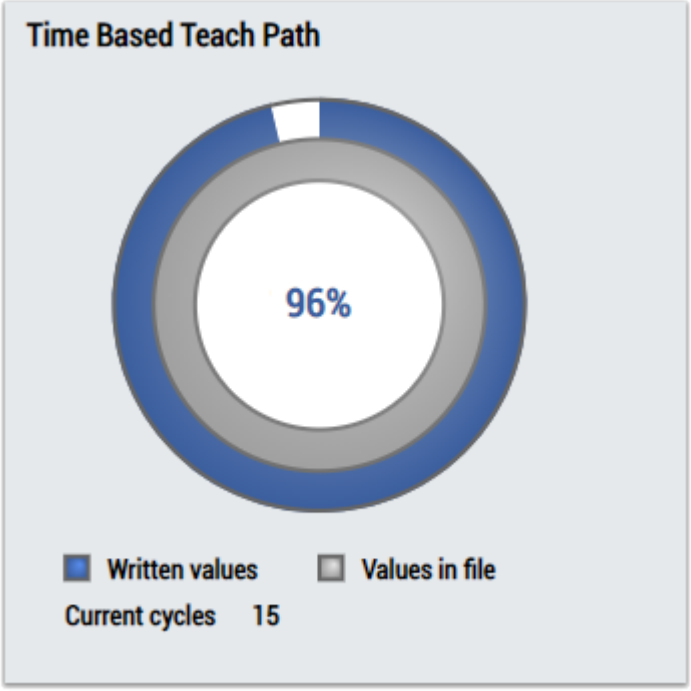
Ausgangswerte

- **Executing Teach:** Zeigt an, ob die Einlernung aktiv ist (Zeitbereich zwischen Start- und Stoppflag).
- **Written Values:** Zeigt die Gesamtanzahl der geschriebenen Werte während des Einlernprozesses an. Nicht zu verwechseln mit der Anzahl der Values in File, die jeden Einlernzyklus überschrieben werden.
- **Values in File:** Zeigt die Anzahl der Werte an, die gegenwärtig in die Datei geschrieben werden (nach einem Zyklus ist der Wert konstant).
- **Current Teach Cycles:** Zeigt die Anzahl der Einlernzyklen in der Datei an.

Standard HMI Controls

Für den Algorithmus Time Based Teach Path 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das TimeBasedTeachPath Control visualisiert die Ausgangswerte Written Values, Values in File und Current Teach Cycles.



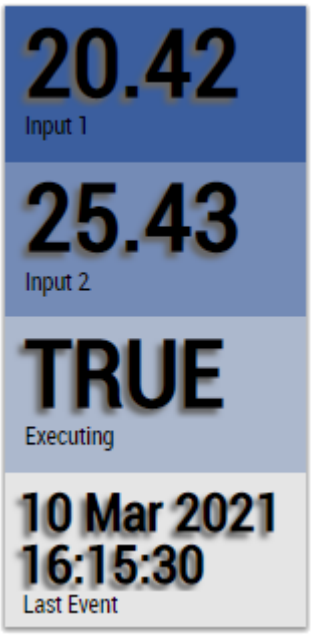
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Written Values, Values in File, Current Teach Cycles.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



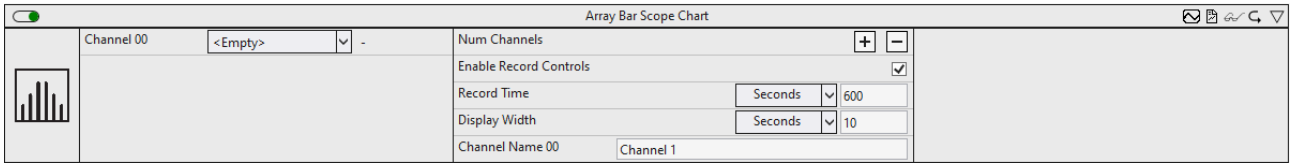
Alternativ können im Algorithmus Time Based Teach Path 1Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.12 Analytics - Visualization Only

Die Algorithmen der Kategorie *Visualization Only* stellen Hilfsbausteine dar, die Funktionen für die spätere Visualisierung im Kontext des HMI-Dashboards bieten.

6.4.12.1 Array Bar Scope Chart

Die Generierung des HMI Scope Controls für das HMI One Click Dashboard ist nur mit einer HMI Version > 1.12.752.0 möglich.



Der *Array Bar Scope Chart* Baustein dient zur Darstellung der Inputs in dem HMI Dashboard. Dargestellt werden die Inputs in einem Array Bar Chart im HMI Scope Control.

Da die Funktion dieses Bausteins für die Analyse selbst irrelevant ist, zählt er zur Kategorie Visualization Only.

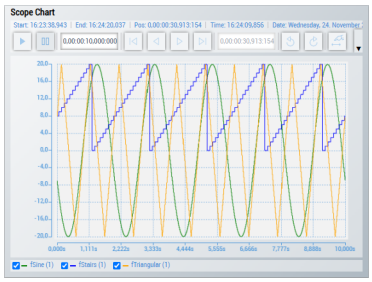
Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Inputs, die im HMI Scope Control angezeigt werden sollen.
- **Channel Name 00/XX:** Die Anzeigenamen der Inputs. Diese tauchen als Legende im HMI Scope Control auf.
- **Enable Record Controls:** Wenn die Checkbox aktiviert ist, kann eine Aufnahme aus der HMI gestartet und gestoppt werden.
- **Record Time:** Die Aufnahmezeit eines Records. Nach Erreichen der Zeit läuft das Scope Chart in einem Ringpuffer.
- **Display Width:** Die ausgewählte Displaybreite der Aufnahme. Diese Eigenschaft kann ebenfalls im HMI Scope Control modifiziert werden.

Standard HMI Controls

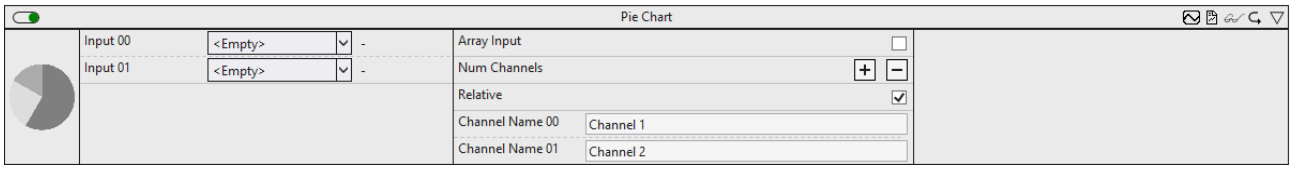
Für den Algorithmus *Array Bar Scope Chart* ist das folgende HMI Control für die Generierung eines Analytics Dashboards verfügbar.

Das HMI Scope Control zur Anzeige eines Scope Charts.



Dem Baustein können keine weiteren HMI Controls hinzugefügt werden, da dieser ausschließlich zur Darstellung des HMI Scope Controls gedacht ist.

6.4.12.2 Pie Chart



Der Baustein *Pie Chart* kann verwendet werden, um ein Tortendiagramm (Pie Chart) zur späteren Verwendung im HMI Dashboard zu erstellen. Die Anzahl der Tortenstücke kann über den Parameter*Num Channels* parametriert werden.

Da die Funktion dieses Bausteins für die Analyse selbst irrelevant ist, zählt er zur Kategorie Visualization Only.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

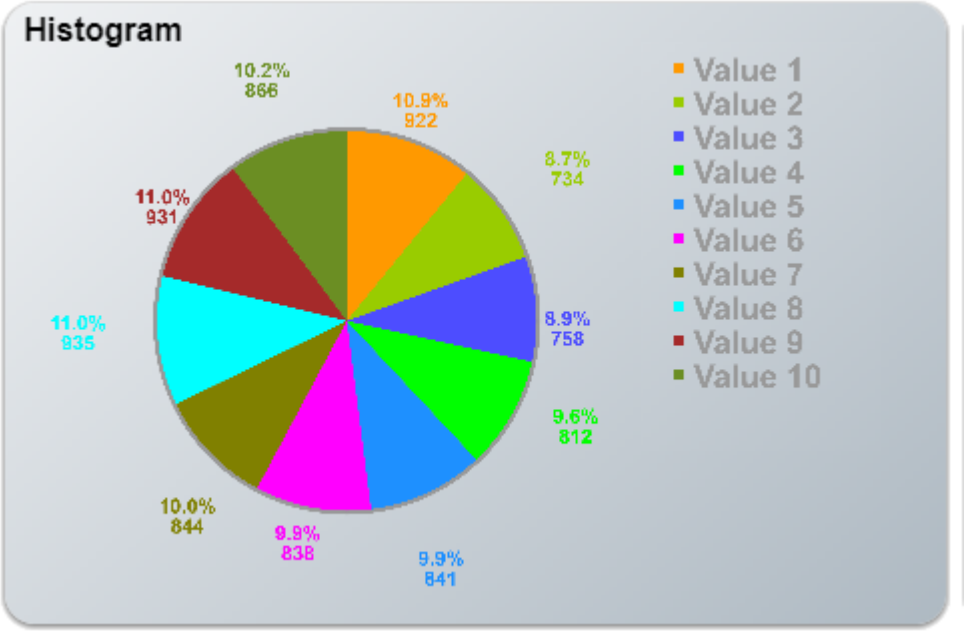
Konfigurationsoptionen

- **Array Input:** Auswahl, ob ein einzelnes Array als Eingang oder mehrere unabhängige Eingänge verwendet werden sollen. Wird automatisch gesetzt, sobald ein Array auf den ersten Eingang gezogen wird. Solange das Array noch verknüpft ist, kann der Parameter nicht zurückgesetzt werden.
- **Num Channels**: Gibt die Anzahl der konfigurierbaren Kanäle an. Diese entspricht der Anzahl von Tortenstücken im Pie Chart.
- **Channel Name 00:** Gibt den Namen des ersten Kanals an. Der hier vergebene Name wird später für diesen Kanal im HMI Dashboard verwendet.
- **Channel Name 01:** Gibt den Namen des zweiten Kanals an. Der hier vergebene Name wird später für diesen Kanal im HMI Dashboard verwendet.
- **Channel Name xx:** Analog für alle weiteren Kanäle.

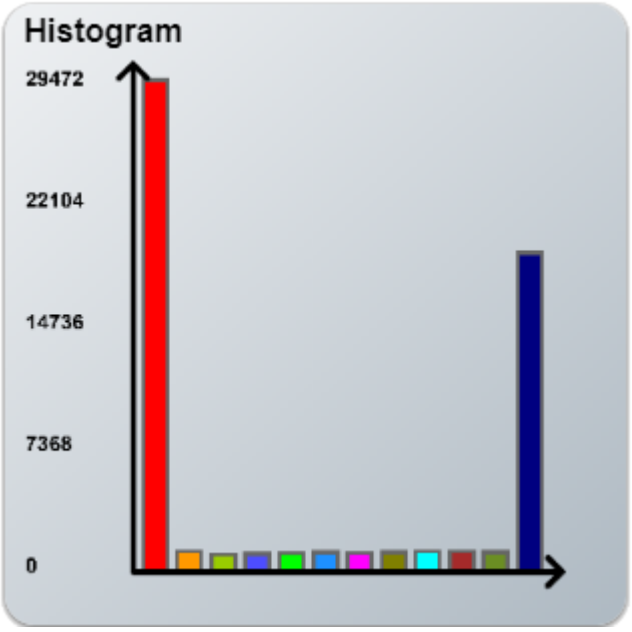
Standard HMI Controls

Für den Algorithmus *Pie Chart* sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar.

Pie Chart zur Anzeige der Daten in einem Kuchendiagramm.

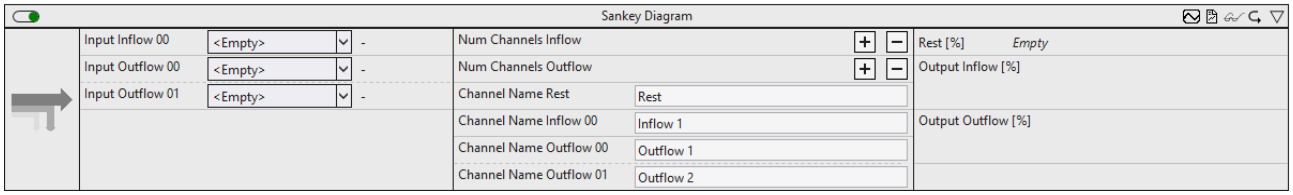


Histogram zur Anzeige der Daten in einem Histogramm.



Alternativ können im Algorithmus *Pie Chart* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.12.3 Sankey Diagram



Der Baustein *Sankey Diagram* kann verwendet werden, um ein Sankey Diagramm zur späteren Verwendung im HMI Dashboard zu erstellen. Es wird zwischen Zu- und Abflüssen unterschieden. Die Anzahl von Zu- und Abflüssen kann jeweils individuell über die Parameter *Num Channels Inflow* und *Num Channels Outflow* parametriert werden.

Da die Funktion dieses Bausteins für die Analyse selbst irrelevant ist, zählt er zur Kategorie Visualization Only.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

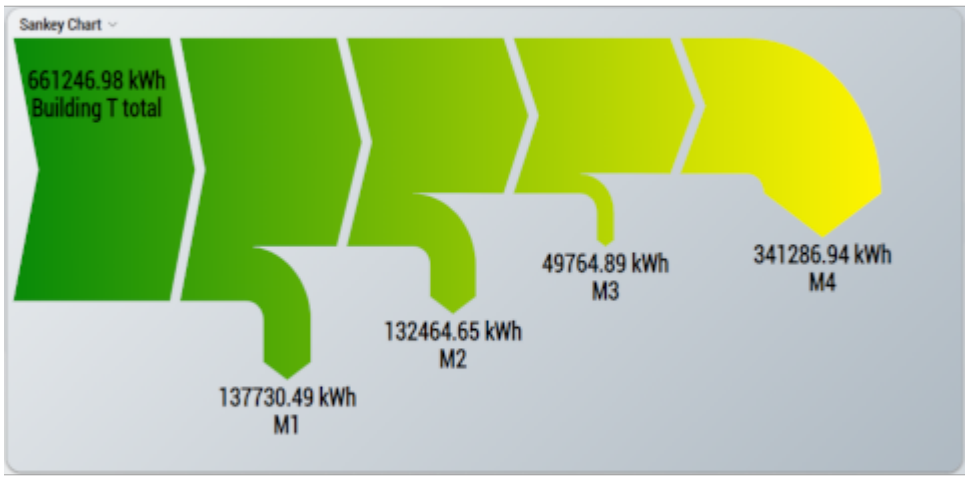
Konfigurationsoptionen

- **Num Channels Inflow**: Gibt die Anzahl der Kanäle von Zuflüssen im Sankey Diagramm an.
- **Num Channels Outflow**: Gibt die Anzahl der Kanäle von Abflüssen im Sankey Diagramm an.
- **Channel Name Rest:** Gibt den Namen des Kanals an, mit dem ein möglicher Restwert bezeichnet werden soll. Der hier vergebene Name wird später für diesen Abfluss im HMI Dashboard verwendet.
- **Channel Name Inflow 00:** Gibt den Namen des ersten Zufluss-Kanals an. Der hier vergebene Name wird später für diesen Zufluss im HMI Dashboard verwendet.
- **Channel Name Inflow xx:** Analog für alle weiteren Zufluss-Kanäle.
- **Channel Name Outflow 00:** Gibt den Namen des ersten Abfluss-Kanals an. Der hier vergebene Name wird später für diesen Abfluss im HMI Dashboard verwendet.
- **Channel Name Outflow 01:** Gibt den Namen des zweiten Abfluss-Kanals an. Der hier vergebene Name wird später für diesen Abfluss im HMI Dashboard verwendet.
- **Channel Name Outflow xx:** Analog für alle weiteren Kanäle.

Standard HMI Controls

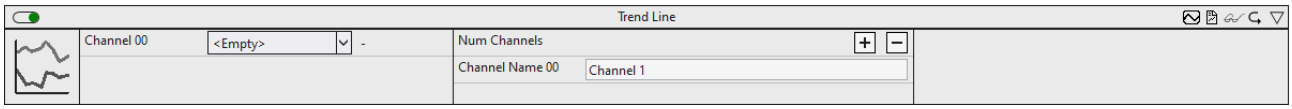
Für den Algorithmus *Sankey Diagram* ist das folgende HMI Control für die Generierung eines Analytics Dashboards verfügbar.

Sankey Chart zur Anzeige der Daten.



Alternativ können im Algorithmus *Sankey Diagram* mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.12.4 Trend Line



Der *Trend Line* Baustein sorgt für die Weiterleitung von Trends der Eingangssignale zur späteren Verwendung im HMI-Dashboard. Es können beliebige Kanäle für die Weiterleitung ausgewählt werden, wobei die Anzahl der Kanäle über NumChannels gesteuert werden kann. Für jeden Kanal wird dann jeweils ein Wert pro Minute gespeichert. Damit ist der Baustein nicht zur Signalweiterleitung geeignet, sondern eher für die Darstellung von Trends für Signale, die sich langsam verändern.

Da die Funktion dieses Bausteins für die Analyse selbst irrelevant ist, zählt er zur Kategorie Visualization Only.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

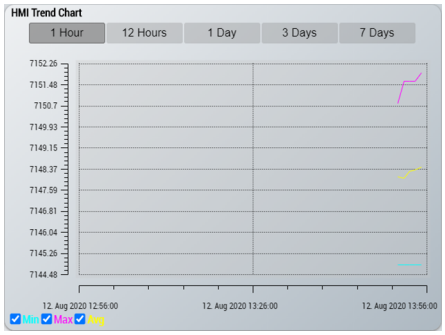
- **Num Channels**: Gibt die Anzahl der konfigurierbaren Kanäle für die Weiterleitung der Eingangssignale an.

- **Channel 00:** Gibt den Namen des *ersten* Kanals an, der weitergeleitet werden soll. Der hier vergebene Name wird später für diesen Kanal im HMI-Dashboard verwendet.
- **Channel 01:** Gibt den Namen des *zweiten* Kanals an, der weitergeleitet werden soll. Der hier vergebene Name wird später für diesen Kanal im HMI-Dashboard verwendet.
- **Channel 02:** Gibt den Namen des *dritten* Kanals an, der weitergeleitet werden soll. Der hier vergebene Name wird später für diesen Kanal im HMI-Dashboard verwendet.
- **Channel xx:** Analog für alle weiteren Kanäle.

Standard HMI Controls

Für den Algorithmus *Trend Line* ist das folgende HMI Control für die Generierung eines Analytics Dashboards verfügbar.

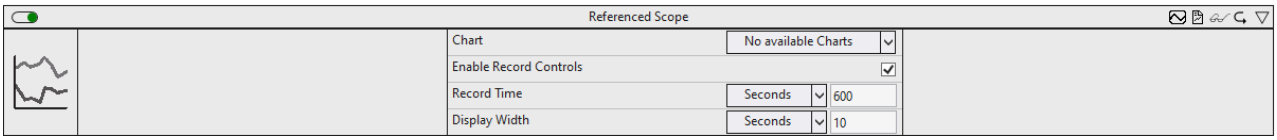
Trend Line zur Anzeige der Daten über die Zeit.



Dem Baustein können keine weiteren HMI Controls hinzugefügt werden, da dieser ausschließlich zur Darstellung des Trend Line Controls gedacht ist.

6.4.12.5 Referenced Scope

Die Generierung des HMI Scope Controls für das HMI One Click Dashboard ist nur mit einer HMI Version > 1.12.752.0 möglich.



Der *Referenced Scope* Baustein dient zur Darstellung eines Scope Charts aus dem Referenced Scope im HMI Dashboard

Da die Funktion dieses Bausteins für die Analyse selbst irrelevant ist, zählt er zur Kategorie Visualization Only.

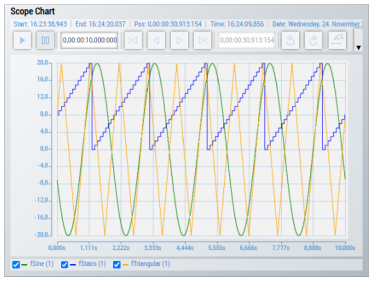
Konfigurationsoptionen

- **Chart:** Das Chart aus dem Referenced Scope, welches in der HMI angezeigt werden soll. Bisher werden folgende Charts unterstützt: YT Chart, XY Chart, Array Bar Chart und Single Bar Chart.
- **Enable Record Controls:** Wenn die Checkbox aktiviert ist, kann eine Aufnahme aus der HMI gestartet und gestoppt werden.
- **Record Time:** Die Aufnahmezeit eines Records. Nach Erreichen der Zeit läuft das Scope Chart in einem Ringpuffer.
- **Display Width:** Die ausgewählte Displaybreite der Aufnahme. Diese Eigenschaft kann ebenfalls im HMI Scope Control modifiziert werden.

Standard HMI Controls

Für den Algorithmus *Referenced Scope* ist das folgende HMI Control für die Generierung eines Analytics Dashboards verfügbar.

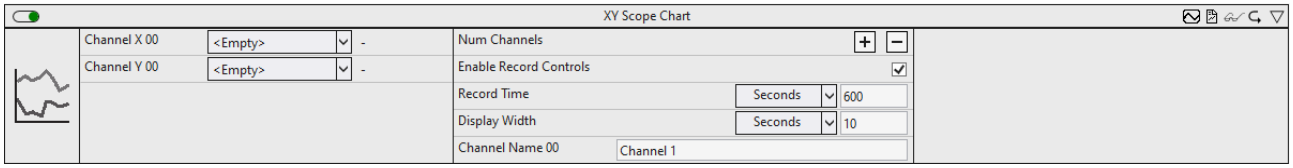
Das HMI Scope Control zur Anzeige eines Scope Charts.



Dem Baustein können keine weiteren HMI Controls hinzugefügt werden, da dieser ausschließlich zur Darstellung des HMI Scope Controls gedacht ist.

6.4.12.6 XY Scope Chart

Die Generierung des HMI Scope Controls für das HMI One Click Dashboard ist nur mit einer HMI Version > 1.12.752.0 möglich.



Der *XY Scope Chart* Baustein dient zur Darstellung der Inputs in dem HMI Dashboard. Dargestellt werden die Inputs in einem XY Chart im HMI Scope Control.

Da die Funktion dieses Bausteins für die Analyse selbst irrelevant ist, zählt er zur Kategorie Visualization Only.

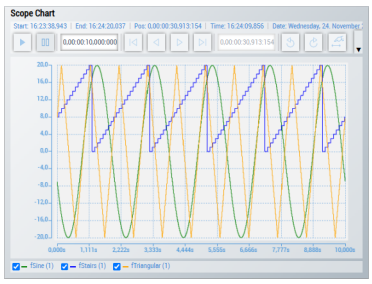
Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Kanäle bzw. Input-Paare, die im HMI Scope Control angezeigt werden sollen.
- **Channel Name 00/XX:** Die Anzeigenamen der Kanäle. Diese tauchen als Legende im HMI Scope Control auf.
- **Enable Record Controls:** Wenn die Checkbox aktiviert ist, kann eine Aufnahme aus der HMI gestartet und gestoppt werden.
- **Record Time:** Die Aufnahmezeit eines Records. Nach Erreichen der Zeit läuft das Scope Chart in einem Ringpuffer.
- **Display Width:** Die ausgewählte Displaybreite der Aufnahme. Diese Eigenschaft kann ebenfalls im HMI Scope Control modifiziert werden.

Standard HMI Controls

Für den Algorithmus *XY Scope Chart* ist das folgende HMI Control für die Generierung eines Analytics Dashboards verfügbar.

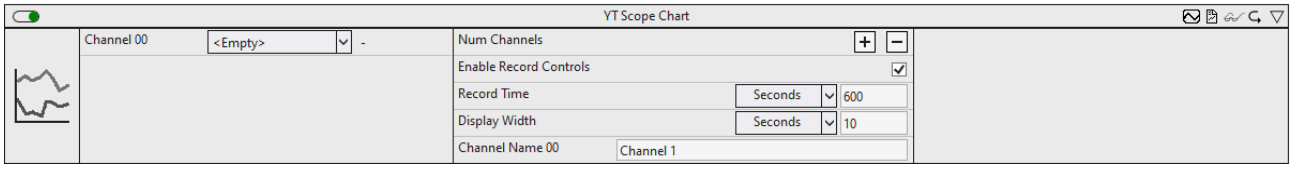
Das HMI Scope Control zur Anzeige eines Scope Charts.



Dem Baustein können keine weiteren HMI Controls hinzugefügt werden, da dieser ausschließlich zur Darstellung des HMI Scope Controls gedacht ist.

6.4.12.7 YT Scope Chart

Die Generierung des HMI Scope Controls für das HMI One Click Dashboard ist nur mit einer HMI Version > 1.12.752.0 möglich.



Der *YT Scope Chart* Baustein dient zur Darstellung der Inputs in dem HMI Dashboard. Dargestellt werden die Inputs in einem YT Chart im HMI Scope Control.

Da die Funktion dieses Bausteins für die Analyse selbst irrelevant ist, zählt er zur Kategorie Visualization Only.

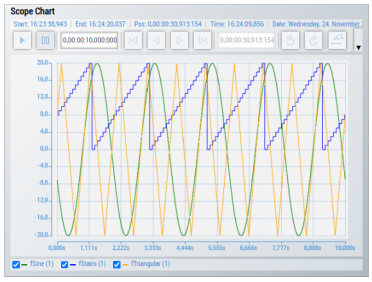
Konfigurationsoptionen

- **Num Channels:** Die Anzahl der Inputs, die im HMI Scope Control angezeigt werden sollen.
- **Channel Name 00/XX:** Die Anzeigenamen der Inputs. Diese tauchen als Legende im HMI Scope Control auf.
- **Enable Record Controls:** Wenn die Checkbox aktiviert ist, kann eine Aufnahme aus der HMI gestartet und gestoppt werden.
- **Record Time:** Die Aufnahmezeit eines Records. Nach Erreichen der Zeit läuft das Scope Chart in einem Ringpuffer.
- **Display Width:** Die ausgewählte Displaybreite der Aufnahme. Diese Eigenschaft kann ebenfalls im HMI Scope Control modifiziert werden.

Standard HMI Controls

Für den Algorithmus *YT Scope Chart* ist das folgende HMI Control für die Generierung eines Analytics Dashboards verfügbar.

Das HMI Scope Control zur Anzeige eines Scope Charts.

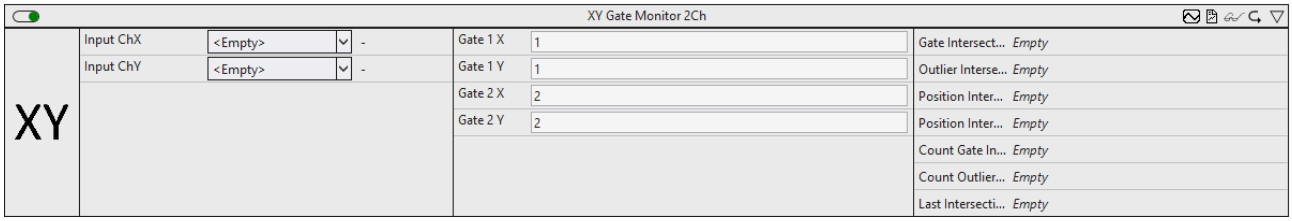


Dem Baustein können keine weiteren HMI Controls hinzugefügt werden, da dieser ausschließlich zur Darstellung des HMI Scope Controls gedacht ist.

6.4.13 Analytics - XY Path Analysis

Die Algorithmen der Kategorie *XY Path Analysis* bieten Funktionen für die Positionsauswertung von XY-Kanälen. Es kann beispielsweise analysiert werden, ob sich die Position, die durch die Eingangskanäle bestimmt wird, innerhalb bestimmter Schranken oder Formen befindet und wie oft Grenzüberschreitungen stattfinden.

6.4.13.1 XY Gate Monitor 2Ch

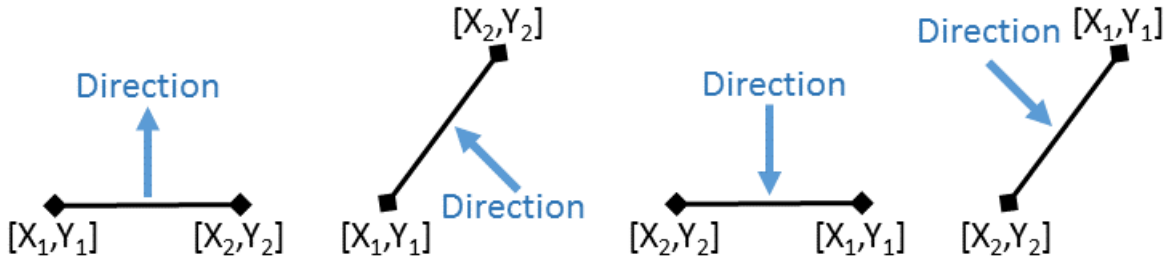


XY Gate Monitor 2Ch zählt die Anzahl der Schnittpunkte eines XY-Eingangs mit einem bestimmten Gate oder seiner Projektion (gerade Linie zwischen den Gate-Punkten) abhängig vom konfigurierten Gate Mode. Die Analyseperiode kann mit den Eingängen *Start* und *Stop* gestartet werden. Der Algorithmus ist richtungsempfindlich, d. h. es werden nur Schnittpunkte in der richtigen Richtung gezählt. Die Richtungsauslegung hängt von der Reihenfolge der Gate-Punkte (X1/Y1) und (X2/Y2) ab. Die möglichen Richtungen der Schnittpunkte sind nachstehend veranschaulicht.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Richtungen der Schnittpunkte:

Der blaue Pfeil stellt die Signalrichtung dar und die schwarzen Linien veranschaulichen das Gate mit seinen Gate-Punkten (X1/Y1) und (X2/Y2). Die Richtung der Schnittpunkte wird gezählt, wenn sich das Signal gegen den Uhrzeigersinn um den ersten Gate-Punkt (X1/Y1) dreht.



Konfigurationsoptionen

- **Gate Mode:** Modus des Gate Monitors:

  Intersect Gate: Bestimmt, ob das XY-Signal das Gate in der konfigurierten Richtung kreuzt. Liegt während der Analyseperiode ein Schnittpunkt vor, wird dies als *OK* eingestuft, ansonsten *NOK*.

  Not Intersect Gate: Überwacht, ob sich das XY-Signal während der Analyseperiode nicht mit dem Gate in der konfigurierten Richtung kreuzt. Dann wird dies als *OK* eingestuft, ansonsten *NOK*.

  Intersect Projection: Bestimmt, ob das XY-Signal die Projektion des Gates in der konfigurierten Richtung kreuzt. Liegt während der Analyseperiode ein Schnittpunkt vor, wird dies als *OK* eingestuft, ansonsten *NOK*.

  Not Intersect Projection: Überwacht, ob sich das XY-Signal während der Analyseperiode nicht mit der Projektion des Gates in der konfigurierten Richtung kreuzt. Dann wird dies als *OK* eingestuft, ansonsten *NOK*.

  Intersect Gate Or Projection: Bestimmt, ob das XY-Signal das Gate oder seine Projektion in der konfigurierten Richtung kreuzt. Liegt während der Analyseperiode ein Schnittpunkt vor, wird dies als *OK* eingestuft, ansonsten *NOK*.
- **Gate 1 X:** X-Position des ersten Gate-Punkts.
- **Gate 1 Y:** Y-Position des ersten Gate-Punkts.

- **Gate 2 X:** X-Position des zweiten Gate-Punkts.
- **Gate 2 Y:** Y-Position des zweiten Gate-Punkts.

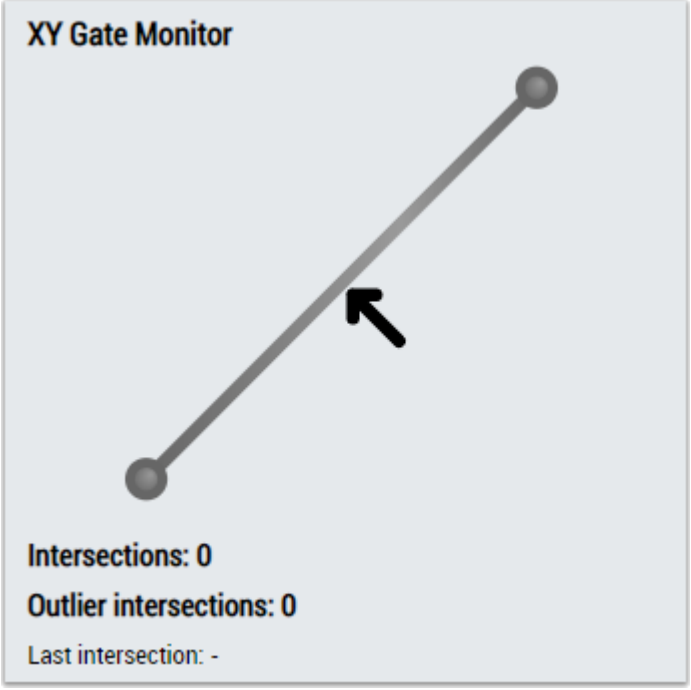
Ausgangswerte

- **Gate Intersection:** Zeigt an, ob gegenwärtig ein Gate-Schnittpunkt vorliegt.
- **Outlier Intersection:** Zeigt an, ob gegenwärtig ein Ausreißer-Schnittpunkt vorliegt (Schnittpunkt der Gate-Projektion).
- **Position Intersection X:** X-Position des letzten Schnittpunkts.
- **Position Intersection Y:** Y-Position des letzten Schnittpunkts.
- **Count Gate Intersections:** Zeigt die Gesamtanzahl der Gate-Schnittpunkte an.
- **Count Outlier Intersections:** Zeigt die Gesamtanzahl der Ausreißer-Schnittpunkte an.
- **Last Intersection:** Gibt den Zeitpunkt des letzten Schnittpunkt-Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.
- **New Result: Gibt an, ob ein neues Ergebnis berechnet wurde oder nicht.**
- **Executing:** Gibt an, ob der Algorithmus aktiv oder inaktiv ist.
- **Classification:** Zeigt das Klassifizierungsergebnis an. *OK* oder *NOK*. Die Klassifizierung hängt vom Gate Mode ab, wie oben zu sehen ist.

Standard HMI Controls

Für den Algorithmus XY Gate Monitor 2Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das XYGateMonitor Control visualisiert die Ausgangswerte Gate Intersection, Outlier Intersection, Count Gate Intersections, Count Outlier Intersections und Last Intersection sowie die Richtung der Schnittpunkte.



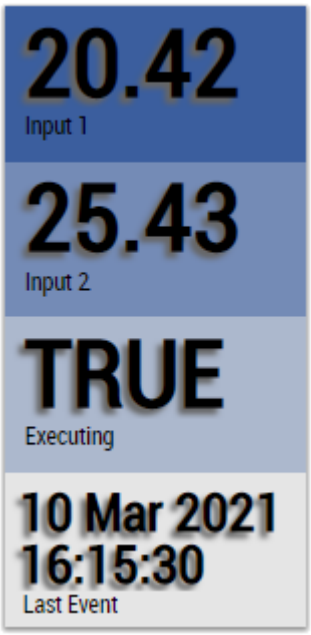
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Executing, Gate Intersection, Outlier Intersection, Position Intersection X, Position Intersection Y, Count Gate Intersections, Count Outlier Intersections, Last Intersection, Classification.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

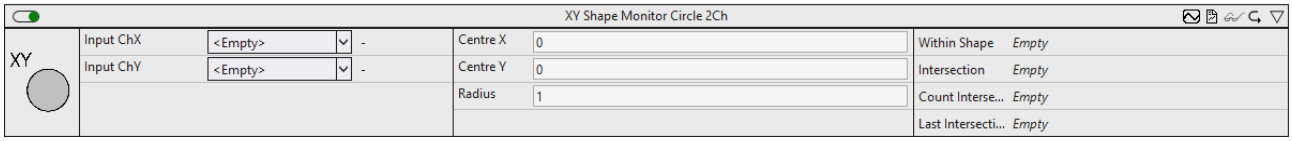
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus XY Gate Monitor 2Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.13.2 XY Shape Monitor Circle 2Ch



Der *XY Shape Monitor Circle 2Ch* zählt die Anzahl der Schnittpunkte eines XY-Eingangs mit einer bestimmten Kreisform.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Centre X:** X-Position des Kreismittelpunkts.
- **Centre Y:** Y-Position des Kreismittelpunkts.
- **Radius:** Radius des Kreises.

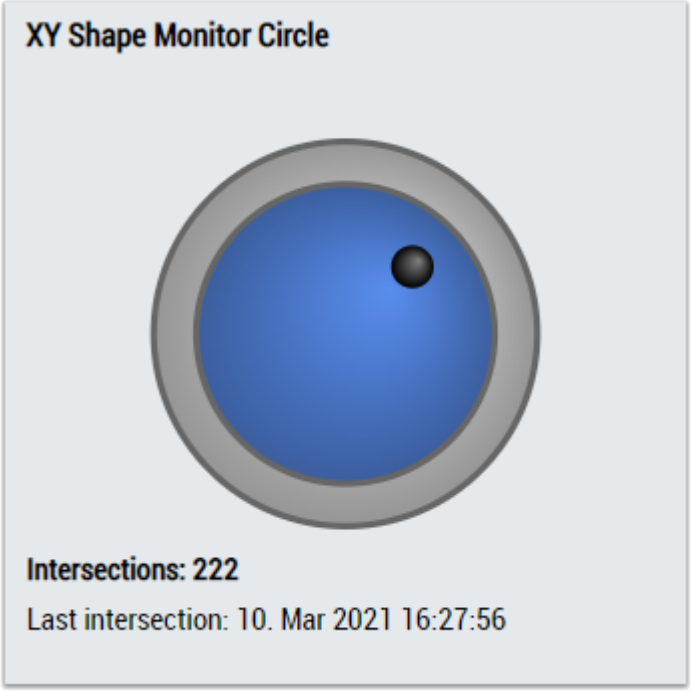
Ausgangswerte

- **Within Shape:** Zeigt an, ob sich das Eingangssignal gegenwärtig innerhalb der bestimmten Form befindet.
- **Intersection:** Zeigt an, ob das Eingangssignal gegenwärtig die bestimmte Form kreuzt.
- **Count Intersections:** Zählt die Gesamtanzahl der Schnittpunkte von Eingangssignal und Form.
- **Last Intersection:** Gibt den Zeitpunkt des letzten Schnittpunkt-Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

Standard HMI Controls

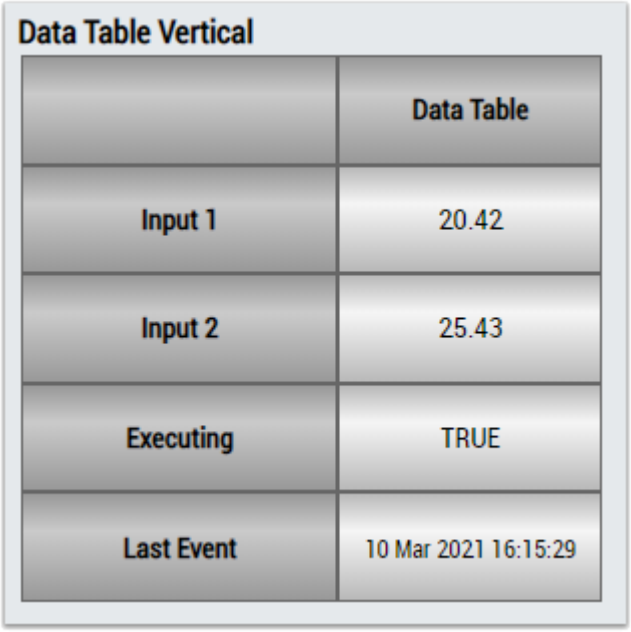
Für den Algorithmus XY Shape Monitor Circle 2Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das XYShapeMonitor Control visualisiert die Ausgangswerte Within Shape, Intersection, Count Intersections und Last Intersection.



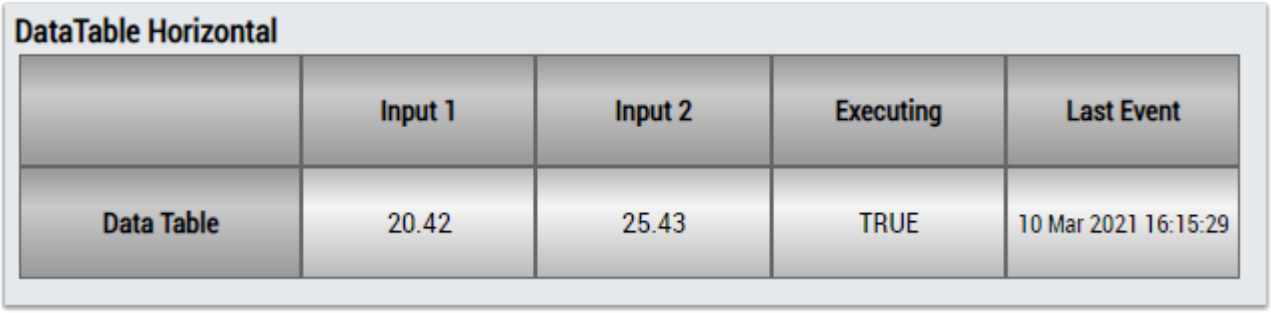
2. Das SingleValue Control visualisiert die Ausgangswerte Intersection und Last Intersection.

3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Within Shape, Intersection, Count Intersections, Last Intersection.



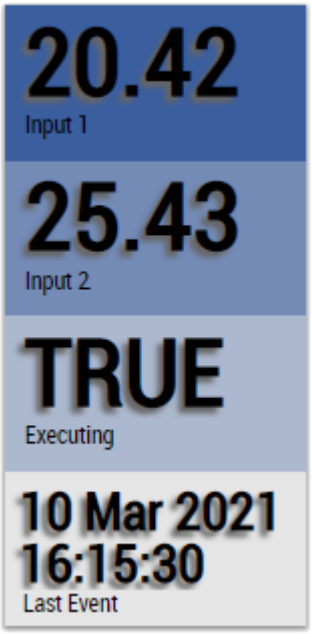
Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29



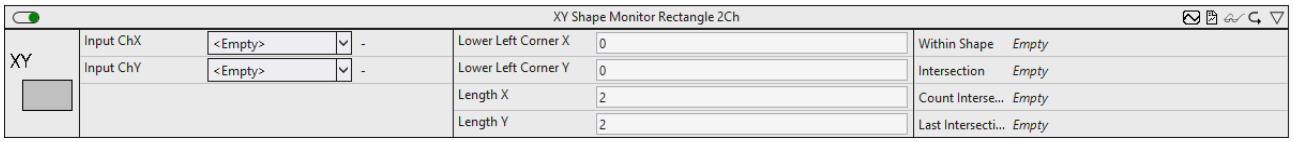
DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



Alternativ können im Algorithmus XY Shape Monitor Circle 2Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.13.3 XY Shape Monitor Rectangle 2Ch



Der *XY Shape Monitor Rectangle 2Ch* zählt die Anzahl der Schnittpunkte eines XY-Eingangs mit einer bestimmten Rechteckform.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Lower Left Corner X:** X-Position der unteren linken Ecke des Rechtecks.
- **Lower Left Corner Y:** Y-Position der unteren linken Ecke des Rechtecks.
- **Length X:** Länge des Rechtecks in positiver X-Richtung.
- **Length Y:** Länge des Rechtecks in positiver Y-Richtung.

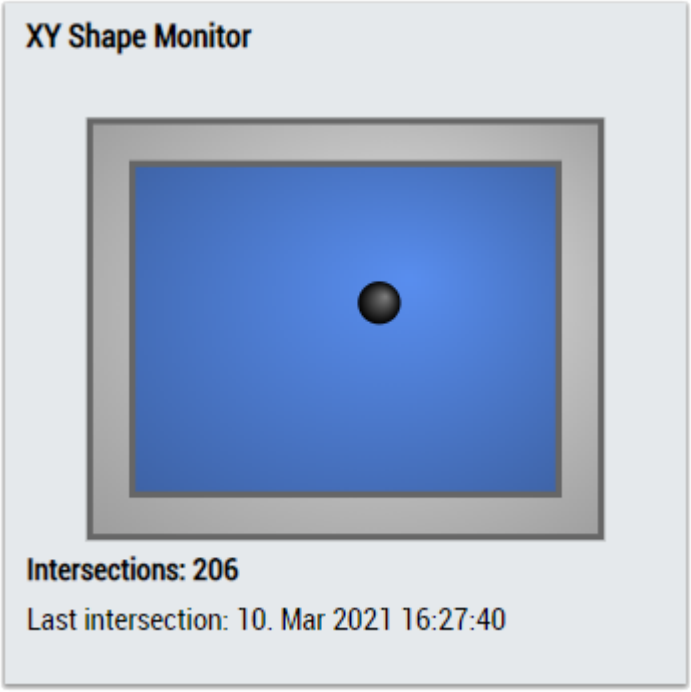
Ausgangswerte

- **Within Shape:** Zeigt an, ob sich das Eingangssignal gegenwärtig innerhalb der bestimmten Form befindet.
- **Intersection:** Zeigt an, ob das Eingangssignal gegenwärtig die bestimmte Form kreuzt.
- **Count Intersections:** Zählt die Gesamtanzahl der Schnittpunkte von Eingangssignal und Form.
- **Last Intersection:** Gibt den Zeitpunkt des letzten Schnittpunkt-Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

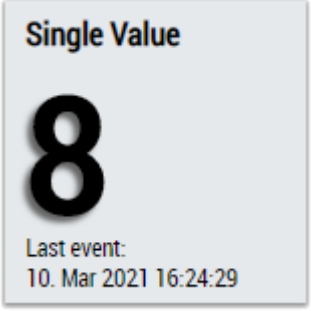
Standard HMI Controls

Für den Algorithmus XY Shape Monitor Rectangle 2Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

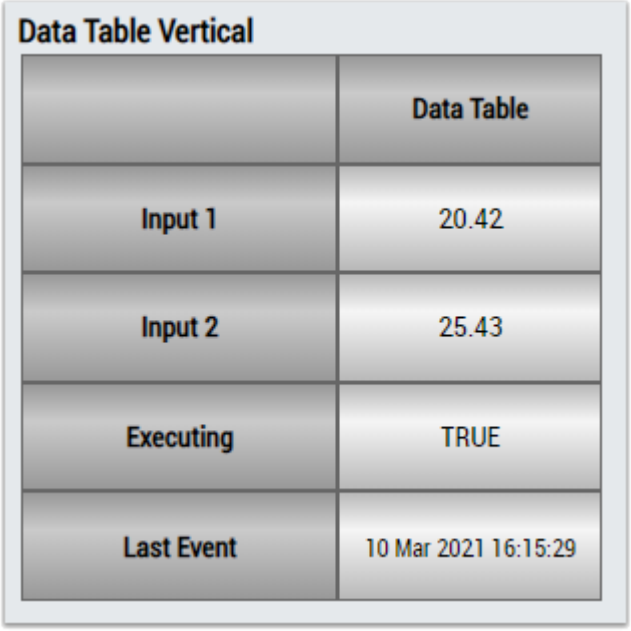
1. Das XYShapeMonitor Control visualisiert die Ausgangswerte Within Shape, Intersection, Count Intersections und Last Intersection.

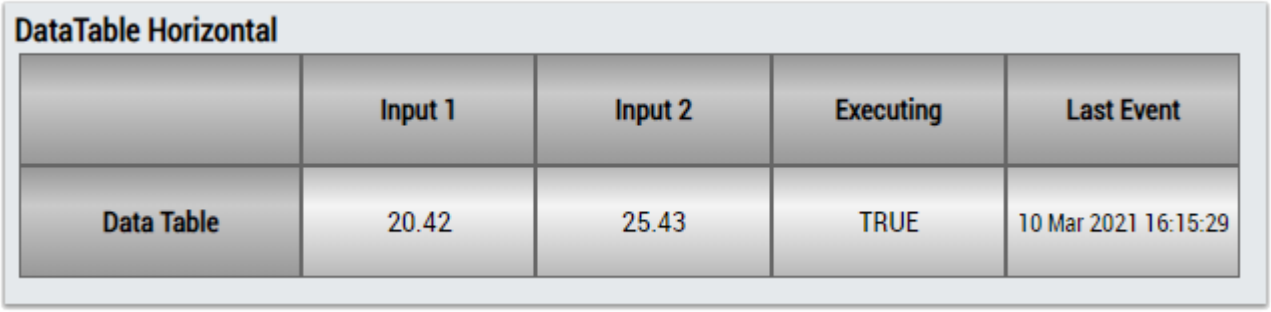


2. Das SingleValue Control visualisiert die Ausgangswerte Intersection und Last Intersection.



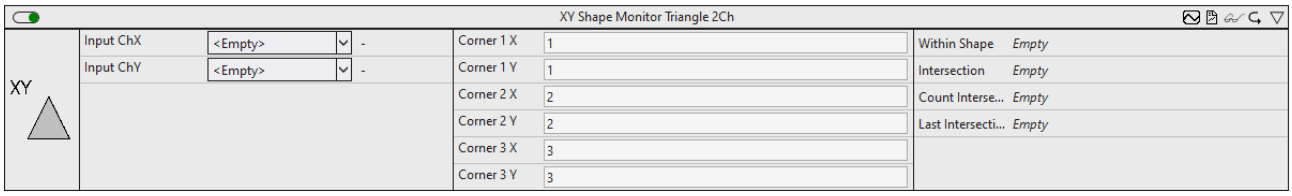
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Within Shape, Intersection, Count Intersections, Last Intersection.





Alternativ können im Algorithmus XY Shape Monitor Rectangle 2Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.13.4 XY Shape Monitor Triangle 2Ch



Der *XY Shape Monitor Triangle 2Ch* zählt die Anzahl der Schnittpunkte eines XY-Eingangs mit einer bestimmten Dreieckform.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Corner 1 X:** X-Position der ersten Ecke des Dreiecks.
- **Corner 1 Y:** Y-Position der ersten Ecke des Dreiecks.
- **Corner 2 X:** X-Position der zweiten Ecke des Dreiecks.
- **Corner 2 Y:** Y-Position der zweiten Ecke des Dreiecks.
- **Corner 3 X:** X-Position der dritten Ecke des Dreiecks.
- **Corner 3 Y:** Y-Position der dritten Ecke des Dreiecks.

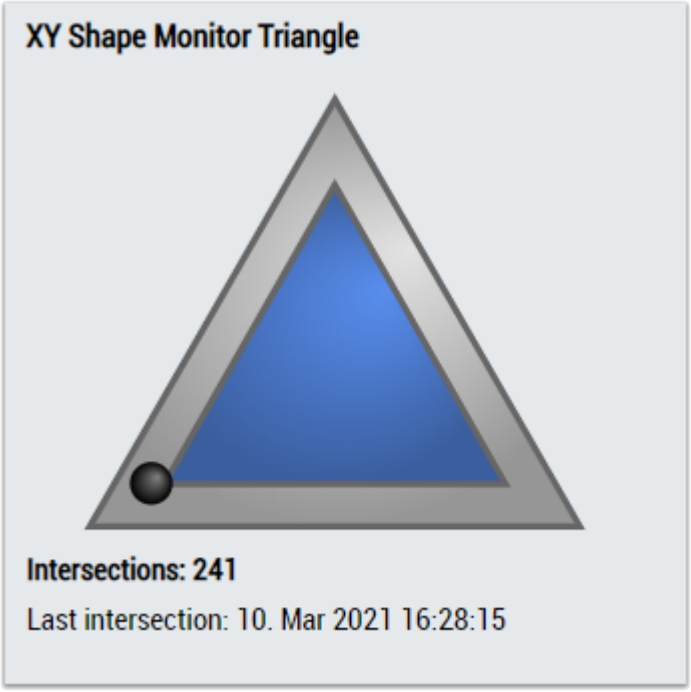
Ausgangswerte

- **Within Shape:** Zeigt an, ob sich das Eingangssignal gegenwärtig innerhalb der bestimmten Form befindet.
- **Intersection:** Zeigt an, ob das Eingangssignal gegenwärtig die bestimmte Form kreuzt.
- **Count Intersections:** Zählt die Gesamtanzahl der Schnittpunkte von Eingangssignal und Form.
- **Last Intersection:** Gibt den Zeitpunkt des letzten Schnittpunkt-Ereignisses an → das Ereignis kann per Drag-and-Drop in den Scope-Chart gezogen werden, um es als Trigger-Event anzuzeigen.

Standard HMI Controls

Für den Algorithmus XY Shape Monitor Triangle 2Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das XYShapeMonitor Control visualisiert die Ausgangswerte Within Shape, Intersection, Count Intersections und Last Intersection.



2. Das SingleValue Control visualisiert die Ausgangswerte Intersection und Last Intersection.

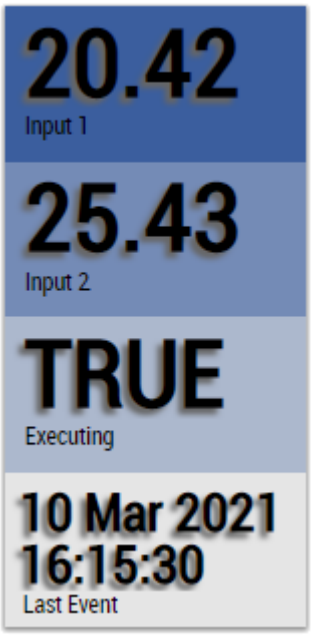
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Within Shape, Intersection, Count Intersections, Last Intersection.

Data Table Vertical

	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

DataTable Horizontal

	Input 1	Input 2	Executing	Last Event
Data Table	20.42	25.43	TRUE	10 Mar 2021 16:15:29



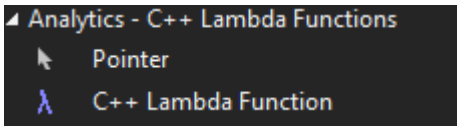
Alternativ können im Algorithmus XY Shape Monitor Triangle 2Ch mit dem Mapping Wizard [▶ 446] kundenspezifische HMI Controls gemappt werden.

6.4.14 Analytics - Erweiterung der Algorithmik

Um die Algorithmen der TwinCAT Analytics Library, die im TwinCAT Analytics Workbench-Konfigurator beziehungsweise im TwinCAT Analytics Service Tool zur Verfügung stehen, zu erweitern, sind zwei Erweiterungsmöglichkeiten verfügbar. Zum einen können mit den Analytics Lambda Funktionen anwenderspezifische Algorithmen entwickelt und in den Analytics Workflow integriert werden. Zum anderen können Algorithmen aus anderen Bibliotheken die bestehenden Algorithmen erweitern.

6.4.14.1 C++ Lambda Funktionen

Die C++ Lambda Funktionen bieten die Möglichkeit anwenderspezifische Algorithmen zu entwickeln und in den TwinCAT Analytics Workflow zu integrieren. Um durch die Entwicklung des Algorithmus zu führen, ist in der Visual Studio® Toolbox ein Lambda Template verfügbar.



Über das Lambda Template kann die Lambda Funktion konfiguriert, programmiert und veröffentlicht werden. Nach der erfolgreichen Entwicklung der Lambda Funktion befindet sich der Algorithmus in der Toolbox-Gruppe „Analytics - C++ Lambda Functions“. Im weiteren Verlauf stehen alle nachfolgenden Schritte des Analytics Workflows zur Verfügung. Beispielsweise kann die Lambda Funktion mit den HMI Bedienelementen verknüpft werden. Nach dem Deployment sind die Lambda Funktionen in der TwinCAT Analytics Runtime vorhanden und werden nach entsprechender Konfiguration auf dem generierten HMI One Click Dashboard dargestellt.

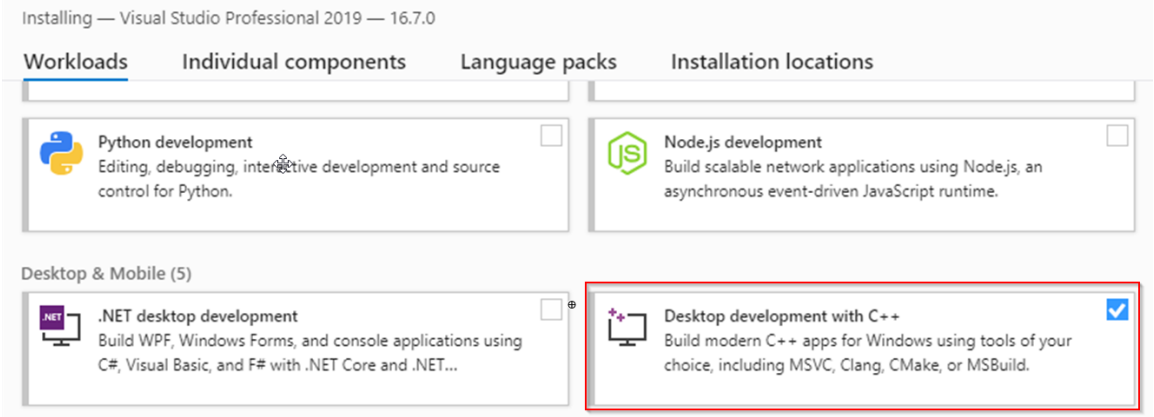
6.4.14.1.1 Anforderungen

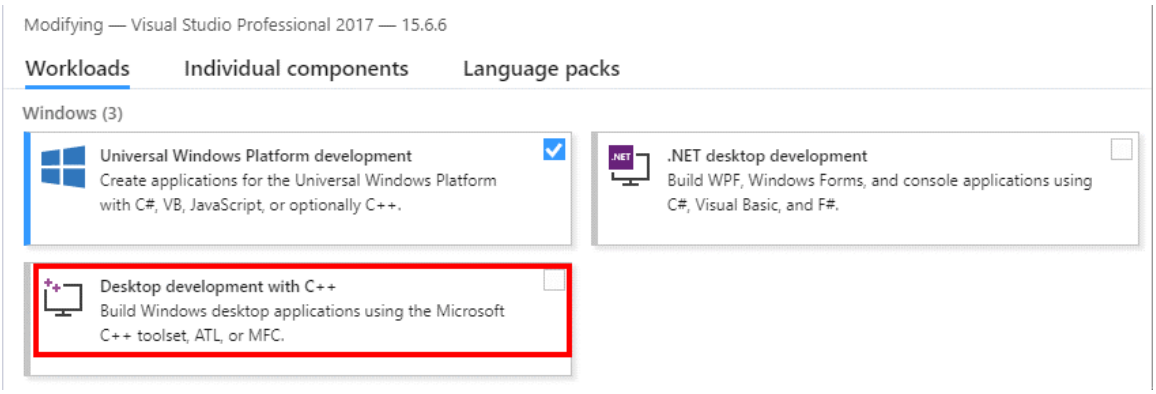
Übersicht der Mindestanforderungen

Die Implementierung und das Debuggen von TwinCAT 3 C++-Modulen erfordert die folgenden Mindestanforderungen.

Auf dem Engineering PC muss installiert sein:

- Microsoft Visual Studio® 2017 oder 2019 Professional / Enterprise.
  - Bei der Installation von Visual Studio® muss zusätzlich die Option zu **Desktop development with C++** ausgewählt werden, da diese Option bei der automatischen Installation nicht selektiert ist:





- TwinCAT 3 Installation (XAE Engineering)
- Für die Integration und Nutzung bestehender, binärer C++-Module in eine TwinCAT 3 SPS-Umgebung ist die XAE-Shell ausreichend (kein Visual Studio® erforderlich).

Auf dem Laufzeit-PC:

- IPC oder Embedded CX PC mit einem der Betriebssysteme Windows 7 / 10 oder TwinCAT/BSD. Es werden sowohl 32 wie auch 64bit Varianten unterstützt.
- Microsoft Visual Studio® muss **nicht** installiert werden.
- TwinCAT 3.1 Installation (XAR-Laufzeit)

6.4.14.1.2 Vorbereitung - nur einmal

Ein PC für das Engineering von Lambda Funktionen und somit von TwinCAT C++-Modulen muss vorbereitet werden. Diese Schritte müssen Sie nur einmal pro System durchführen:

- Konfigurieren Sie die TwinCAT Basis sowie die Konfigurations- und Plattform-Symbolleiste.
- Erstellen Sie ein Testzertifikat, um die Module zu signieren
- Nutzen Sie das TcSignTool, um das Zertifikatspasswort außerhalb des Projektes zu speichern
- Erstellen Sie eine Umgebungsvariable mit dem Zertifikatsnamen

Beim Publishen der C++ Lambda Funktion wird in der Log-Datei eine Übersicht zu den entsprechenden Signatur-Voraussetzungen ausgegeben. In der Log-Datei kann nach „Checking Signing Preconditions“ gesucht werden und die entsprechende Fehlerbehandlung entnommen werden. Eine erfolgreiche Signierung der Treiber sieht folgendermaßen aus:

```
== Checking Signing Preconditions
==> Succeeded: Certificate folder 'C:\TwinCAT\3.1\CustomConfig\Certificates' is valid.
==> Succeeded: Environment variable 'TWINCATCERTIFICATENAME' is set with value 'TwinCATLfCertNew'.
==> Succeeded: Certificate folder 'C:\TwinCAT\3.1\CustomConfig\Certificates' contains a certificate 'TwinCATLfCertNew.tccert'.
==> Succeeded: Environment variable 'TWINCATTESTCERTIFICATE' is set with value 'MyTestSigningCert'.
==> Succeeded: One or more certificate passwords are stored outside of the project with the TcSignTool.
```

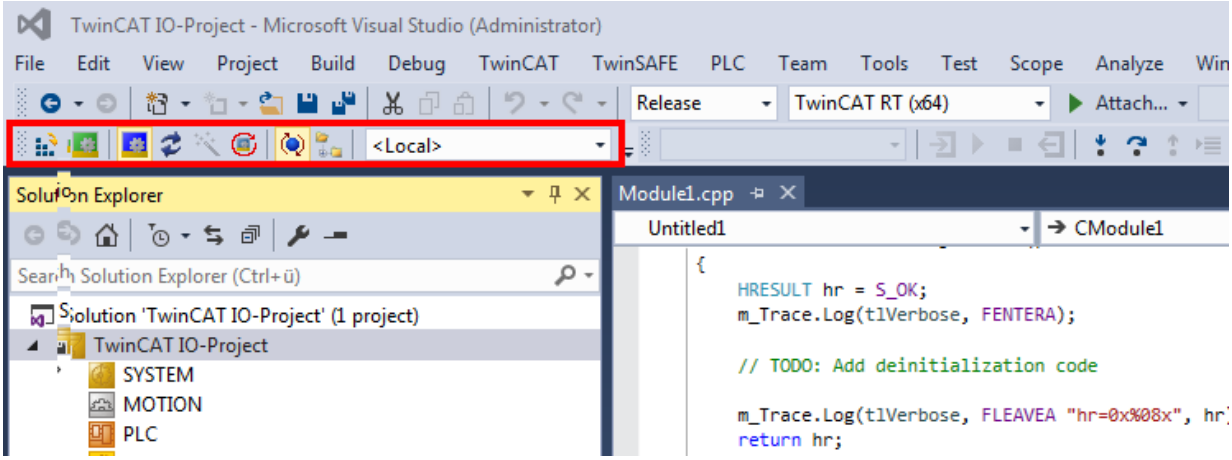
6.4.14.1.2.1 Visual Studio - TwinCAT XAE Base-Symbolleiste

Effizientes Engineering durch TwinCAT XAE Base-Symbolleiste

TwinCAT 3 integriert für eine bessere Effizienz seine eigene Symbolleiste in das Visual Studio-Menü, die Sie bei der Erstellung von C++-Projekten unterstützt. Diese Symbolleiste wird vom TwinCAT 3 Setup automatisch zum Visual Studio-Menü hinzugefügt. Wenn Sie sie allerdings manuell hinzufügen möchten, tun Sie folgendes:

1. Öffnen Sie das Menü **View** und wählen **Toolbars\TwinCAT XAE Base**

⇨ Die ausgewählte Symbolleiste erscheint unter dem Menü.



6.4.14.1.2.2 Treibersignierung

TwinCAT C++-Module müssen durch ein Zertifikat signiert werden, damit sie ausgeführt werden können.

Durch die Signatur wird sichergestellt, dass auf Produktiv-Systemen nur C++-Software ausgeführt wird, deren Herkunft nachvollziehbar ist.

Zu Testzwecken können Zertifikate zum Signieren verwendet werden, die nicht überprüft werden können. Dies ist allerdings nur möglich, wenn das Betriebssystem im Testmodus ist, damit diese Zertifikate nicht auf Produktiv-Systemen genutzt werden.

Engineering erfordert keine Signierung

Lediglich die Ausführung erfordert Zertifikate - das Engineering nicht.

Es gibt zwei Arten, wie Module geladen werden. Hierfür werden unterschiedliche Zertifikate zur Signierung verwendet:

- TwinCAT: Hierbei werden die C++-Module durch das TwinCAT-Laufzeitsystem geladen und müssen dafür mit einem TwinCAT-Nutzerzertifikat signiert sein.
  - Mit TwinCAT 3.1. 4024, und neuer, steht dieses Verfahren zur Verfügung.
  - Dieses Verfahren wird benötigt, um neue Funktionen wie versionierte C++-Projekte und damit auch den C++ Online Change durchzuführen.
  - Für TwinCAT/BSD notwendig.
- (Für <4024.0, nicht mehr empfohlen für Neuprojekte. Bestandsprojekte sollten migriert werden) Betriebssystem: Hierbei werden die C++-Module als normale Kernel-Treiber geladen und müssen somit auch eine Signatur besitzen.
  - Mit TwinCAT 3.1. 4022, oder früher, steht nur dieses Verfahren bereit.
  - Windows 7 (Embedded) x86 (32bit) erfordert keine Signierung.
  - Kann nicht mit TwinCAT/BSD genutzt werden.

Da ein veröffentlichtes Modul auf verschiedenen PCs lauffähig sein soll, ist für eine Veröffentlichung (Publish) immer eine Signierung erforderlich.

Organisatorische Trennung von Entwicklungs- und Produktiv-Software

Beckhoff empfiehlt organisatorisch mit (mindestens) zwei Zertifikaten zu arbeiten.

1. Ein Zertifikat, welches nicht gegengezeichnet ist, somit den Testmodus benötigt, für den Entwicklungsprozess. Dieses Zertifikat kann sich ggf. auch individuell jeder Entwickler selber ausstellen.
2. Nur die Software, welche durch entsprechende finale Tests gelaufen ist, wird durch ein gegengezeichnetes Zertifikat signiert. Diese Software kann damit auch auf Maschinen aufgespielt und ausgeliefert werden.

Eine solche Trennung der Entwicklung und des Betriebes stellt sicher, dass nur getestete Software auf produktiven Anlagen läuft.

6.4.14.1.2.2.1 TwinCAT

Versionierte C++ Projekte werden als Binary in einer TMX-Datei (TwinCAT Module Executeable) abgelegt.

Für die Implementierung von TwinCAT 3 C++ Modulen muss diese kompilierte, ausführbare TMX Datei mit einem TwinCAT-Nutzerzertifikat signiert sein, wenn sie durch die TwinCAT Runtime geladen werden soll.

Signierung

TwinCAT TMX Dateien können nur nach einer erfolgreichen Signierung geladen werden.

HINWEIS

Signierung auf 32bit und 64bit Systemen

Im Gegensatz zu der Betriebssystem-Signierung ist die TwinCAT Signierung sowohl auf 32bit wie auch auf 64bit Systemen vorgesehen. Damit wird für eine Testsignierung auch auf 32bit Systemen der Testmodus vorausgesetzt.

Für die Signierung einer TMX Datei wird ein TwinCAT Nutzerzertifikat benötigt [▶ 282], welches entsprechend im Projekt zur Signierung konfiguriert wird.

Testsignierung

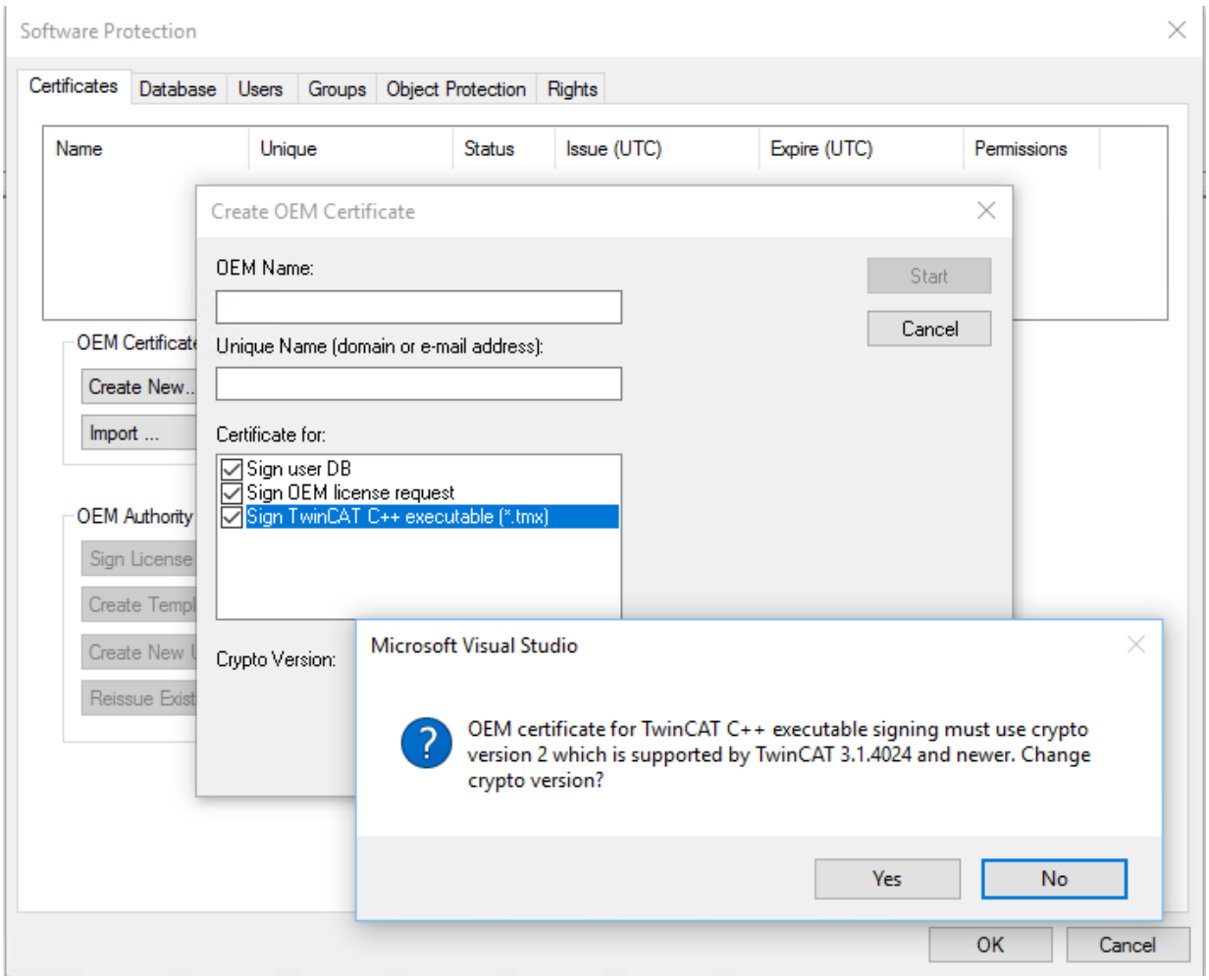
Das Nutzerzertifikat kann lokal in TwinCAT erstellt werden. Solange es nicht von Beckhoff gegengezeichnet ist, ist es nötig den Testmode zu aktivieren, wie hier [▶ 282] beschrieben.

Sobald das TwinCAT-Nutzerzertifikat von Beckhoff gegengezeichnet [▶ 284] wurde, kann auf den Testmodus entsprechend verzichtet werden. Er kann analog zum aktivieren auch wieder deaktiviert werden.

6.4.14.1.2.2.1.1 Testsignierung

Die Testsignierung kann für TwinCAT mit dem gleichen TwinCAT-Nutzerzertifikat erfolgen, wie bei der eigentlichen Auslieferung (vgl.TwinCAT 3 Nutzerzertifikat beantragen [▶ 285]).

1. Für einen Testbetrieb z. B. während der Software-Entwicklung genügt die Erstellung eines TwinCAT-Nutzerzertifikates, wie hier [▶ 286] beschrieben. Beachten Sie dabei, dass Sie den Verwendungszweck „Sign TwinCAT C++ executeable (*.tmx)“ wählen. Hierfür ist dann die Crypto Version 2 erforderlich, eine Meldung dazu erscheint.

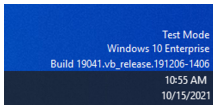


Auf XAR (und XAE, wenn es sich um einen lokalen Test handelt) aktivieren Sie den Testmodus, damit das Betriebssystem die selbstsignierten Zertifikate akzeptieren kann. Dies kann sowohl auf dem Engineering System (XAE) als auch auf dem Laufzeit-Systemen (XAR) vorgenommen werden.

Für Windows

Führen Sie mittels Administrator-Eingabeaufforderung folgendes aus:
`bcdedit /set testsigning yes`
und starten Sie das Zielsystem neu.
Es kann sein, dass Sie hierfür „SecureBoot“ abschalten müssen, was im Bios erfolgen kann.

Wenn der Testsigniermodus aktiviert ist, wird das rechts unten auf dem Desktop angezeigt. Das System akzeptiert nun alle signierten Treiber zur Ausführung.



Für TwinCAT/BSD

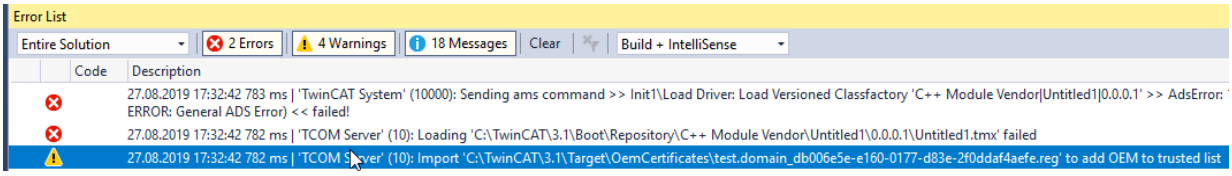
In der Datei `/usr/local/etc/TwinCAT/3.1/TcRegistry.xml` den Eintrag „`<Value Name="EnableTestSigning" Type="DW">1</Value>`“ unter Key „`System`“ eintragen.

```
<Key Name="System">
  <Value Name="RunAsDevice" Type="DW">1</Value>
  <Value Name="RTimeMode" Type="DW">0</Value>
  <Value Name="AmsNetId" Type="BIN">052445B00101</Value>
  <Value Name="LockedMemSize" Type="DW">33554432</Value>
  <Value Name="EnableTestSigning" Type="DW">1</Value>
</Key>
```

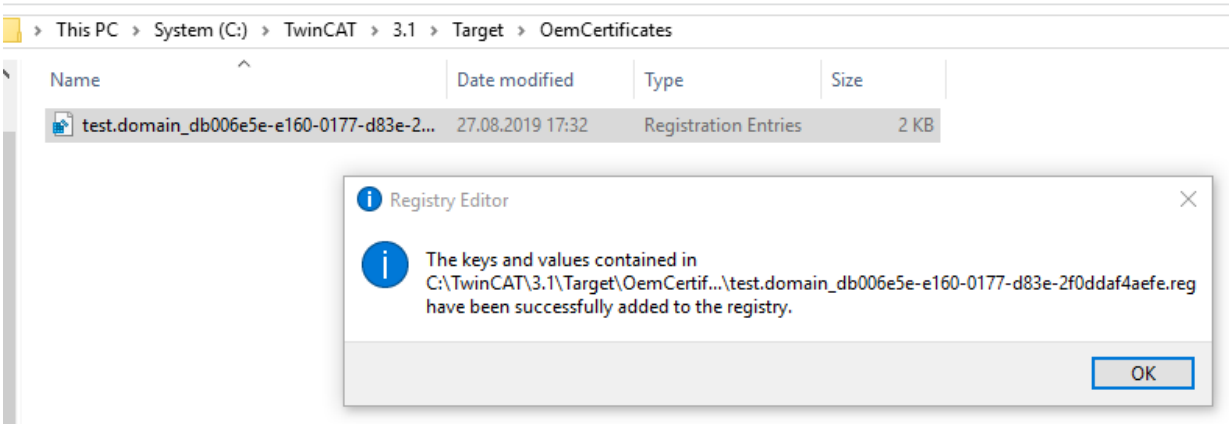
Danach den TwinCAT System Service neu starten:
`doas service TcSystemService restart`

Nach der jeweiligen Vorgehensweise akzeptiert das System alle signierten Treiber zur Ausführung.

1. Bei dem ersten Aktivieren (Activate Configuration) mit einem TwinCAT-Nutzerzertifikat wird von dem Zielsystem festgestellt, dass dem Zertifikat nicht vertraut wird und der Aktivierungsvorgang wird abgebrochen:



Für Windows:
Ein lokaler Nutzer mit Administrationsrechten kann über die erstellte REG-Datei durch einfaches Ausführen dem Zertifikat vertrauen:



Für TwinCAT/BSD:
Falls das Paket „Tcimportcert“ nicht installiert ist dieses installieren: `pkg install tcimportcert`
Mittels `doas tcimportcert /usr/local/etc/TwinCAT/3.1/Target/OemCertificates/<CreatedFile>.reg` dem Zertifikat vertrauen.
Danach den TwinCAT System Service neu starten oder das System neu starten:
`doas service TcSystemService restart`

   ⇨ Diese Verfahren ermöglichen es nur, C++ Module mit einer Signatur von den vertrauten TwinCAT-Nutzerzertifikaten in die Ausführung zu bringen.

2. Nach diesem Prozess können Sie das TwinCAT-Nutzerzertifikat zur Signierung mit dem Testmodus des Betriebssystems verwenden.
   Dieses wird in den Projekt-Eigenschaften konfiguriert.
   Um das Passwort des TwinCAT-Nutzerzertifikates nicht im Projekt abzulegen, wo es beispielsweise auch in einer Versionsverwaltung landen würde, verwenden Sie das TcSignTool [▶ 296].

Wenn Sie das TwinCAT-Nutzerzertifikat ohne TestMode zur Auslieferung nutzen wollen, müssen Sie das Zertifikat von Beckhoff gegenzeichnen [▶ 284] lassen.

6.4.14.1.2.2.1.2 Signierte TwinCAT-Nutzerzertifikate zur Auslieferung ohne Testmode

Systemvoraussetzungen
- Min. TwinCAT 3.1 Build 4024
- Min. Windows 10 oder TwinCAT/BSD (auf dem Zielsystem)

Mit TwinCAT Build 4024 bietet Beckhoff Bestandskunden die Ausstellung eines „TwinCAT 3 OEM-Nutzerzertifikates“ an, das für die Signierung von mit TwinCAT 3 in C++ erstellte TMX-Dateien verwendet werden kann.

- Dieses Zertifikat erfordert durch die Nutzung im Windows-Umfeld eine sichere Validierung der Antragstellerdaten. TwinCAT 3 Nutzerzertifikate müssen daher zur Validierung der Adress- und Kontaktdaten offiziell bestellt werden, und werden nur an Beckhoff Bestandskunden vergeben.
- Bestellnummer: **TC0008**
- Die Ausstellung dieses TwinCAT 3 Nutzerzertifikates ist kostenlos.
- Verzeichnis zum Speichern des Zertifikates: **C:\TwinCAT\3.1\CustomConfig\Certificates**

Das TwinCAT 3 Nutzerzertifikat ist nicht für die Nutzung der TwinCAT 3 TMX-Dateien erforderlich

Das TwinCAT 3 Nutzerzertifikat wird ausschließlich für die einmalige Signierung der TMX-Dateien verwendet., und ist nicht für die Nutzung der damit signierten TMX-Dateien erforderlich.

Auf welchen Rechnern ist das TwinCAT 3 Nutzerzertifikat TC0008 erforderlich?

Das TwinCAT 3 Nutzerzertifikat sollte sich ausschließlich auf dem Engineering-Rechner befinden, auf dem die Signierung der TMX-Dateien erfolgt – also **NICHT** auf jedem Zielsystem.

Gültigkeit des TwinCAT 3 Nutzerzertifikats

Die Gültigkeit des TwinCAT 3 Nutzerzertifikats ist aus Sicherheitsgründen auf zwei Jahre begrenzt.

Was passiert, wenn die Gültigkeit des Zertifikats abgelaufen ist?

Sie können neue TMX-Dateien nicht mehr signieren.
Die Nutzung der bereits signierter TMX-Dateien ist aber ohne Einschränkung weiter möglich.

Sie können vor Ablauf der zwei Jahre (und auch noch danach) eine Verlängerung Ihres Zertifikats beantragen.

Um ein TwinCAT 3 Nutzerzertifikat zu verlängern, gilt der gleiche Prozess wie bei der Beantragung eines neuen Zertifikats. Auch in diesem Fall muss das Zertifikat bestellt werden (die Bestellnummern für eine Zertifikatsverlängerung sind dieselben wie zur Zertifikatsneubeantragung).

Sie erzeugen in diesem Falle aber kein neues „OEM Certificate Request File“, sondern schicken Ihr bestehendes Zertifikat zur Verlängerung an die Beckhoff Zertifikatsstelle. Teilen Sie in der Email mit, das es sich um eine Zertifikatsverlängerung, und keine Neuausstellung handelt. Für den restlichen Inhalt der Email gelten ansonsten dieselben Kriterien wie bei der Beantragung eines neuen Zertifikates.

Das bestehende Zertifikat erhält ein neues Ablaufdatum, wird dann neu signiert und ist weitere 2 Jahre gültig.

Das neu signierte Zertifikat ist damit vollständig kompatibel zur Ursprungsversion.

6.4.14.1.2.2.1.2.1 TwinCAT 3 Nutzerzertifikat beantragen

Übersicht über den Bestell- und Validierungsprozess

Die Beantragung eines TwinCAT Nutzerzertifikates erfordert eine offizielle Bestellung.

- Bestellnummer: **TC0008** (TwinCAT 3 Certificate Extended Validation)
- Die Ausstellung (und Verlängerung) eines TwinCAT 3 Nutzerzertifikates ist kostenlos.
- Da es sich bei einem TwinCAT 3 Nutzerzertifikat um einen digitalen Ausweis handelt, ist eine Verifizierung der Kontaktdaten des Anfragers nach den marktüblichen Standards erforderlich.
- Die Ausstellung eines TwinCAT 3 Nutzerzertifikates erfolgt daher nur für Beckhoff Bestandskunden.

Ihre Email-Adresse muss ein Firmen-Email-Account sein (Freemailer wie GMail oder Ähnliches sind nicht zulässig) und mit dem Firmennamen des Bestellers korrespondieren.

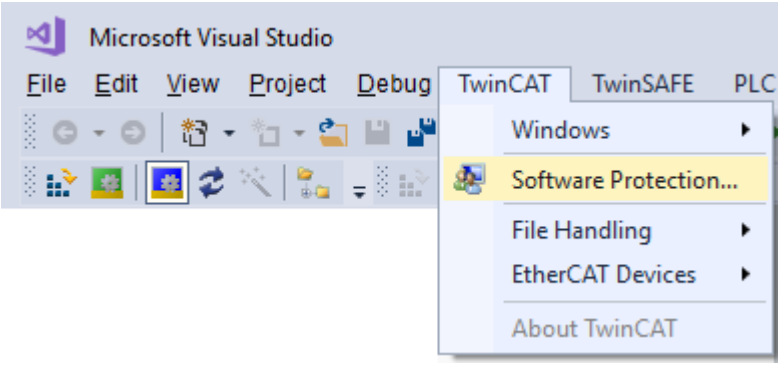
1. Kontaktieren Sie Ihren Beckhoff Vertriebskontakt und kündigen Sie die Beantragung eines TwinCAT 3 OEM-Zertifikats an. Bestellen Sie “TC0008”.
2. Wichtig: Geben Sie, als Anfrager, Ihre Kontaktdaten als Lieferadresse (= Kontaktname und Email-Adresse) und den Einsatzbereich des Zertifikats an (Firmenname, Adresse).
3. Die im Auftrag angegebenen Kontaktdaten werden verifiziert und Sie (der in der Lieferadresse genannte Anfrager) werden vom Beckhoff Vertrieb kontaktiert.
4. Bei der Beantragung eines neuen OEM-Zertifikats erstellen Sie im TwinCAT 3 Engineering eine „Certificate Request Datei“ [▶ 286].
5. Ermitteln Sie mit Hilfe des TwinCAT Engineerings den „File Fingerprint“ der OEM-Zertifikatsdatei (siehe File Fingerprint der Zertifikatsdatei ermitteln [▶ 289]). Teilen Sie diesen File Fingerprint im Rahmen Ihrer Kontaktdatenverifizierung dem Beckhoff Vertriebskontakt mit. Die Übermittlung des File Fingerprints muss auf einem anderen Kommunikationsweg erfolgen als für die Zusendung der OEM-Zertifikatsanfragedatei.
6. Schicken Sie die „OEM-Zertifikatsdatei“ nun an den Beckhoff Vertriebskontakt.
7. Nach der Signierung der Zertifikatsdatei in der Beckhoff Zentrale erhalten Sie diese über Ihren Ansprechpartner per Email zugesandt.

Beachten Sie, dass die Validierung der Kontaktdaten und die Ausstellung des Zertifikats einige Tage Zeit in Anspruch nehmen können.

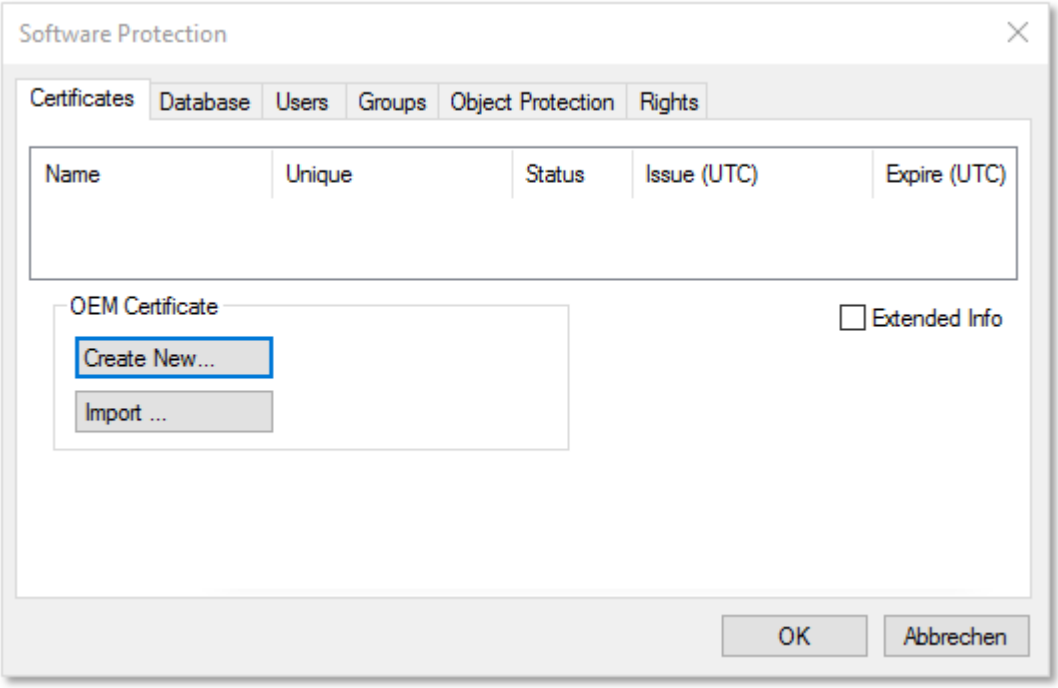
6.4.14.1.2.2.1.2.2 Erstellung der Certificate Request Datei für TC0008

Systemvoraussetzungen
- Min. TwinCAT 3.1 Build 4024
- Min. Windows 10 oder TwinCAT/BSD (auf dem Zielsystem)

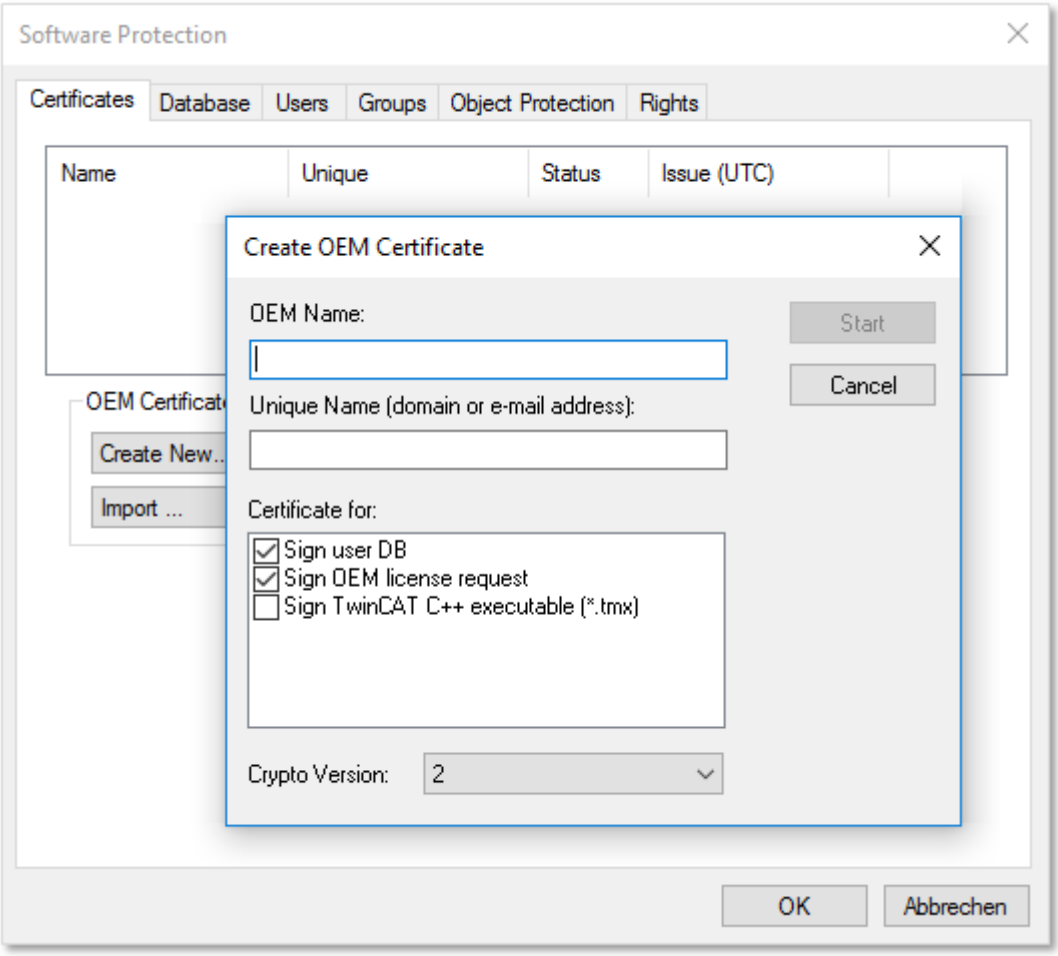
1. Rufen Sie den Konfigurator der Software Protection auf. Wählen Sie dazu im Hauptmenü unterhalb des Punktes **TwinCAT** den Menüpunkt **Software Protection** aus:



2. In dem sich öffnenden Fenster wählen Sie die Registerkarte **Certificates** aus.
   Klicken Sie auf **Create New….**:

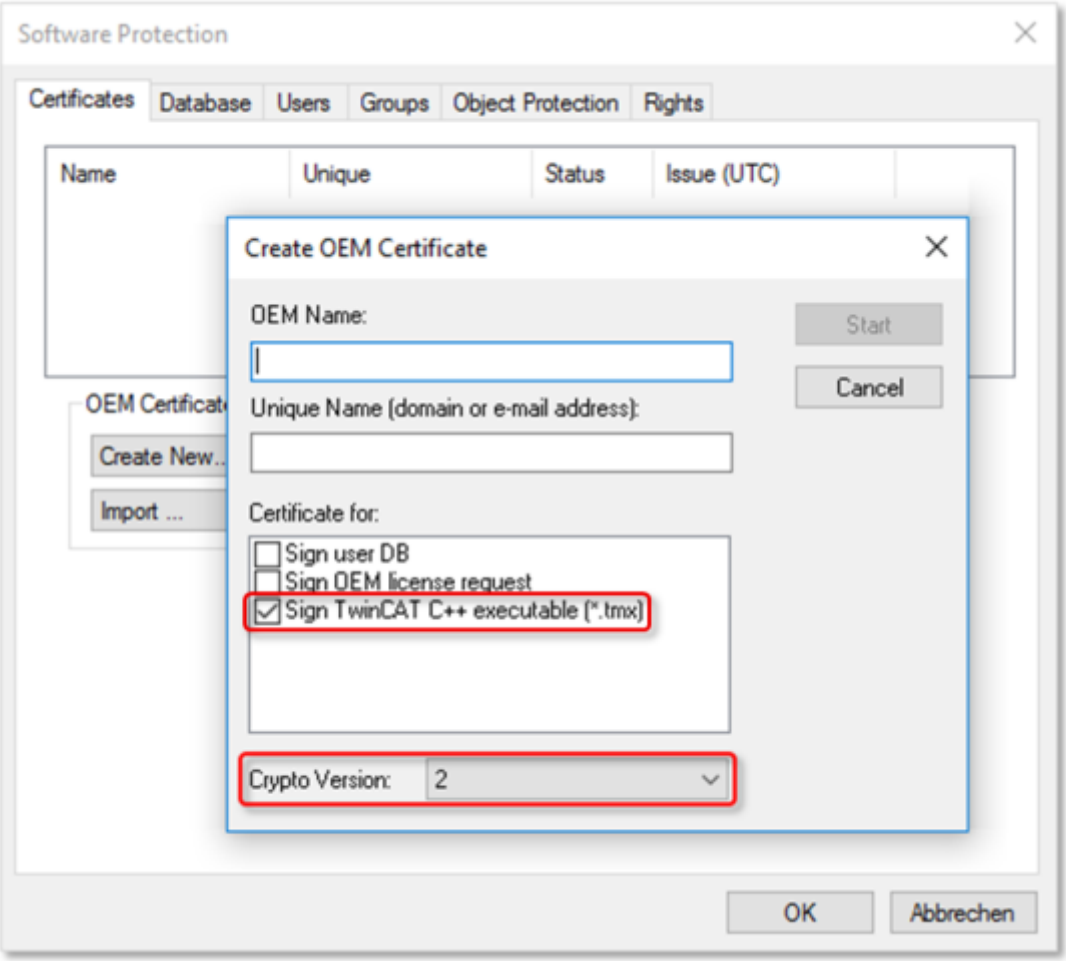


3. Das Eingabefenster **Create OEM Certificate** öffnet sich:



4. Geben Sie die erforderlichen Daten ein:
   - Geben Sie im Textfeld **OEM Name** Ihren Firmennamen ein. Der Name muss einen klaren Bezug zu Ihrer Firma oder Ihrem Unternehmensteil haben.

- Geben Sie einen **Unique Name** ein. Der „OEM Unique Name“ muss ein einmaliger Name sein, anhand dessen der Eigentümer des Zertifikats weltweit eindeutig identifiziert werden kann, vorzugsweise die URL der Webseite Ihrer Firma oder Ihre E-Mail-Adresse. Die E-Mail-Adresse muss eine Firmen-E-Mail-Adresse sein, also eindeutig Ihrer Firma zugeordnet werden können.
- Markieren Sie die Checkbox **Sign TwinCAT C++ executables**:



Wenn Sie mit diesem Zertifikat nur TwinCAT Treiber Software signieren wollen, schalten Sie die anderen beiden Checkboxen aus. (Diese werden nur im PLC-Bereich genutzt.)

- Achten Sie darauf, dass die Crypto-Version 2 (für den verschlüsselten Inhalt des Zertifikatsinhaltes) eingestellt ist. (Standardeinstellung)

5. Wenn Sie die Daten eingegeben haben, klicken Sie auf **Start** und Sie können ein Verzeichnis auswählen, um die Datei zu speichern.
   Übernehmen Sie einfach das vorgeschlagene Verzeichnis „c:\twincat\3.1\customconfig\certificates“. Sie benötigen die neu erzeugte Datei in diesem Verzeichnis, um in einem späteren Schritt den „File Fingerprint“ für diese Datei auslesen [▶ 289] **zu können.**
   ⇨ Ein Dialog zur Auswahl eines Passworts für den OEM Private Key öffnet sich.
6. Vergeben Sie ein Passwort für den OEM Private Key**.**

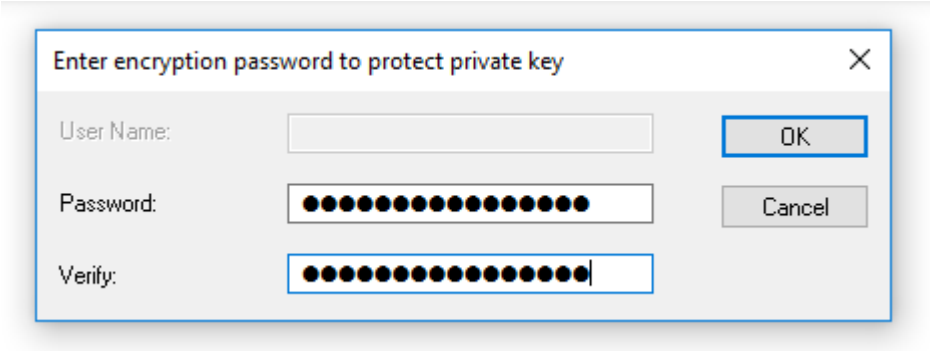
Wichtig: Passwort-Sicherheit!

Verwenden Sie unbedingt ein starkes Passwort für Ihr Zertifikat!
Schützen Sie Ihr Passwort durch geeignete Maßnahmen, damit es nicht in fremde Hände fallen kann!

Passwort bei Verlust nicht wiederherstellbar

Beckhoff kann Ihr Passwort nicht wiederherstellen oder zurücksetzen. Wenn Sie das Passwort für Ihr Zertifikat vergessen oder verlieren, können Sie das Zertifikat nicht mehr verwenden und müssen ein neues Zertifikat ausstellen lassen.

7. Bestätigen Sie das Passwort durch eine wiederholte Eingabe und schließen Sie den Dialog mit **OK**.



⇨ Die Datei wird gespeichert.

Die so erzeugte „Certificate Request Datei“ muss nun noch von der Beckhoff Zertifikatsstelle signiert werden, um gültig zu sein. Das Verfahren dazu wird im Kapitel Zertifikat beantragen [▶ 285] beschrieben.

6.4.14.1.2.2.1.2.3 File Fingerprint der OEM-Zertifikatsdatei ermitteln

Diese Funktionalität benötigen Sie für die Beantragung eines **TwinCAT OEM Certificate Extended Validation** (TC0008).

Systemvoraussetzung

Diese Funktionalität erfordert mindestens die Version TwinCAT 3.1 Build 4024.

Die OEM Certificate Request Datei wird durch die Signierung von Beckhoff zum TwinCAT OEM-Zertifikat. Bis auf diese Signatur unterscheiden sich die Dateien nicht. Daher wird im Folgenden für beide Dateiversionen der Begriff „TwinCAT OEM-Zertifikatsdatei“ verwendet.

Den „File Fingerprint“ einer OEM-Zertifikatsdatei über das TwinCAT 3 Engineering auslesen

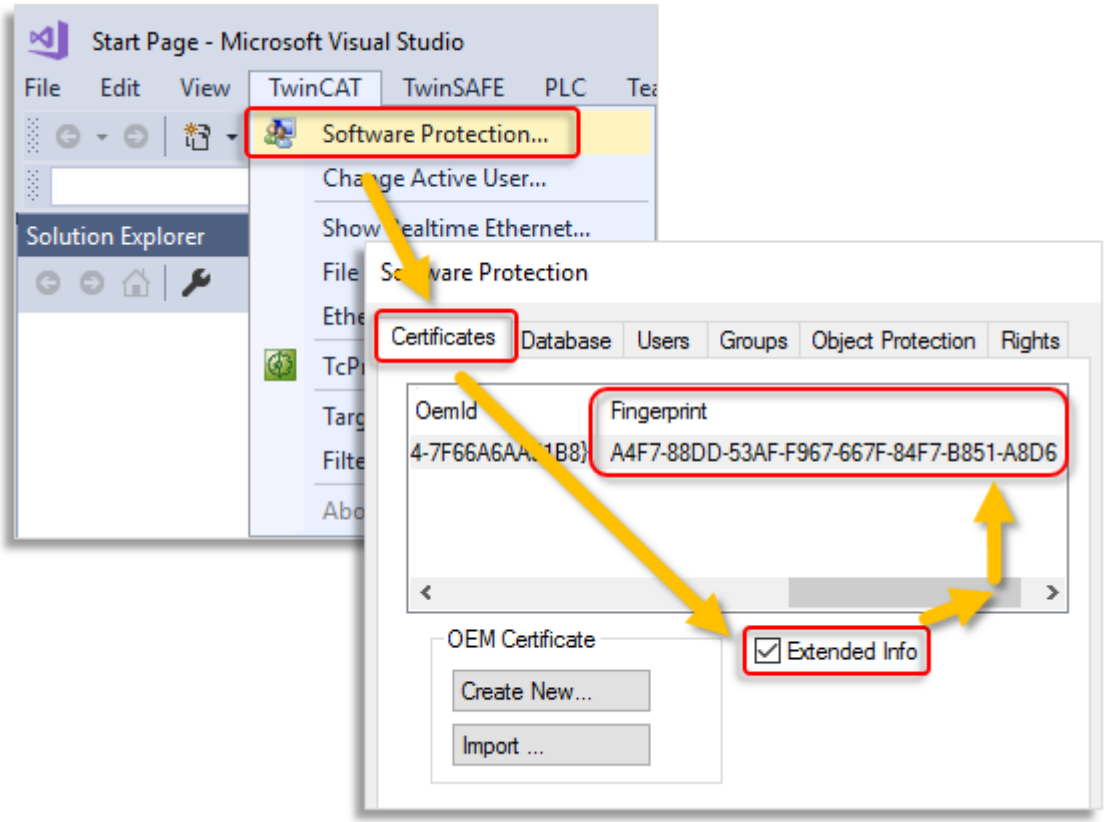
Für diese Funktion ist es erforderlich, dass die OEM-Zertifikatsdatei in diesem Verzeichnis liegt: „c:\twincat\3.1\customconfig\certificates“.

In diesem Verzeichnis liegt ihr OEM-Zertifikat, sofern Sie bereits eines haben und dieses verlängern wollen.

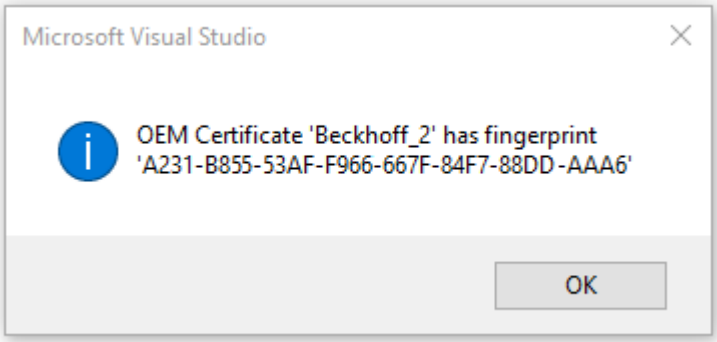
Sofern Sie beim Erstellen der „OEM Certificate Request Datei“ das vorgeschlagene Verzeichnis nicht geändert haben, liegt die Datei bereits in diesem Verzeichnis.

Vorgehensweise:

1. Rufen Sie den TwinCAT 3 Software Protection Konfigurator auf.



2. Wählen Sie den Tab **Certificates** aus.
3. Markieren Sie die Checkbox **Extended Info.**
4. Scrollen Sie im Fenster so weit nach rechts, bis Sie die Spalte **Fingerprint** sehen. (Alternativ können Sie auch einfach einen Doppelklick auf die Zertifikatszeile machen. In einem Popup-Fenster wird dann der File Fingerprint angezeigt):



Mit [Ctrl] + [C] können Sie die Fingerprint-Daten aus dem Meldungsfenster ins Windows Clipboard kopieren.

6.4.14.1.2.2.1.2.4 Speichern des fertig signierten TwinCAT Nutzerzertifikates

Empfohlenes Verzeichnis zum Speichern des Zertifikates: **C:\TwinCAT\3.1\CustomConfig\Certificates**

Systemvoraussetzungen

- Min. TwinCAT 3.1 Build 4024
- Min. Windows 10 oder TwinCAT/BSD (auf dem Zielsystem)

Das TwinCAT 3 Nutzerzertifikat ist nicht für die Nutzung der TwinCAT 3 TMX-Dateien erforderlich

Das TwinCAT 3 Nutzerzertifikat wird ausschließlich für die einmalige Signierung der TMX-Dateien verwendet., und ist nicht für die Nutzung der damit signierten TMX-Dateien erforderlich.

Auf welchen Rechnern ist das TwinCAT 3 Nutzerzertifikat TC0008 erforderlich?

Das TwinCAT 3 Nutzerzertifikat sollte sich ausschließlich auf dem Engineering-Rechner befinden, auf dem die Signierung der TMX-Dateien erfolgt – also **NICHT** auf jedem Zielsystem.

6.4.14.1.2.2.2 Betriebssystem

HINWEIS

Migration zu TMX mit TwinCAT Loader empfohlen

Seit TwinCAT 3.1 4024.0 stehen versionierte C++ Projekte bereit, deren Binaries direkt von TwinCAT geladen werden können. Eine Migration wird empfohlen!

Für die Implementierung von TwinCAT 3 C++ Modulen auf x64-Plattformen muss der Treiber (*.sys Datei) mit einem Zertifikat signiert sein, wenn er durch das Betriebssystem geladen werden soll.

Die Signatur, die beim TwinCAT 3 Build-Prozess automatisch vorgenommen wird, wird von 64-Bit-Windows-Betriebssystemen für die Authentifizierung der Treiber verwendet.

Für die Signierung eines Treibers wird ein Zertifikat benötigt. Diese Microsoft Dokumentation beschreibt den Prozess und Hintergrundwissen zum Erhalt eines Test- und Freigabezertifikats, das von 64-Bit-Windows-Betriebssystemen akzeptiert wird.

Um ein solches Zertifikat in TwinCAT 3 zu verwenden, konfigurieren Sie den Schritt nach Kompilieren Ihres x64 Build-Targets wie in „Ein Testzertifikat für den Testmodus erstellen [▶ 291]“ dokumentiert.

Testzertifikate

Zum Testen können selbstsignierte Testzertifikate ohne technische Limitierung erstellt und verwendet werden.

In den folgenden Lernprogrammen ist beschrieben, wie diese Möglichkeit aktiviert wird.
Um Treiber mit echten Zertifikaten für Produktionsmaschinen zu erstellen, muss diese Möglichkeit deaktiviert werden.

- Ein Testzertifikat für den Testmodus erstellen [▶ 291]
- (Test-)Zertifikate löschen [▶ 294]

Weitere Referenzen:

MSDN, MakeCert Testzertifikate (Windows Treiber),
https://docs.microsoft.com/en-us/windows-hardware/drivers/install/makecert-test-certificate

6.4.14.1.2.2.2.1 Testsignierung

Übersicht

Für die Implementierung von TwinCAT 3 C++ Modulen auf x64-Plattformen muss der Treiber mit einem Zertifikat signiert sein.

Dieser Artikel beschreibt, wie Sie ein Testzertifikat zum Testen eines C++ Treibers erstellen und installieren können.

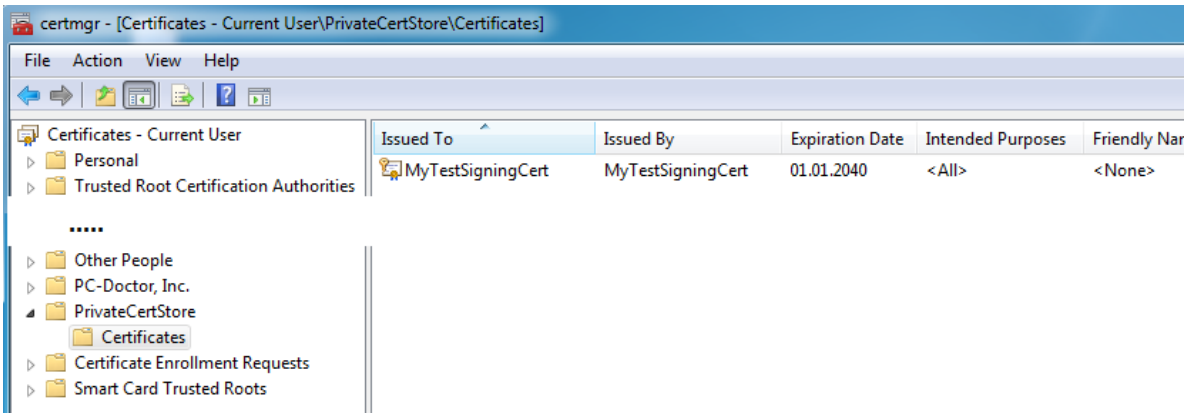
Vorgehen beim Erstellen von Testzertifikaten beachten

Entwickler verfügen möglicherweise über eine Vielzahl an Werkzeugen, um selber Zertifikate zu erstellen - bitte folgen Sie dieser Beschreibung genauestens, um den Testzertifikat-Mechanismus anzuschalten.

Die folgenden Befehle müssen in einer Kommandozeile ausgeführt werden, die auf eine der beiden Arten geöffnet wurde:

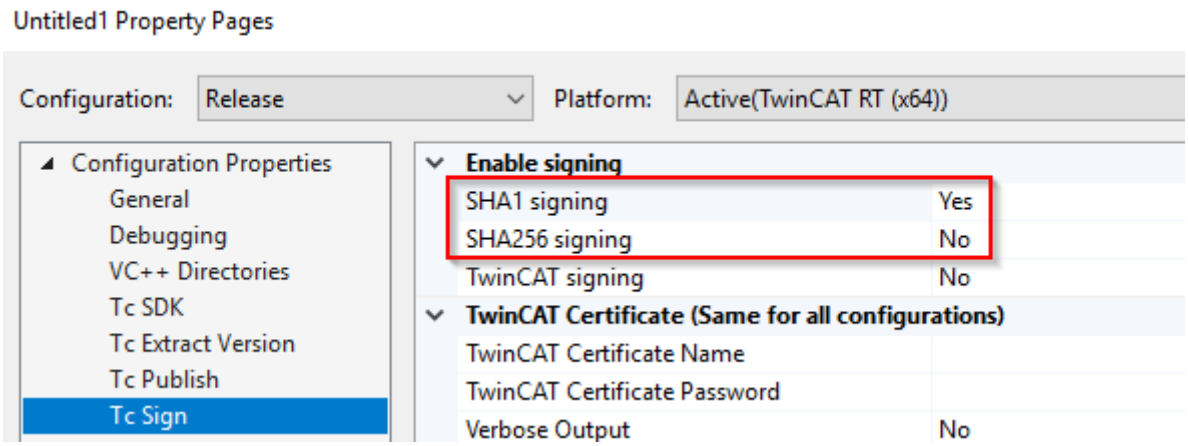
- **Eingabeaufforderung von Visual Studio 2010 / 2012 mit Administratorrechten**. (Über: **All Programs** -> **Microsoft Visual Studio 2010/2012** -> **Visual Studio Tools** -> Visual Studio Command Prompt, dann Rechtsklick auf **Run as administrator**)
- **Developer Command Prompt von Visual Studio 2017 / 2019 mit Administratorrechten**. (Über: **All Programs** -> **Visual Studio 2017** -> Visual Studio Command Prompt for VS 2017/2019, dann Rechtsklick auf **Run as administrator**)
- Nur, wenn das WINDDK installiert wurde:
  Normale Eingabeaufforderung (**Start** -> Command Prompt) mit Administratorrechten, dann in das Verzeichnis *%WINDDK7%\bin\x86* wechseln, das die entsprechenden Tools enthält.

1. Auf XAE:
   geben Sie im Engineering System folgenden Befehl in die Visual Studio 2010 / 2012 Eingabeaufforderung mit Administratorrechten ein (siehe Hinweis oben):
   ```
   makecert -r -pe -ss PrivateCertStore -n CN=MyTestSigningCert
   MyTestSigningCert.cer
   ```
   (Sollten Sie keine Zugriffsrechte auf den PrivateCertStore haben, können Sie eine andere Ablage nutzen. Diese muss dann im PostBuild-Event auch verwendet werden, wie hier [▶ 295] **beschrieben.)**
   ⇨ Daraufhin wird ein selbstsigniertes Zertifikat erstellt und in der Datei „MyTestSigningCert.cer“ und im Windows Certificate Store gespeichert.
   ⇨ Überprüfen Sie das Ergebnis mit mmc (**Use File->Add/Remove Snap-in->Certificates**):



2. Auf XAE:
   konfigurieren Sie das Zertifikat, sodass es von TwinCAT XAE auf dem Engineering System erkannt wird. Setzen Sie die Umgebungsvariable TWINCATTESTCERTIFICATE auf „MyTestSigningCert“ im Engineering System oder bearbeiten Sie jeweils den Post-Build Event von Debug|TwinCAT RT (x64) und Release|TwinCAT RT (x64).
   Der Name der Variablen ist NICHT der Name der Zertifikatsdatei, sondern der CN-Name (in diesem Falle MyTestSigningCert).

Hinweis **Ab TwinCAT 3.1 4024.0 wird die Konfiguration des zu verwendenden Zertifikates unter dem Punkt Tc Sign in den Projekt-Eigenschaften vorgenommen. Wenn Sie die Signierung über das Betriebssystem verwenden wollen, wie sie hier beschrieben ist, dann achten Sie bitte auf die Projekt-Einstellungen:**

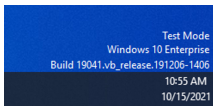


Auf XAR (und XAE, wenn es sich um einen lokalen Test handelt) aktivieren Sie den Testmodus, damit das Betriebssystem die selbstsignierten Zertifikate akzeptieren kann. Dies kann sowohl auf dem Engineering System (XAE) als auch auf dem Laufzeit-Systemen (XAR) vorgenommen werden.

Für Windows

Führen Sie mittels Administrator-Eingabeaufforderung folgendes aus:
`bcdedit /set testsigning yes`
und starten Sie das Zielsystem neu.
Es kann sein, dass Sie hierfür „SecureBoot“ abschalten müssen, was im Bios erfolgen kann.

Wenn der Testsigniermodus aktiviert ist, wird das rechts unten auf dem Desktop angezeigt. Das System akzeptiert nun alle signierten Treiber zur Ausführung.



Nach der jeweiligen Vorgehensweise akzeptiert das System alle signierten Treiber zur Ausführung.

1. Prüfen Sie, ob eine Konfiguration mit einem in einen TwinCAT C++ Treiber implementierten TwinCAT-Modul auf dem Zielsystem aktiviert und gestartet werden kann.

⇨ Die Kompilierung des x64-Treibers generiert eine Ausgabe wie folgt:

```
Output
Show output from: Build
1>------ Build started: Project: Untitled2, Configuration: Debug TwinCAT RT (x64) ------
1>  header file << C:\TwinCAT\3.1\SDK\\_products\TwinCAT RT (x64)\Debug\Untitled2\\Untitled2Version.h >> is up-to-date!
1>  TcPch.cpp
1>  Module1.cpp
1>  Untitled2ClassFactory.cpp
1>  Untitled2Driver.cpp
1>  Untitled2.vcxproj -> C:\TwinCAT\3.1\SDK\\_products\TwinCAT RT (x64)\Debug\Untitled2.sys
1>  The following certificate was selected:
1>      Issued to: MyTestSigningCert
1>
1>      Issued by: MyTestSigningCert
1>
1>      Expires:   Sun Jan 01 00:59:59 2040
1>
1>      SHA1 hash: E27A66E6A0C7BC0C86DFDD093DDF2486D1EE502E
1>
1>
1>  Done Adding Additional Store
1>  Successfully signed and timestamped: C:\TwinCAT\3.1\SDK\\_products\TwinCAT RT (x64)\Debug\Untitled2.sys
1>
1>
1>  Number of files successfully Signed: 1
1>
1>  Number of warnings: 0
1>
1>  Number of errors: 0
1>
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Referenzen:
MSDN, MakeCert Testzertifikate (Windows Treiber),
https://docs.microsoft.com/en-us/windows-hardware/drivers/install/makecert-test-certificate

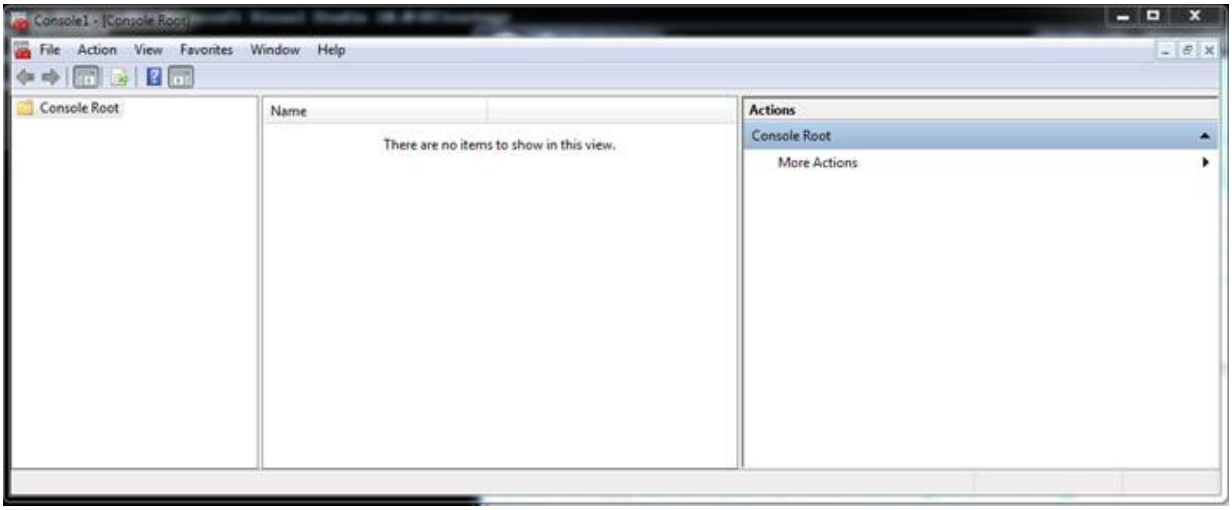
6.4.14.1.2.2.2.2 Testzertifikat löschen

Dieser Artikel beschreibt, wie ein Testzertifikat gelöscht wird.

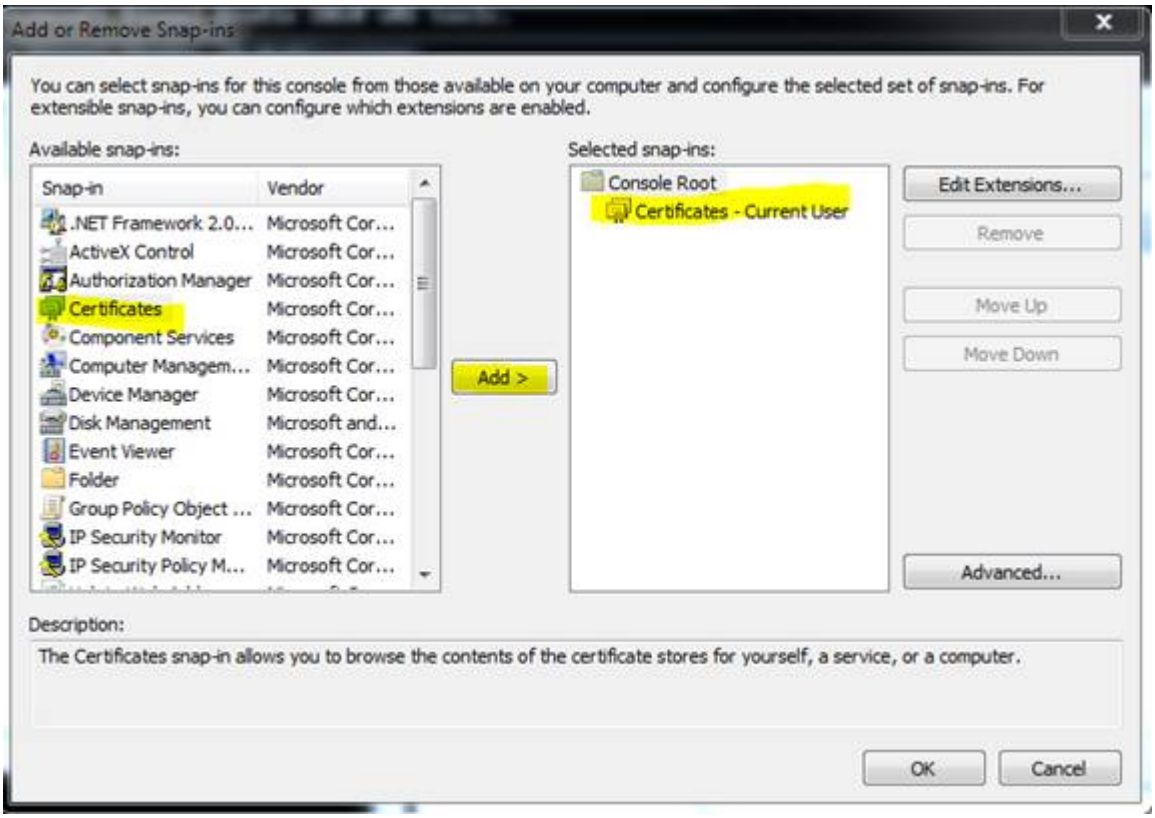
Übersicht

Ein Zertifikat kann mit der Microsoft Management Console gelöscht werden:

1. Starten Sie die Management console MMC.exe über Start-Menü oder die Oberfläche.

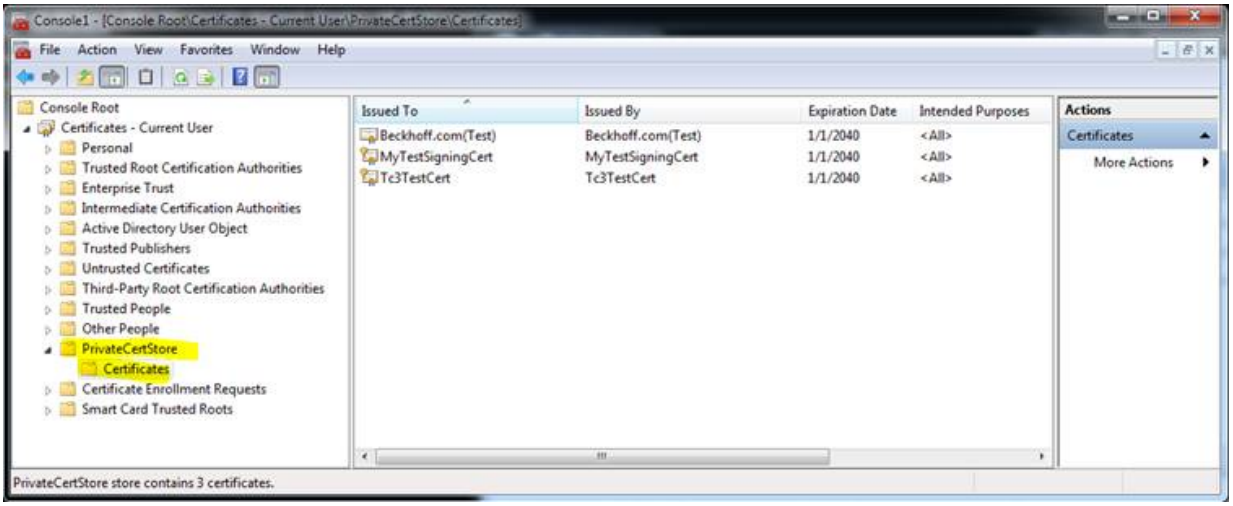


2. Klicken Sie im Menü auf **File** -> **Add/Remove Snap-in..** und wählen Sie Zertifikat-Snap-In für den aktuellen Nutzer, schließen Sie mit **OK** ab.



⇨ Die Zertifikate sind im Knoten unter **PrivateCertStore/Certificates** aufgelistet.

3. Wählen Sie das zu löschende Zertifikat aus.



6.4.14.1.2.2.2.3 Windows Treiber ohne Testmode

Für Windows Betriebssysteme ist eine Signierung des Treibers über das „Attestation Signing“ nötig. Hierfür wird ein sogenanntes EV-Zertifikat benötigt.
Microsoft stellt hierfür eine Anleitung bereit: https://docs.microsoft.com/en-us/windows-hardware/drivers/dashboard/attestation-signing-a-kernel-driver-for-public-release
Die hierdurch erzeugten Treiber sind auch geeignet für Geräte mit angeschaltetem Secure-Boot.

Das frühere Verfahren mittels CrossSigning-Zertifikaten (Signtool mit Parameter /ac) ist von Microsoft zum Juli 2021 abgekündigt. Es kann (in Abhängigkeit von dem Ablaufzeitpunkt des individuellen CrossSigning-Zertifikats) nicht mehr genutzt werden, wie hier dokumentiert ist.

Für TwinCAT C++ existieren seit längerem die Versionierten C++ Projekte, welche über den TwinCAT Loader geladen werden, sodass sie keine Treiber im Sinne des Betriebssystems sind. Beckhoff empfiehlt deswegen auf die Versionierten C++ Projekte zu setzen.

Bei der Migration von TwinCAT C++ Treibern zu Versionierten C++ Projekten existiert eine Anleitung im How-to Bereich.

6.4.14.1.2.3 TcSignTool – Ablage des Zertifikatspasswords außerhalb des Projektes

Das TcSignTool-kann verwendet werden, um ein Passwort zu einem TwinCAT-Nutzerzertifikat in der Registry abzulegen. Somit benötigt man das Passwort nicht in den Projekten, wo die Passwörter ggf. unbeabsichtigt in Versionskontrollsystemen landen würden.

Das TcSignTool ist ein Kommandozeilen-Programm, welches sich im Pfad *C:\TwinCAT\3.x\sdk\Bin* befindet.

Die Ablage des Passworts wird mit folgenden Parametern durchgeführt:

```
tcsigntool grant /f "C:\TwinCAT\3.1\CustomConfig\Certificates\MyCertificate.tccert" /p MyPassword
```

Mit folgenden Parametern wird das Passwort gelöscht:

```
tcsigntool grant /f "C:\TwinCAT\3.1\CustomConfig\Certificates\MyCertificate.tccert" /r
```

Die Ablage des unverschlüsselten Passworts erfolgt unter
`HKEY_CURRENT_USER\SOFTWARE\Beckhoff\TcSignTool\`

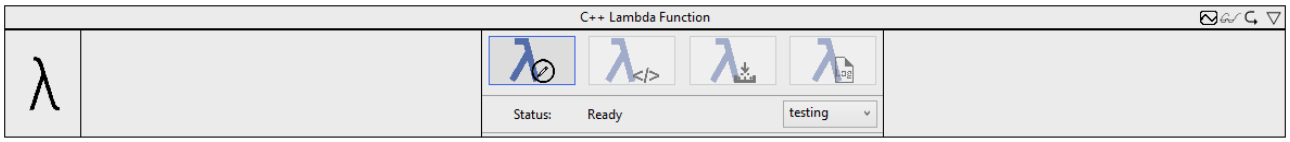
6.4.14.1.2.4 Umgebungsvariablen

Abschließend müssen zwei Umgebungsvariablen mit dem Zertifikatsnamen erstellen werden. Diese werden während des Generierens der Lambda Funktion ausgelesen und signieren die Lambda Funktion mit den entsprechenden Zertifikaten.

Name der Variablen	Wert der Variablen
TWINCATTESTCERTIFICATE	<Betriebssystem-Zertifikatsname> (Bsp.: MyTestSigningCert)
TWINCATCERTIFICATENAME	<TwinCAT-Zertifikatsname> (Bsp.: TwinCATTestCert)

Die Dateiendungen (.cert und .tccert) dürfen nicht mit angegeben werden.

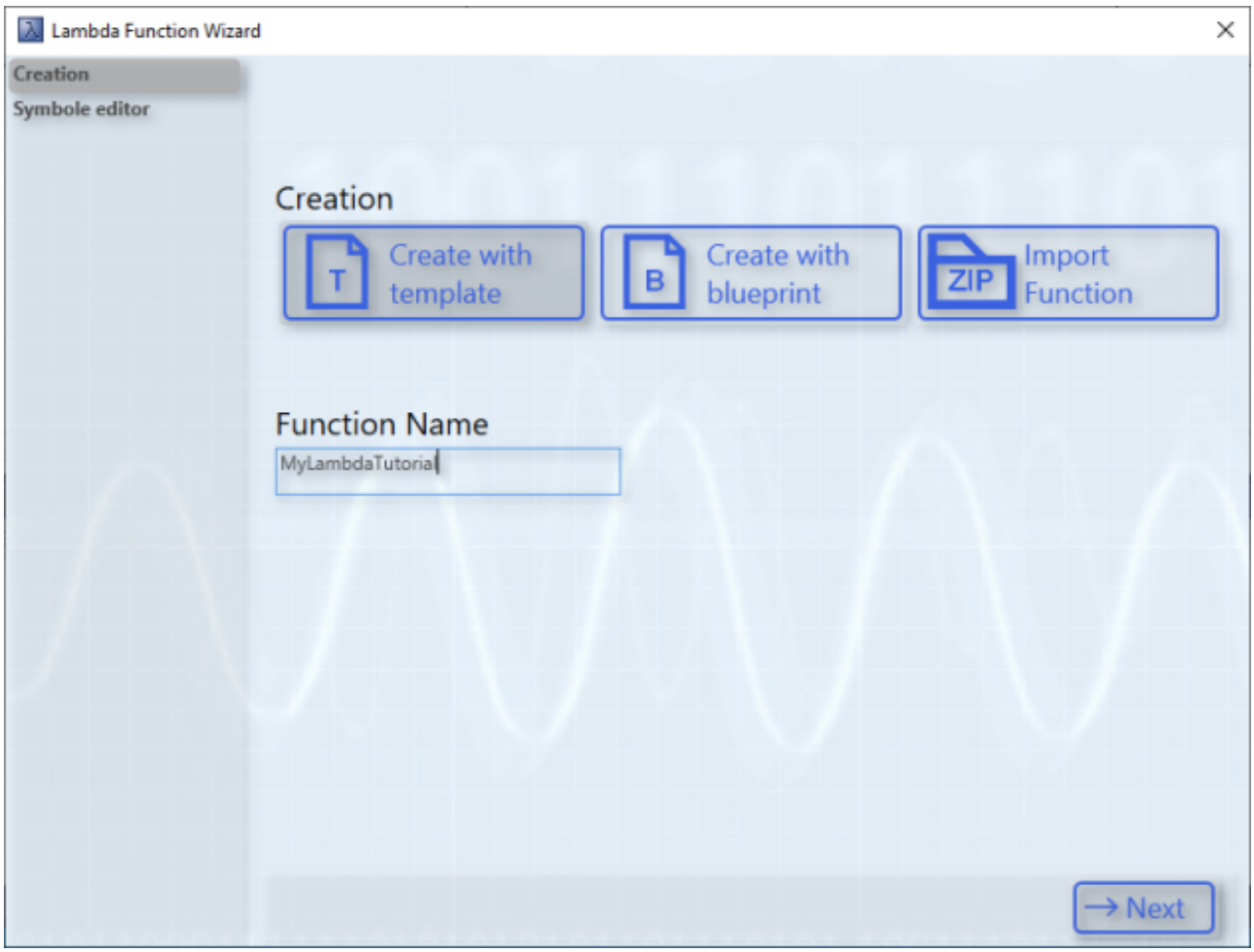
6.4.14.1.3 Lambda Template



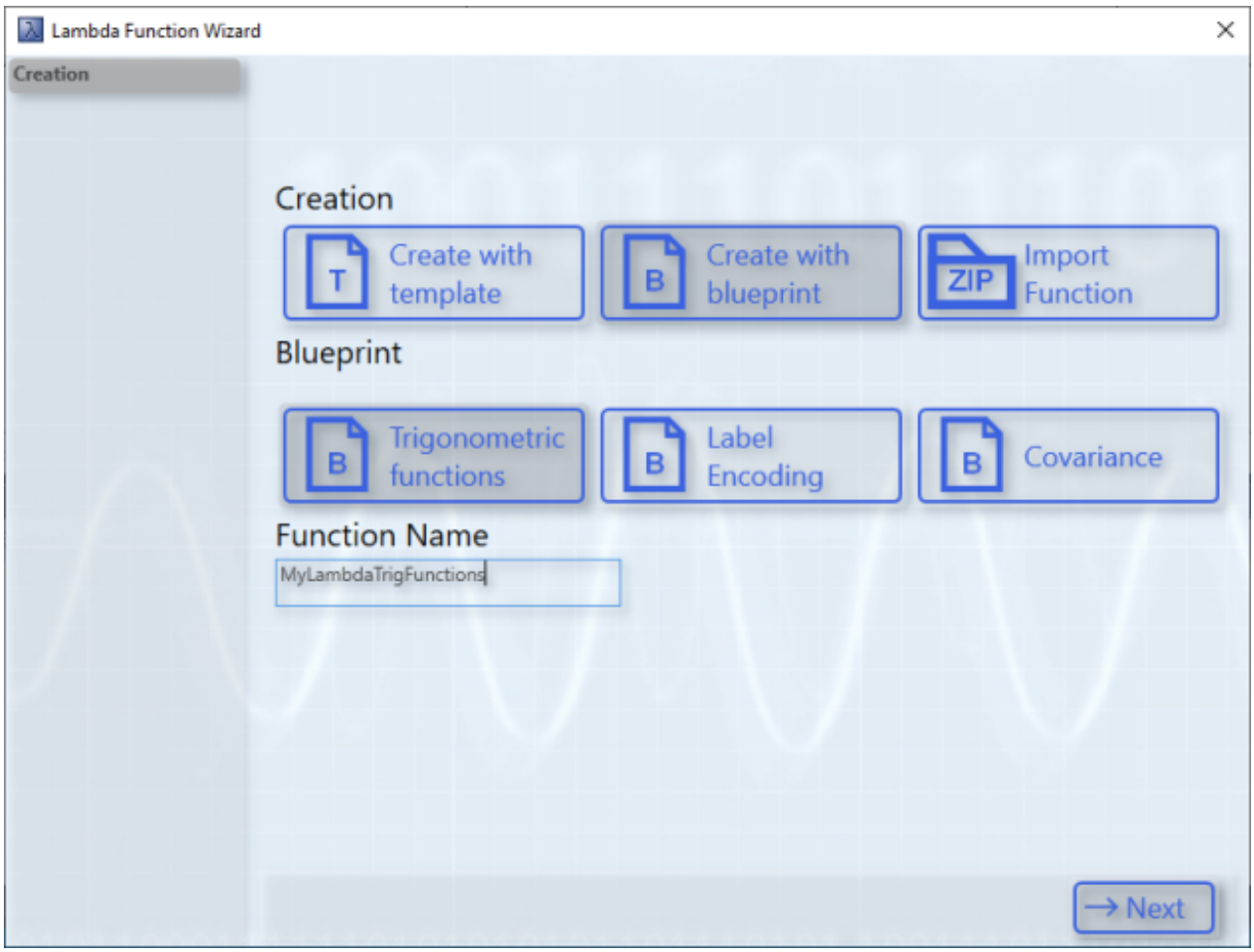
Das Lambda Template leitet durch die Entwicklung einer Lambda Funktion. Innerhalb des Parameter-Bereichs befindet sich ein Control. Dieses Control enthält vier Buttons zum Konfigurieren, zum Programmieren und zum Publishen der Lambda Funktion. Der vierte Button ist zum Einsehen der Log-Dateien des Entwickelungsprozesses. Eine Statusleiste liefert zusätzliche Informationen während des Prozesses. Ein Drop-Down-Menü unten rechts bietet die Möglichkeit, zwischen einer Release- und einer Testing-Version der Lambda Funktion zu wählen.

Konfigurieren der Lambda Funktion

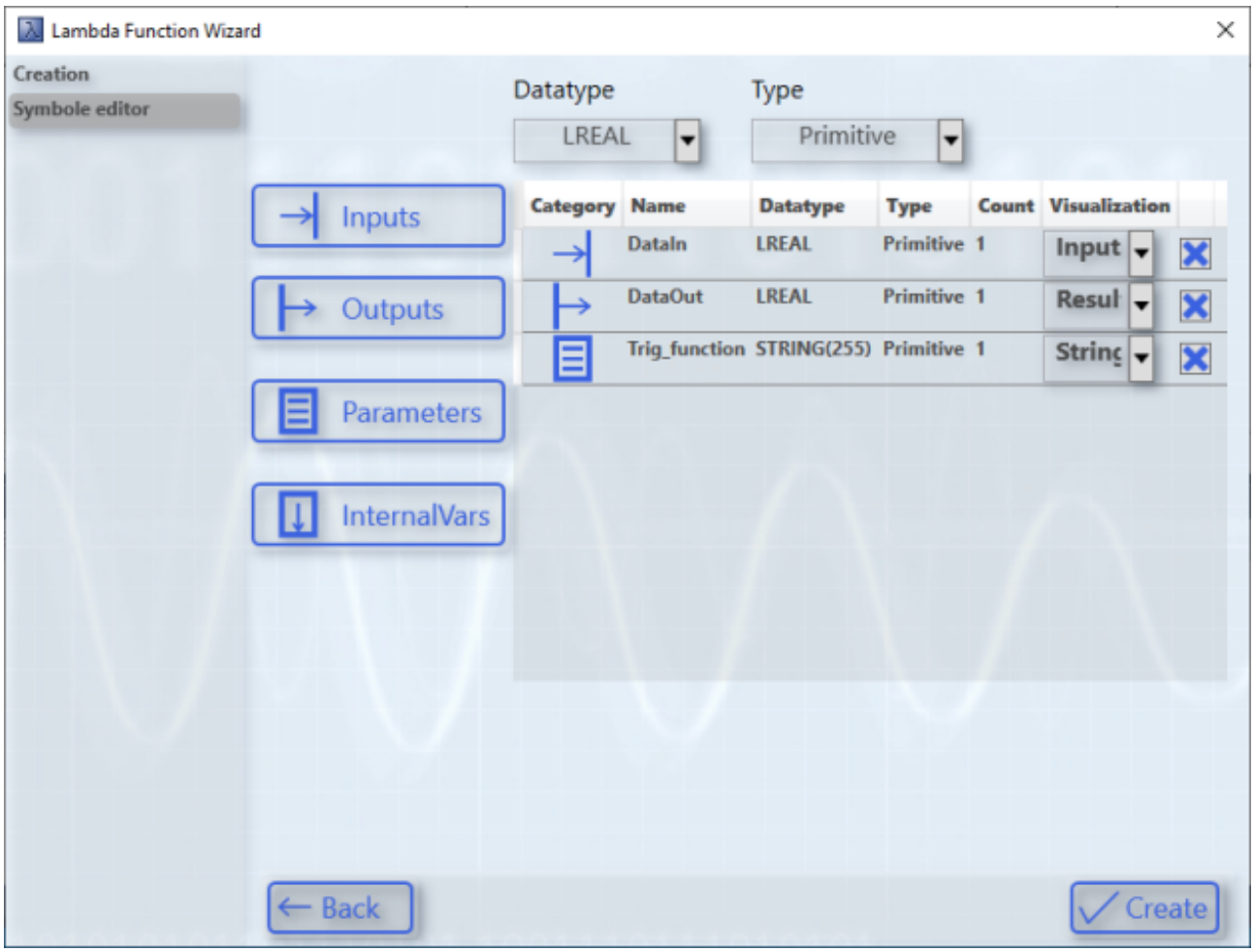
Der erste Button von links öffnet einen Wizard. Mithilfe dieses Wizards können Sie die Lambda Funktion konfigurieren.



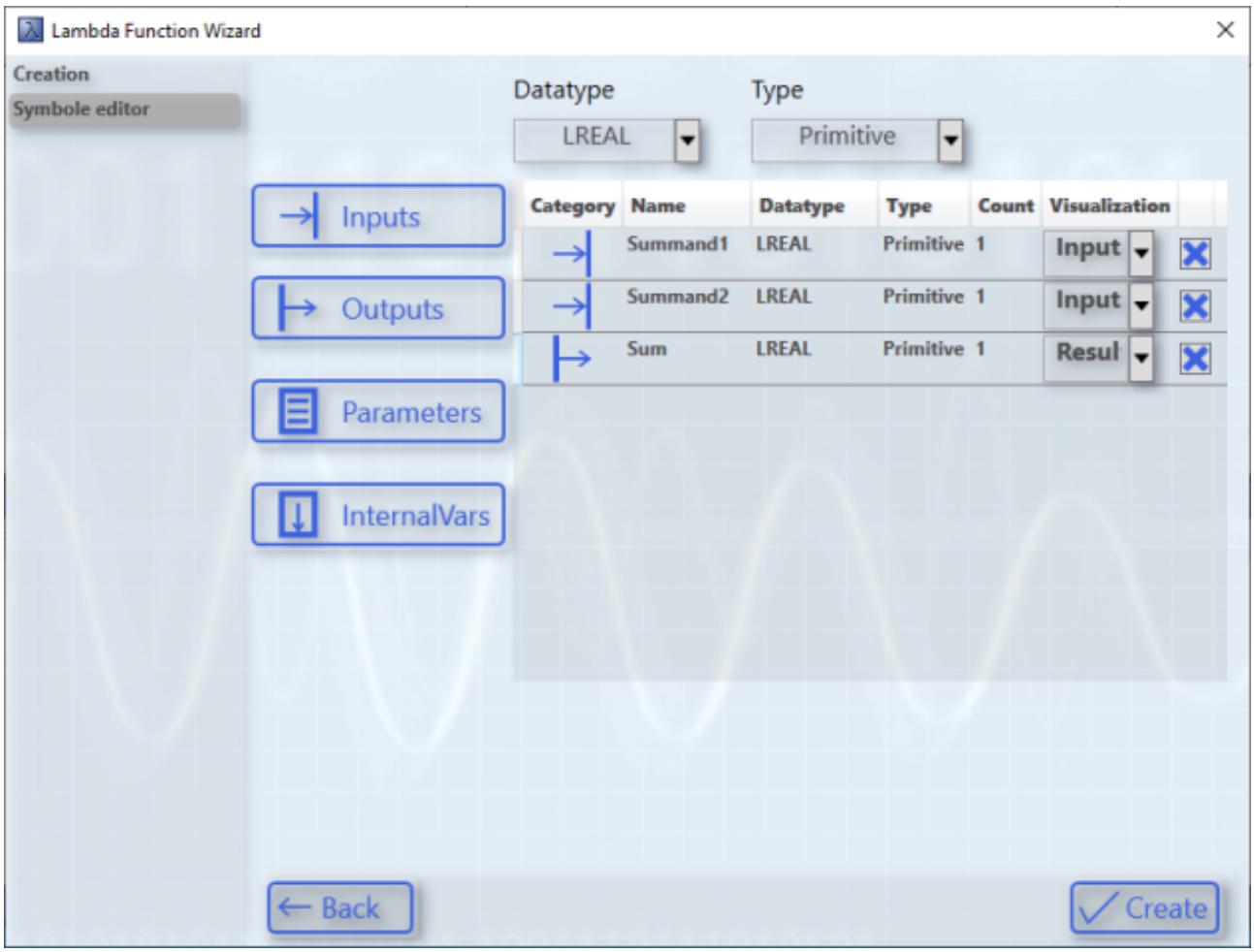
Auf der ersten Seite können Sie zwischen drei Modi auswählen. Der erste Modus heißt Template. Mit diesem Modus können neue Lambda Funktionen entwickelt werden. Alle Dateien und Abhängigkeiten werden in einer Vorlage vorbereitet und können mit anwenderspezifischer Logik implementiert werden.



Der zweite Modus heißt Blueprint. Es gibt jeweils drei bereits implementierte Blueprints. Es gibt jeweils einen Blueprint für das Berechnen trigonometrischer Funktionen, zum Label Encoding und zur Berechnung der Kovarianz. Diese Blueprints dienen als beispielhafte Konfiguration und Implementierung einer Lambda Funktion. Genauere Informationen sind unter dem folgenden Link verfügbar (siehe: Lambda Blueprints [▶ 306]).
Über den dritten Modus importieren Sie exportierte Lambda Funktionen. Mit einem Klick auf den Button **Import Function**, öffnet sich ein Datei-Öffnen-Dialog. Anschließend können Sie die exportierte Lambda Funktion, die als ZIP-Archiv vorliegen sollte, auswählen. Sobald die Lambda Funktion importiert ist, ist der Name zu erkennen und auf der nächsten Seite sind die konfigurierten Symbole dargestellt.
Der Name der Lambda Funktion ist individuell wählbar. Jedoch sollte die Versionierung im Namen der Lambda Funktion nicht angepasst oder verändert werden. Andernfalls kann die einwandfreie Nutzung der Lambda Funktionen beeinträchtigt werden. Ebenfalls sollte der Funktionsname mindestens vier Zeichen enthalten.
Auf der zweiten Seite des Wizard konfigurieren Sie die Symbole und Variablen der Lambda Funktion.
Bei dem Blueprint-Modus sind bereits einige Symbole vorkonfiguriert.



Bei dem Template-Modus ist die Liste der Symbole leer. Wie in der folgenden Abbildung dargestellt, können Sie anwenderspezifische Symbole konfigurieren.



Über die beiden Drop-Down-Boxen ganz oben auf der Seite wählen Sie jeweils den Datentyp und den Datenstrukturtyp aus.

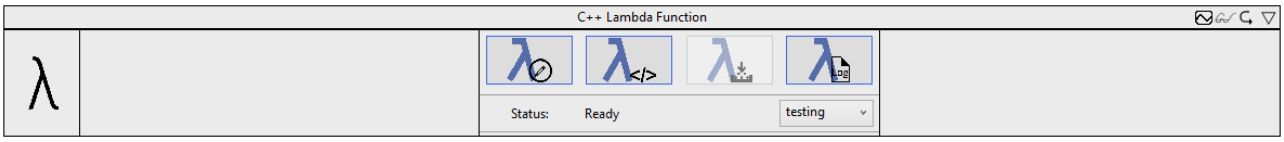
Verfügbare Datentypen				
LREAL	REAL	BOOL	WORD	INT
LINT	ULINT	LTIME	STRING(255)	

Bei den Datenstrukturtypen kann zwischen primitiv und Array gewählt werden.
Fügen Sie über die Buttons rechts die Variablen mit dem ausgewählten Datentyp und Datenstrukturtyp hinzu.
Interne Variablen sind Member-Variablen. Sie werden in der Oberfläche des Algorithmus nicht dargestellt und können in der Analytics Workbench nicht verknüpft werden. Sie haben den Zweck, Variablenwerte über Zyklen zu speichern.
Wenn der Datenstrukturtyp Array bei einer Variablen ausgewählt ist, kann in der Spalte Count der Variable die Größe des Arrays angegeben werden. Um mehrdimensionale Arrays zu erstellen, müssen die Dimensionen mit dem Buchstaben „x“ separiert werden (Beispiel: 2x4).
Die Visualisierung des Symbols in der Workbench können Sie über das Drop-Down-Menü konfigurieren. Je nach Datentyp sind unterschiedliche Visualisierungen möglich. Undefinierte Visualisierungen sind nicht möglich. Bei den internen Variablen kann keine Visualisierung ausgewählt werden, weil sie in der Workbench nicht angezeigt werden. Die folgenden Visualisierungen sind bei den jeweiligen Datentypen für Eingänge, Ausgänge und Parameter möglich:

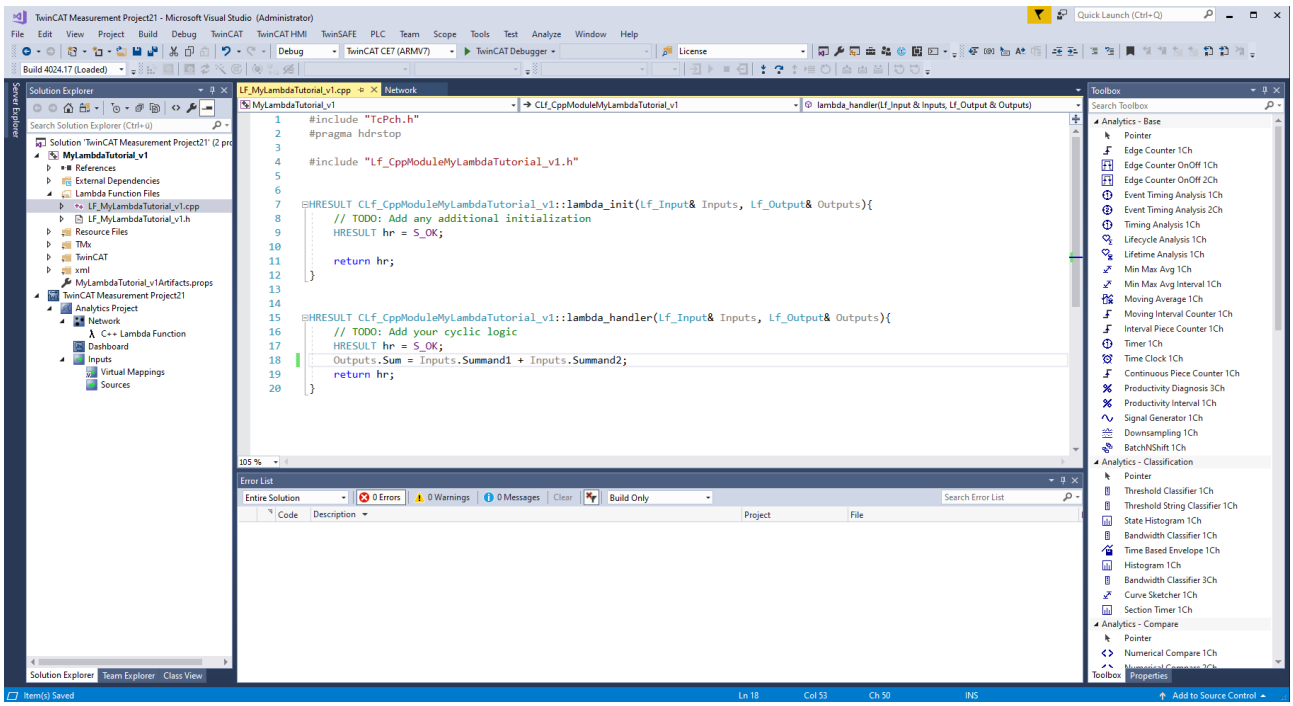
Datentyp	Visualisierung	Beispiel
STRING(255)	String (Input)	
	String (Output)	Output String String Result
	String (Parameter)	String Level 1 String Result
	FileOpen/FileSave (Parameter)	File Path TwinCAT\3.1\Boot\Teach.tas ...
ULINT	Timestamp (Output)	Last Event 15.11.2021 11:14:57.048
BOOL	Boolean (Input)	Enable Execution New Result @ String Compare 1Ch False
	Boolean (Output)	New Result False
	Boolean (Parameter)	Use Absolute Values
REAL	Input (Input)	Input Variables.fTriangular @ Virtual Input (2),... -1,4508
	Result (Output)	Avg -0,224
INT	Count (Output)	Count 6
LINT	Timespan (Output)	Elapsed Time 04:35:28:298
	Clock (Parameter)	Time On hh 12 mm 0 ss 0
	Interval (Parameter)	Interval Seconds 30
WORD	DayOfWeekMask (Parameter)	Day Of Week Mask Mon Tue Wed Thu Fri Sat Sun
LREAL	Input (Input)	Input Variables.fTriangular @ Virtual Input (2),... -1,4508
	Result (Output)	Avg -0,224
	Level (Parameter)	Tolerance 1,25

Nach einem Klick auf den Create-Button wird das C++-Projekt im Hintergrund generiert und der Wizard wird geschlossen.

Programmieren der Lambda Funktion



Nachdem Sie den zweiten Button geklickt haben, wird das C++ Projekt in den Solution Explorer hinzugefügt. Das Projekt enthält alle Dateien, Abhängigkeiten und Konfiguration, um ein lauffähiges TcCOM-Modul für die Analytics Workbench als auch für die Nutzung im Echtzeit-Kontext zu entwickeln.



Die automatisch geöffnete Datei enthält merhere gekapselte Funktionen. Die sogenannte lambda_init-Methode ist für die Initialisierung von Variablen. Die lambda_handler-Methode wird zyklisch ausgeführt. Die lambda_uninit-Methode ist für die Deinitialisierung von Variablen. In den Methoden stehen die Inputs, Outputs, Parameter und interne Variablen als Strukturen in den Methodenparametern zur Verfügung. Die folgenden Eingänge und Ausgänge sind standardmäßig in den Lambda Funktionen verfügbar.

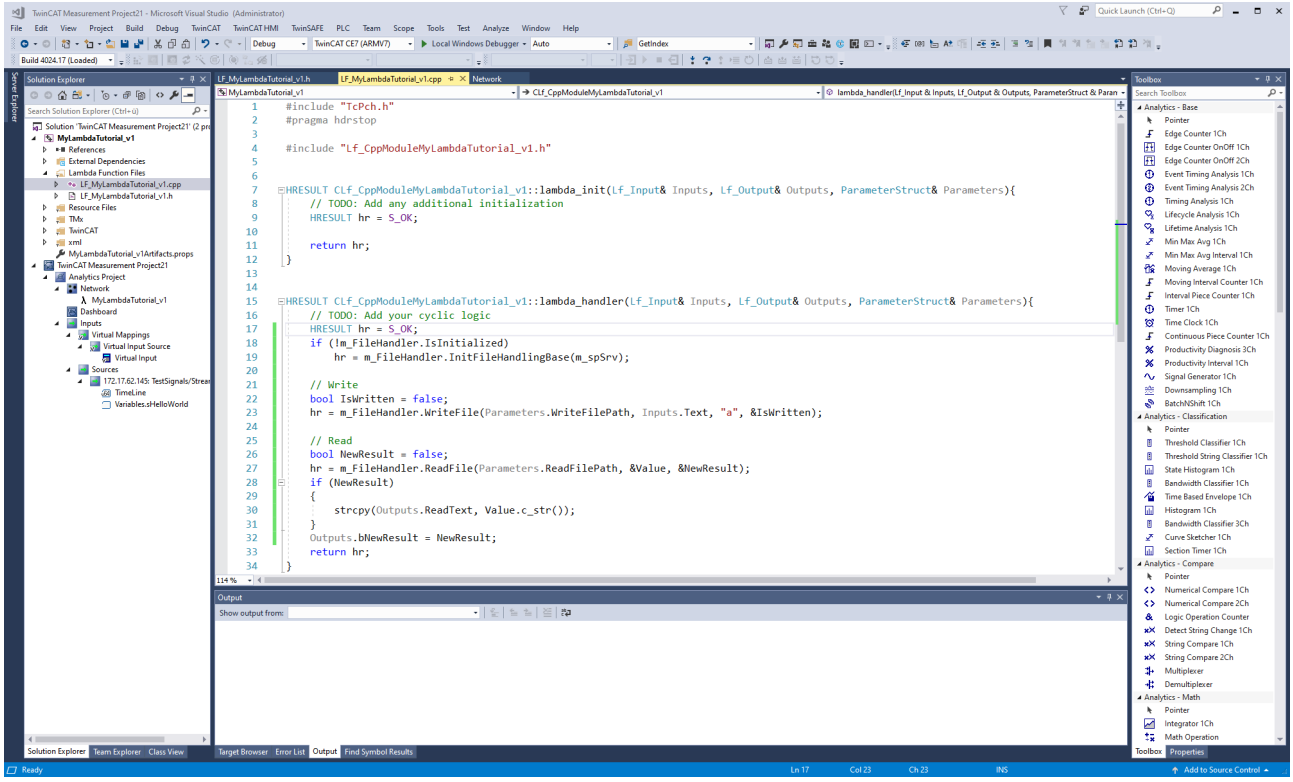
Kategorie	Symbolname	Datentyp	Beschreibung	Sichtbarkeit
Input	tTimestamp	ULINT	Die Variable enthält den TwinCAT Zeitstempel.	Versteckt
	bReset	BOOL	Die Variable setzt die konfigurierten Symbole zurück.	Optional
	bEnableExecution	BOOL	Die Variable aktiviert die lambda_handler-Methode.	Optional
Output	bNewResult	BOOL	Die Variable gibt an, ob der Algorithmus neue Ergebnisse liefert.	Optional

Variablen, die innerhalb der Methoden erstellt werden, verlieren ihren Inhalt nach einem Zyklus. Interne Variablen bieten die Möglichkeiten, Variablenwerte über Zyklen hinweg zu speichern. In den beiden Lambda-Methoden steht ein Objekt mit dem Namen „m_FileHandler“ zur Verfügung. Dieses Objekt müssen Sie einmalig initialisieren, sofern dies noch nicht geschehen ist. Anschließend können Sie die zwei Funktionen „ReadFile“ und „WriteFile“ des Objektes verwenden. Die Funktionen haben folgende Eigenschaften:

Funktionsname	Parameter				Rückgabewert
InitFileHandlingBase	ITComObjectServer* ipSrv				Hresult
ReadFile	PCCH filename	Std::string* value	bool* bNewResult		Hresult
WriteFile	PCCH filename	Std::string value	Std::string mode	bool* bIsWritten	Hresult

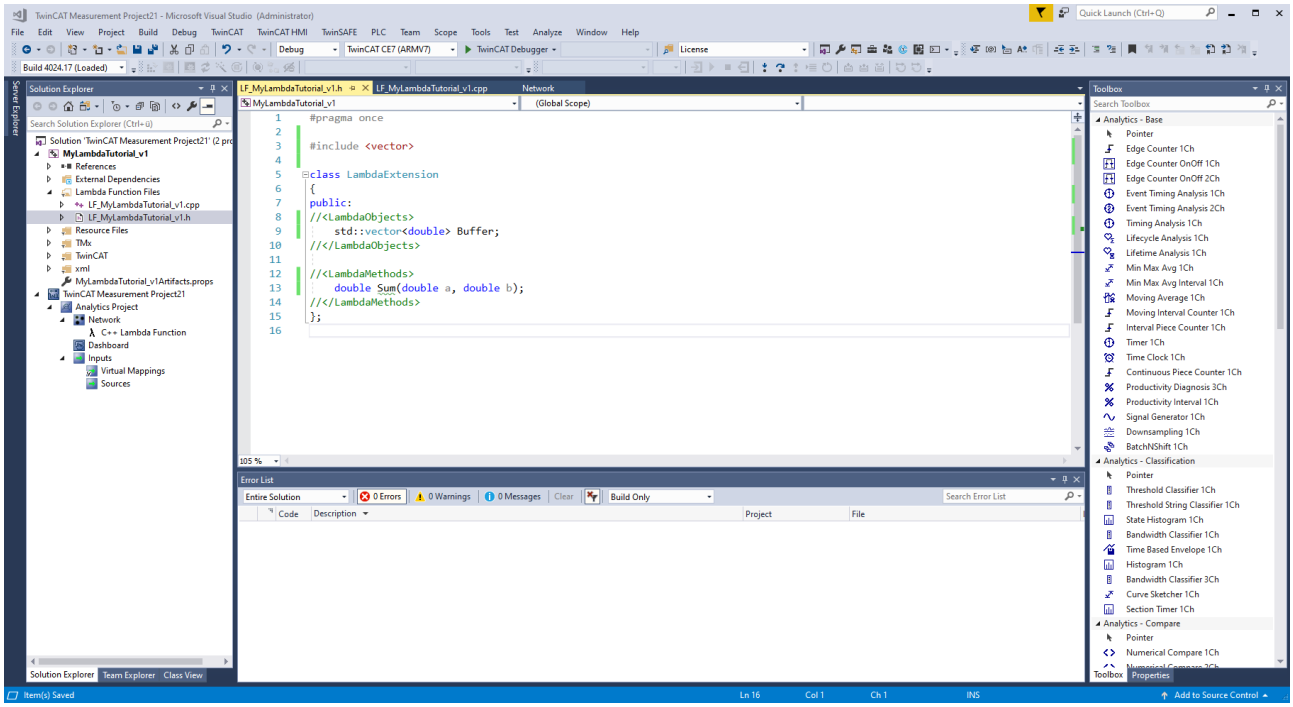
In der TwinCAT Analytics Workbench ist der Dateizugriff synchron, wohingegen er in der Analytics Runtime asynchron ist. Diese Besonderheit wird in den Methoden automatisch unterschieden. Bei einem asynchronen Lesezugriff liefert die Methode nicht im selben Zyklus den gelesenen Wert. Erst sobald der Parameter bNewResult den Wert true zurückliefert, enthält der Parameter value den gelesenen Wert. Bei dem asynchronen Schreibzugriff werden die Daten intern gespeichert und bei der nächsten Gelegenheit geschrieben. Sobald die Daten geschrieben wurden, liefert der Parameter bIsWritten den Wert true zurück.

Die beiden Parameter bNewResult und bIsWritten können Sie bei einem synchronen Dateizugriff analog verwenden. Die folgende Abbildung zeigt eine beispielhafte Verwendung des Dateizugriffes in den Lambda Funktionen:

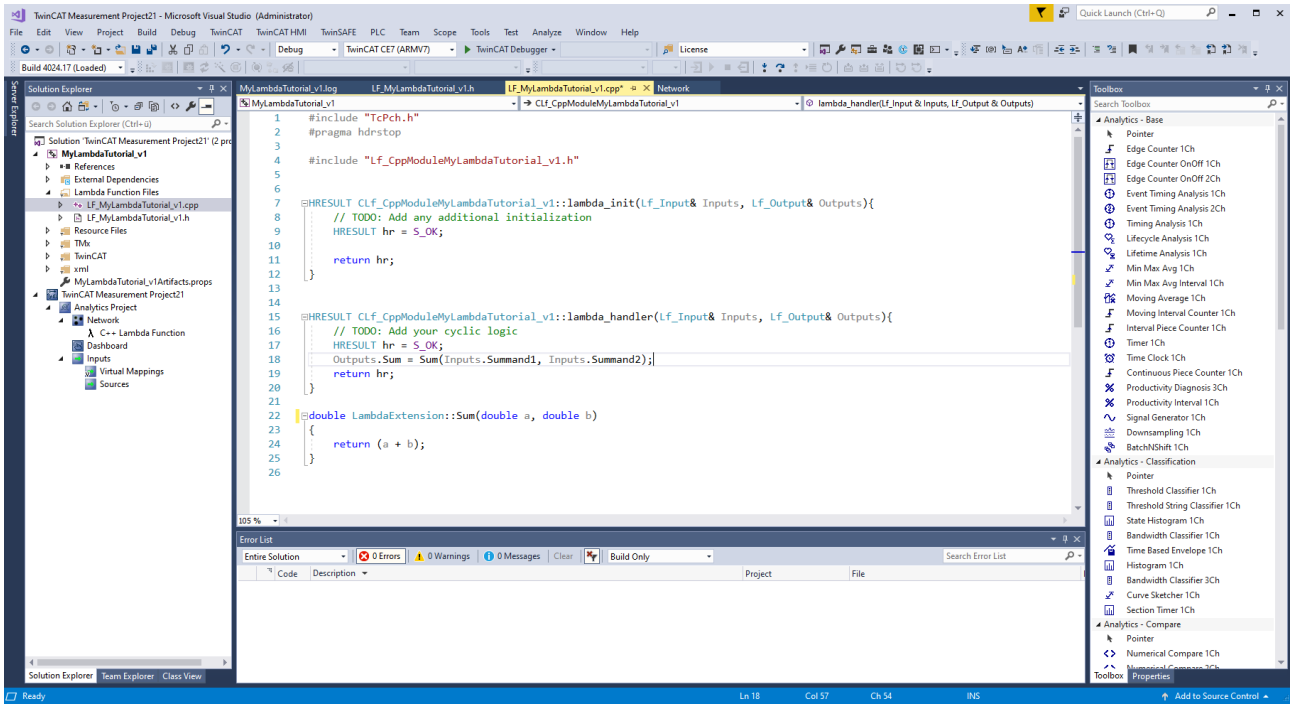


Erstellen Sie das Projekt wie gewohnt in Visual Studio®, um die Syntax zu prüfen.

In der Header-Datei in dem Ordner „Lambda Function Files“ können Sie anwenderspezifische Objekte und Methoden deklarieren.



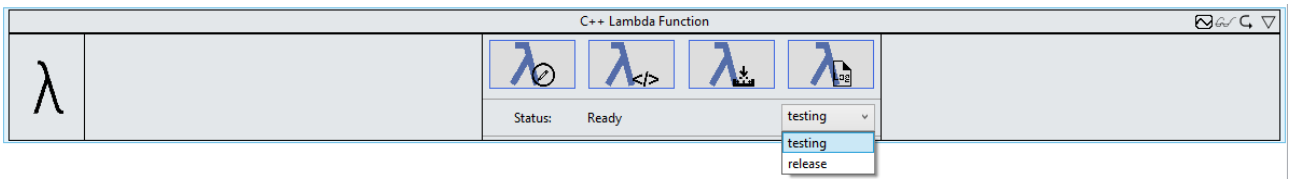
Oberhalb der Klasse „LambdaExtension“ können Bibliotheken inkludiert werden. Innerhalb der Klasse können Objekte und Methoden deklariert und in den Lambda Funktionen verwendet werden. Die neu deklarierten Methoden können unterhalb der beiden Lambda Funktionen implementiert werden.



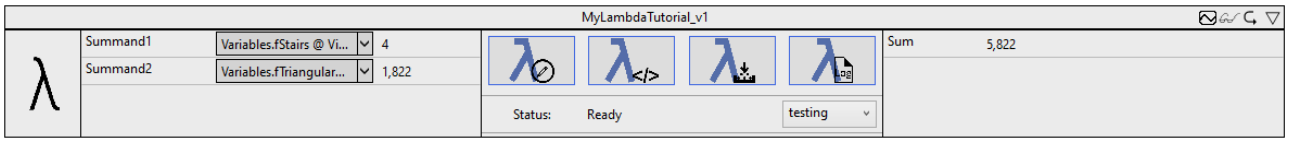
Nachdem die Logik integriert wurde, müssen Sie das Projekt speichern. Wechseln Sie danach zum Lambda Template im Analytics Projekt zurück.
Falls ausreichend Know-how im Umgang mit TcCOM-Modulen vorhanden ist, kann das C++-Projekt individuell erweitert werden. Bei unsachgemäßer Verwendung der TcCOM-Module, können bei nachfolgenden Schritten der Lambda Funktionen Fehler auftreten. Weitere Informationen zu der Entwicklung von TcCOM-Modulen sind unter dem folgenden Link zu finden TwinCAT C++ Entwicklung.

Publishen der Lambda Funktion

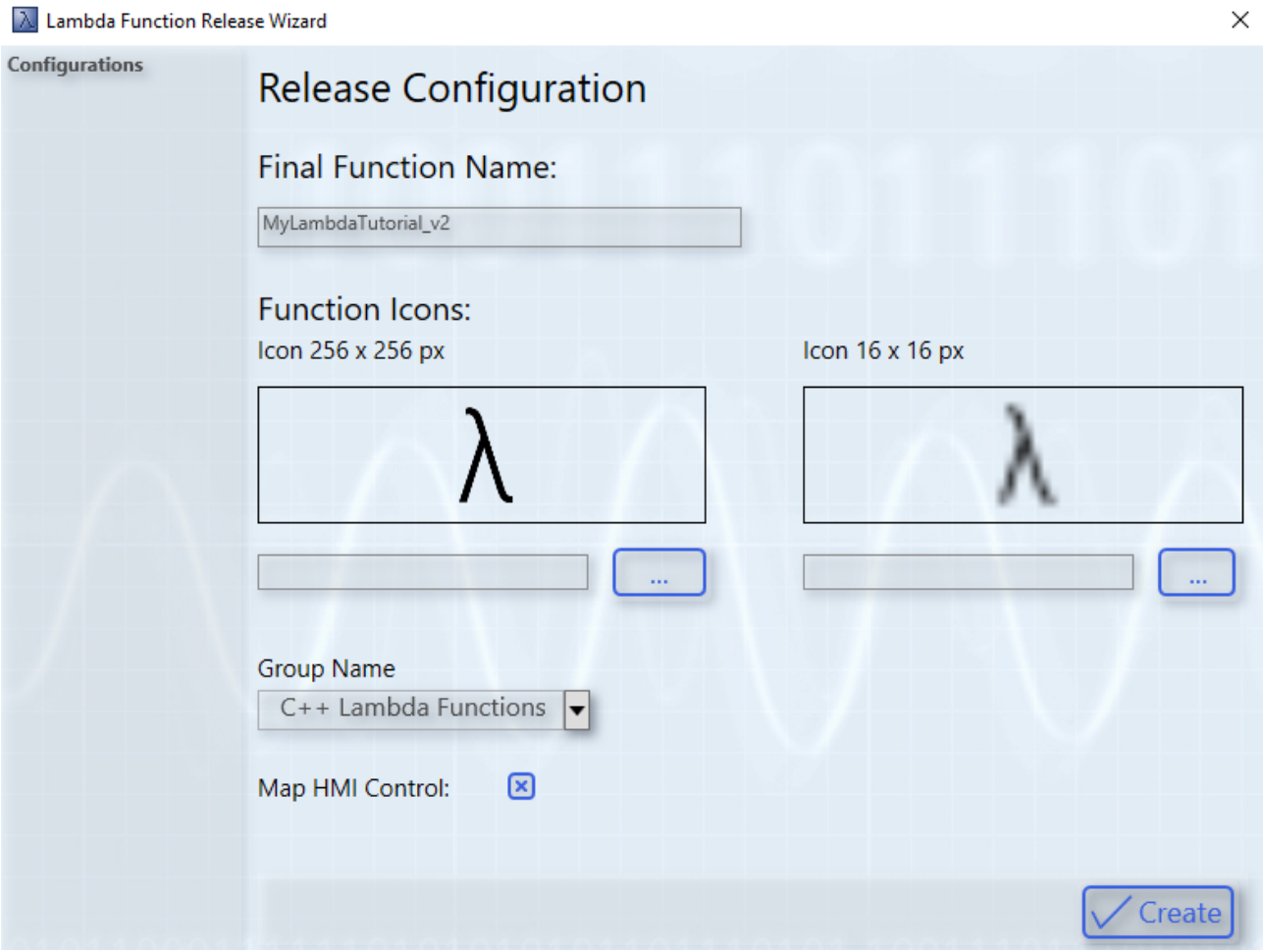
Vor dem Publishen können Sie den Typ der Lambda Funktion über das Drop-Down-Menü unten rechts konfigurieren.



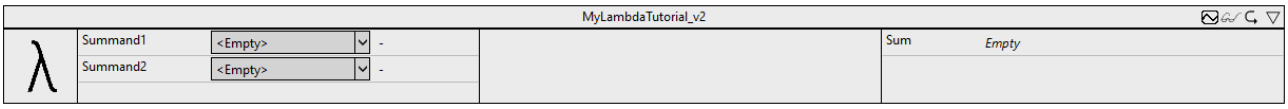
Eine Lambda-Funktion von Typ „testing“ enthält das Control mit den vier Buttons zum Entwickeln von Lambda Funktionen. Dieser Typ bietet sich während der Entwicklung- und Testphase an.



Sobald die Lambda Funktion funktioniert, wählen Sie den Typ „release“ aus. Nach dem Klick auf den dritten Button öffnet sich ein Release Wizard.



Der Wizard bietet die Möglichkeit, einen finalen Namen zu vergeben und individuelle Icons auszuwählen. Es muss jeweils ein Icon mit 256 x 256 Pixel und ein Icon mit 16 x 16 Pixel ausgewählt werden. Alternativ können Sie die Lambda-Icons weiterverwenden. Mit dem Drop-Down-Menü ordnen Sie die Lambda Funktion einer bestimmten Gruppe in der Toolbox zu. Standardmäßig ist die Gruppe „C++ Lambda Functions“ ausgewählt. So kann eine final Lambda Funktion beispielsweise in der Toolbox in den Bereich "Analytics - Base" eingeordnet werden. Wenn die Checkbox "Map HMI Control" aktiviert ist, öffnet sich nach dem Publishen ein Wizard zum Verknüpfen von HMI Controls mit der Lambda Funktion. Beim Analytics HMI Control Manager werden die Punkte zur Modulauswahl direkt mit der entsprechenden Lambda Funktion vorausgewählt (siehe auch Angepasste und eigene Controls verwenden [▶ 446]). Über den Create-Button generieren Sie die Lambda Funktion.



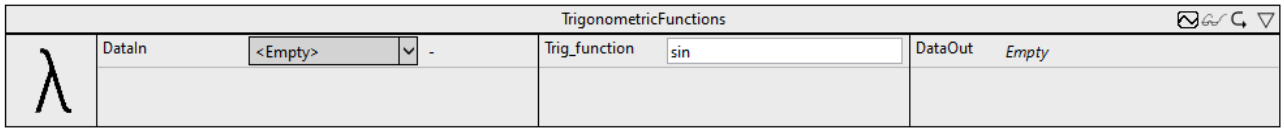
Nach dem erfolgreichen Publishen wird das Lambda Template automatisch durch die neu generierte Lambda Funktion ersetzt.

Einsehen der Log-Dateien

Sollte beim Publishen der Lambda Funktion ein Fehler auftreten, wird es in der Statusanzeige dargestellt und die Log-Datei kann über den vierten Button eingesehen werden.

6.4.14.1.4 Lambda Blueprints

Trigonometric functions



Der Blueprint **Triginometric function** führt die Berechnung einer trigonometrischen Funktion an einem Eingangskanal aus und liefert das Ergebnis an den Ausgangskanal. Der Eingangskanal erwartet den Winkel im Bogenmaß. Die jeweilige trigonometrische Funktion kann in dem Parameterbereich angegeben werden.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

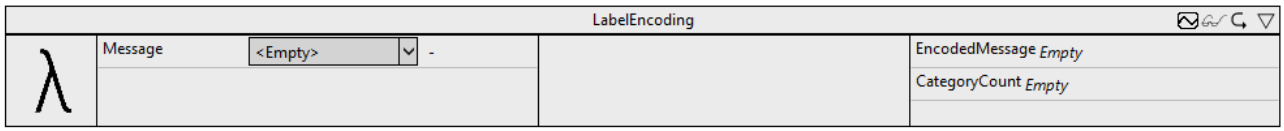
Konfigurationsoptionen

- **Trig_function**: Die Kurzform der zu berechnenden trigomischen Funktion,.z. B. sin für Sinus. Zu den Auswahlmöglichkeiten gehören: sin, cos, tan, asin, acos und atan. Der Eingangswert für asin und acos muss in dem Intervall [-1,1] liegen.

Ausgangswerte

- **DataOut**: Gibt das Ergebnis der Berechnung der trigonometrischen Funktion aus.

Label Encoding



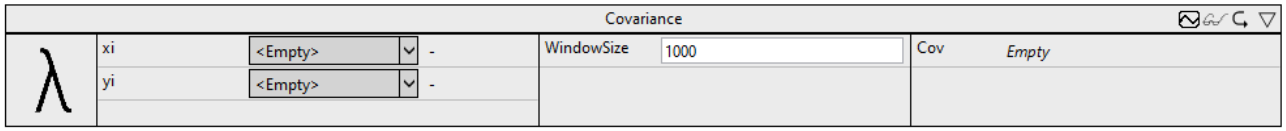
Der Blueprint **Label Encoding** wandelt Text an einem Eingangskanal in numerische Werte um. Dazu werden die Texte am Eingangskanal in unterschiedliche Kategorien eingeordnet. Gleiche Eingangstexte liefern denselben numerischen Wert zurück.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Ausgangswerte

- **EncodedMessage**: Dieser Ausgangskanal liefert den Index der entsprechenden Kategorie zurück.
- **CategoryCount**: Dieser Ausgangskanal gibt die Anzahl der Kategorien aus.

Covariance



Der Blueprint **Covariance** berechnet die Kovarianz zwischen zwei Eingangskanälen. Die WindowSize gibt dabei an, wie viele Werte in die Berechnung der Kovarianz eingehen.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

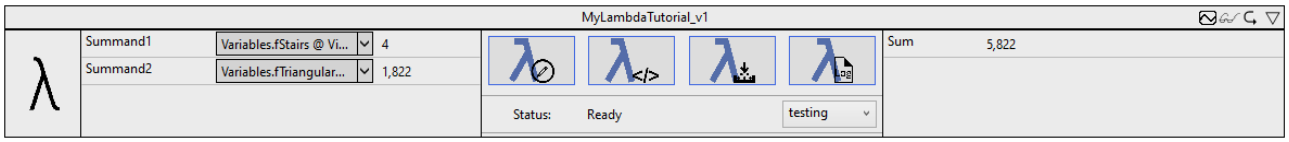
- **WindowSize**: Dieser Parameter gibt die Anzahl der Zyklen an, über die die Kovarianz berechnet wird. In dem Blueprint ist ein Sliding Window implementiert. Dabei fließen die aktuellen Werte in die Analyse ein und es werden mit jedem Zyklus die Ausgänge aktualisiert. Die Größe des Routerspeichers sollte beim Setzen dieses Parameters berücksichtigt werden.

Ausgangswerte

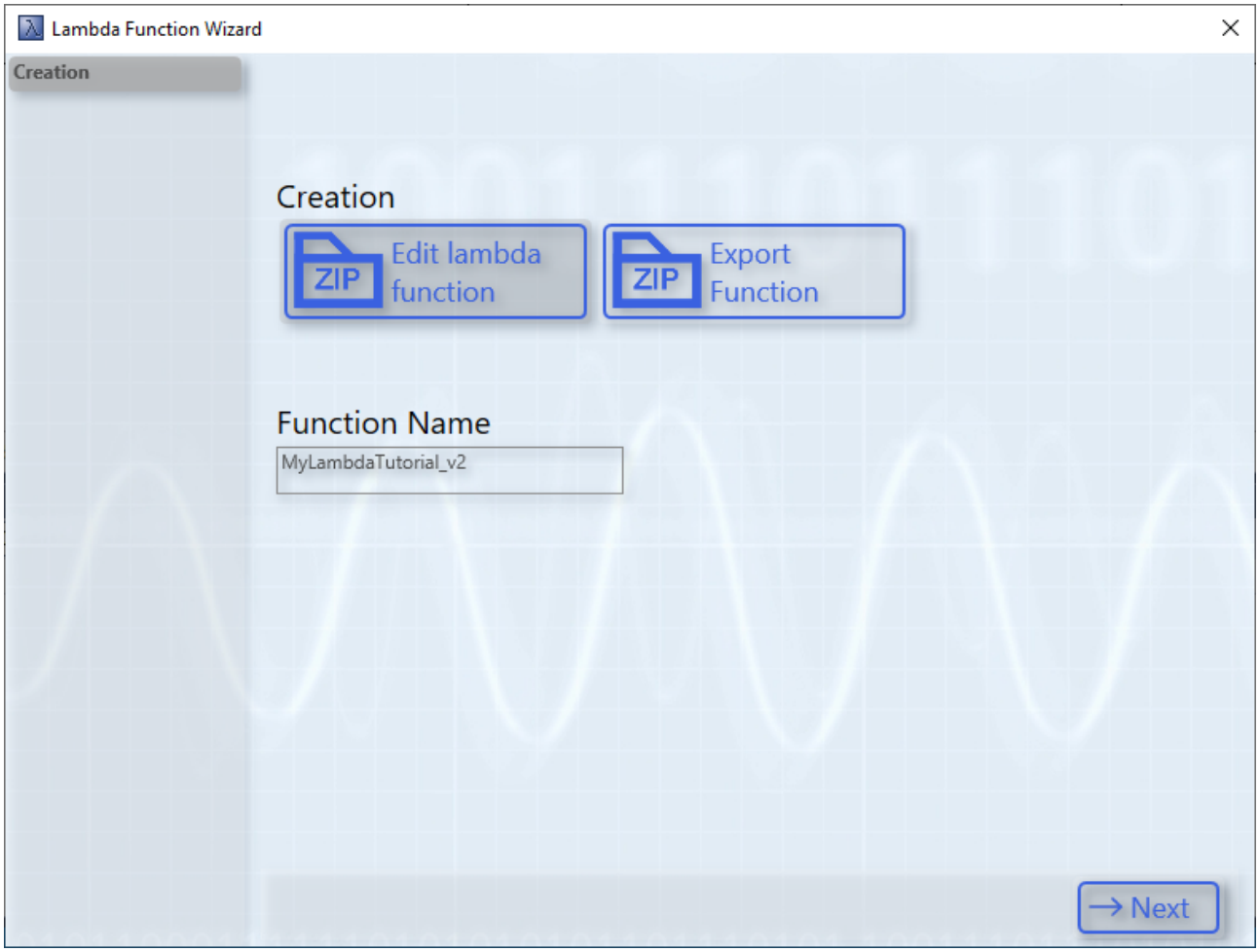
- **Cov**: Dieser Ausgangskanal gibt die Kovarianz der beiden Eingangskanäle über die konfigurierte Fenstergröße an.

6.4.14.1.5 Lambda Function

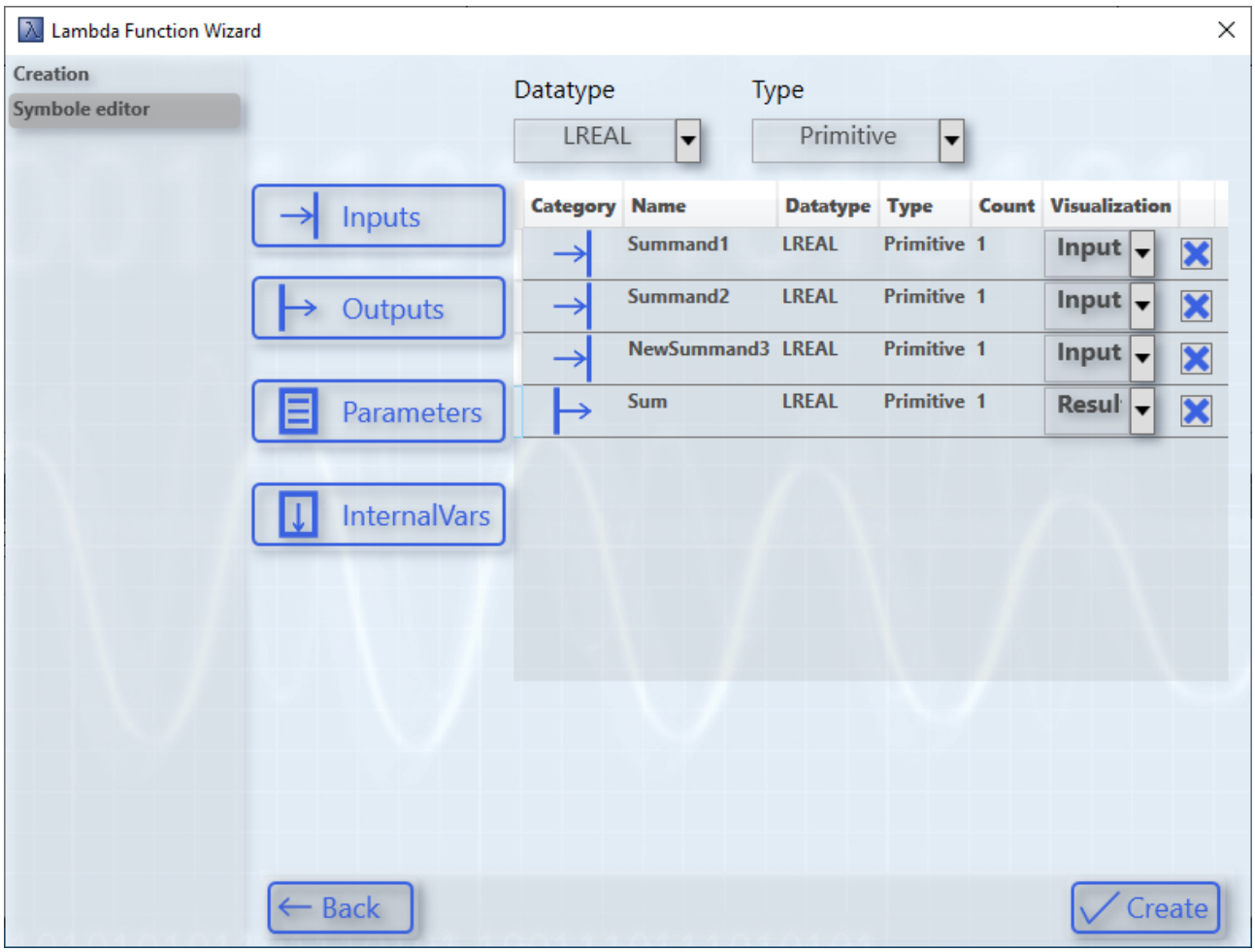
In der Entwicklungs- und Testphase sollten Lambda Funktionen als Testing-Versionen gepublisht werden. Diese enthalten die konfigurierten Symbole, sowie das Lambda-Control mit den vier Buttons.



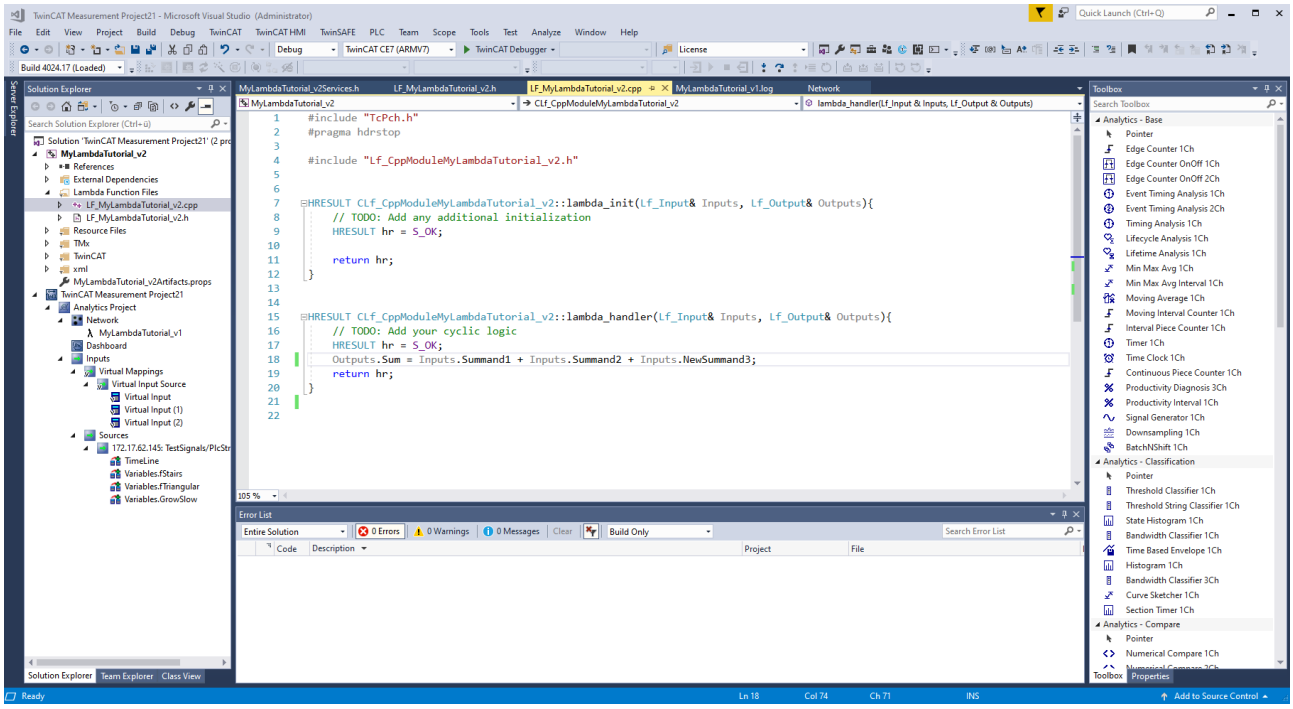
Diese Lambda Funktion kann wie jeder andere Algorithmus in der Analytics Workbench getestet werden. Sollten Änderungen von Nöten sein, leitet das Lambda-Control durch die Weiterentwicklung. Nach dem Klicken des ersten Buttons öffnet sich ein Wizard zum Konfigurieren der Lambda Funktion. Der Wizard für bestehende Lambda Funktionen bietet zwei Modi.



Der Button **Export Function** bietet die Möglichkeit Lambda Funktionen zu exportieren. Dazu wird das C++-Projekt und die Treiber der Lambda Funktion in einem Zip-Archiv gespeichert. Über die Import-Funktion im Lambda Template kann die Lambda Funktion auf einem anderen System komfortabel importiert und integriert werden. Über den Modus **Edit lambda function** kann die bereits entwickelte Lambda Funktion bearbeitet werden. Über den zweiten Button kann die aktuelle Lambda Funktion exportiert werden. Dafür öffnet sich ein Datei-Speichern-Dialog. Über diesen Dialog kann der Dateispeicherort festgelegt werden. Die Exportierung startet nach der Auswahl. Die exportierte Lambda Funktion kann anschließend auf andere Systeme verteilt werden. Die anderen Systeme können die Lambda Funktion über den Wizard vom Lambda-Template importieren.
Beim Modus **Edit lambda function** können die Symbole auf der nächsten Seite des Wizards angepasst werden.



In diesem Beispiel wurde ein weiterer Eingang mit dem Namen „NewSummand3“ eingefügt. Diese Änderungen sind in dem darauffolgenden Schritt auch in der C++-Datei enthalten.



Die neue Variable ist in der Struktur aus Eingängen enthalten und kann im Programmcode verwendet werden. Der Programmcode aus der vorherigen Version wurde übernommen. Sollten Symbole über eine Version hinweg entfernt worden sein, müssen die entsprechenden Änderungen im Programmcode

vorgenommen werden.
Die weiteren Schritte in der Entwicklung der Lambda Funktion sind analog zu dem Ablauf beim Lambda-Template.

Debuggen von Lambda Funktionen

Um Lambda Funktionen zu debuggen, ist auf die Tools von TwinCAT C++ zurückzugreifen. Dazu muss ein neues TwinCAT-Projekt erstellt werden und das C++-Projekt der Lambda Funktion unter dem Knoten C++ eingefügt werden. Dabei stehen alle Möglichkeiten des TwinCAT C++ Debuggings zur Verfügung (siehe Debuggen). Über den Reiter Debug kann sich an den Prozess des TcScopeSever angehangen werden. Anschließend ist die Analyse in der Workbench zu starten und die gesetzten Haltepunkte werden in dem C++-Projekt erreicht.

Aufräumen der Toolbox

Nachdem einige Lambda Funktionen entwickelt und getestet wurden, kann die Toolbox möglicherweise sehr voll sein. Um die Toolbox aufzuräumen, müssen die entsprechenden Lambda Funktionen aus dem Modules-Ordner gelöscht werden. Der Ordner ist unter dem Pfad *<TwinCAT3Dir>/CustomConfig/Modules* zu finden.

6.4.14.1.6 Weitere Schritte für die Lambda Funktionen

Runtime-Deployment

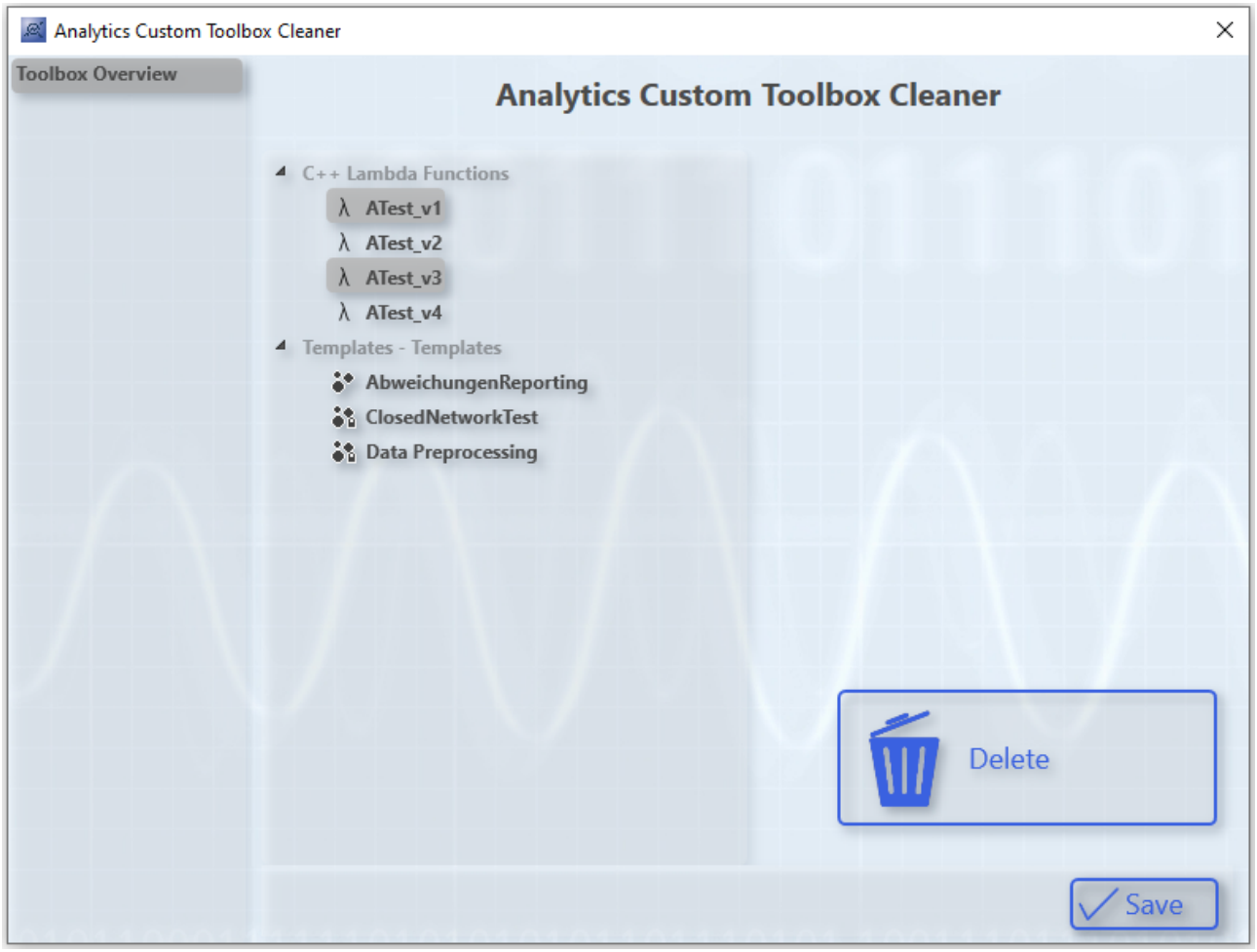
Auch die Lambda Funktionen sind für das Runtime-Deployment geeignet. Die entwickelte Lambda Funktion wird dafür als TcCOM-Modul in dem TwinCAT-Projekt angelegt. Ein SPS-Baustein übernimmt die Handhabung des TcCOM-Moduls. Der SPS-Baustein ruft dafür die entsprechenden Schnittstellen-Methoden der Lambda Funktion auf. Die Schnittstellen-Methoden der Lambda Funktion, sowie der dazugehörige SPS-Baustein werden automatisch generiert und benötigen keine explizite Konfiguration. Somit können die anwenderspezifischen Lambda Funktionen in eine TwinCAT Analytics Runtime heruntergeladen werden und eine individuelle 24/7-Datenanalyse realisiert werden. Weitere Informationen sind unter dem folgenden Link zu finden (siehe Runtime-Deployment [▶ 365]).

HMI One Click Dashboard

Um die Ergebnisse der Lambda Funktionen darzustellen, bietet sich die Integration in das HMI One Click Dashboard an. Über den Dashboard-Knoten eines Analytics Projektes lässt sich ein neues Control verknüpfen. In dem Analytics Dashboard Wizard kann ein HMI Control und die entsprechende Lambda Funktion ausgewählt und verknüpft werden. Der Analytics Dashboard Wizard ist unter dem folgenden Link genauer beschrieben (siehe Dashboard Aufbau und Inhalte im Analytics Projekt verwalten [▶ 436]).

6.4.14.1.7 Analytics Custom Toolbox Cleaner

Durch die Netzwerk-Templates und die Lambda Funktionen können eine Vielzahl von Toolbox-Elementen erstellt werden. Um diese Elemente komfortabel zu entfernen, kann der Analytics Custom Toolbox Cleaner verwendet werden. Die folgende Abbildung zeigt den Wizard:



Über die Baumstruktur können die einzelnen Toolbox-Elemente ausgewählt werden. Durch das gedrückt halten von STRG können mehrere Elemente in der Baustruktur selektiert werden. Der Button Delete entfernt die Elemente aus der Baustruktur. Durch einen Klick auf den Button Save werden die Änderungen übernommen. Sollte der Wizard anderweitig geschlossen werden, werden die Änderungen nicht übernommen.

6.4.14.2 Algorithmen aus weiteren TwinCAT Bibliotheken

Durch die Einbindung der TwinCAT Filter- und TwinCAT Condition Monitoring- Bibliotheken steht eine weitere Vielzahl an Algorithmen zur Verfügung. Durch die Integration dieser Algorithmen in TwinCAT Analytics werden Programmieraufgaben zu reinen Konfigurationsaufgaben. Das reduziert den Aufwand für die Erstellung einer Analyse und ermöglicht zudem die gemeinsame Benutzung dieser Algorithmen mit denen der TwinCAT Analytics Library direkt im TwinCAT Analytics Workbench-Konfigurator beziehungsweise Service Tool. Sie befinden sich analog zu den anderen Gruppen von Algorithmen in der Toolbox. Die zusätzlichen Algorithmen können für das Engineering ohne weitere Lizenz verwendet werden. Für das Deployment der Algorithmen in die TwinCAT Analytics Runtime sowie die Generierung eines HMI One Click Dashboards werden allerdings zusätzlich die Lizenzen der jeweiligen Bibliothek benötigt. Die folgende Liste zeigt eine Übersicht, über die integrierten Bibliotheken und Algorithmen, die Lizenz, die jeweils für das Deployment oder die Dashboardgenerierung benötigt wird, sowie einen Link zur Dokumentation des jeweiligen SPS-Funktionsbausteins.

Bibliothek	Algorithmen	Lizenz	Dokumentation
TwinCAT Filter	*Filter*	TF3680 TC3 Filter / TF3600 TC3 Condition Monitoring	
	PT1		PT1 [▶ 328]
	IIRCoeff		IIRCoeff [▶ 329]
	IIRSpec		IIRSpec [▶ 329]
	PT2		PT2 [▶ 330]
	PT3		PT3 [▶ 330]
	PTn		PTn [▶ 331]
	IIRSos		IIRSos [▶ 331]
	Notch		Notch [▶ 331]
	LeadLag		LeadLag [▶ 332]
	PT2oscillation		PT2oscillation [▶ 332]
	PTt		PTt [▶ 333]
	Median		Median [▶ 333]
	ActualValue		ActualValue [▶ 333]
	Gaussian		Gaussian [▶ 334]
TwinCAT Condition Monitoring	*Condition Monitoring*	TF3600 TC3 Condition Monitoring	
	ArgSort		ArgSort [▶ 313]
	Crest Factor		Crest Factor [▶ 314]
	Crest Factor Plus		Crest Factor Plus [▶ 314]
	DiscreteClassification		Discrete Classification [▶ 315]
	Envelope		Envelope [▶ 315]
	Envelope Spectrum		Envelope Spectrum [▶ 316]
	Fatigue Analysis		Fatigue Analysis [▶ 317]
	Instantaneous Frequency		Instantaneous Frequency [▶ 319]
	Instantaneous Phase		Instantaneous Phase [▶ 313]
	Integrated RMS		Integrated RMS [▶ 319]
	Magnitude Spectrum		Magnitude Spectrum [▶ 321]
	Multi-Band RMS		Multi Band RMS [▶ 322]
	Power Spectrum		Power Spectrum [▶ 323]
	RMS		RMS [▶ 324]
	Spike Energy Spectrum		Spike Energy Spectrum [▶ 324]
	Vibration Assessment		Vibration Assessment [▶ 326]
	Watch Upper Thresholds		Watch Upper Thresholds [▶ 327]

Die Bibliothek **TwinCAT Filter** ermöglicht die Realisierung von digitalen Filtern unterschiedlicher Filtertypen mithilfe der verschiedenen Funktionsbausteine.

Die Bibliothek **TwinCAT Condition Monitoring** bietet mathematische Algorithmen zur Zustandsüberwachung von Maschinen und Anlagen.

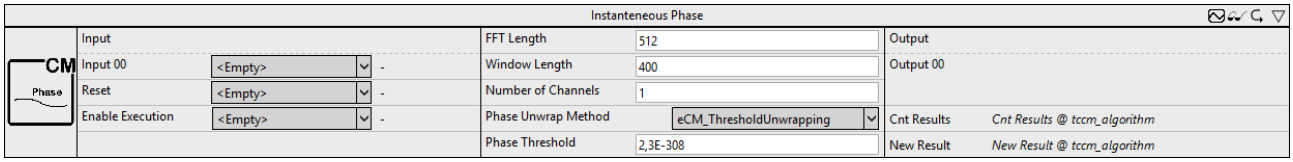
6.4.14.2.1 Condition Monitoring

Die Bibliothek **TwinCAT Condition Monitoring** bietet mathematische Algorithmen zur Zustandsüberwachung von Maschinen und Anlagen.

Übersicht der Algorithmen

Algorithmus	Beschreibung	SPS Funktionsbaustein
ArgSort [▶ 313]	Sortiert die eingehenden Argumente.	FB_CMA_ArgSort
Crest Factor [▶ 314]	Berechnet für ein- und mehrkanalige Zeitreihen den Crest-Faktor für jeden Kanal.	FB_CMA_CrestFactor
Crest Factor Plus [▶ 314]	Berechnet für ein- und mehrkanalige Zeitreihen den Crest-Faktor Plus für jeden Kanal.	FB_CMA_CrestFactorPlus
Discrete Classification [▶ 315]	Klassifikation von mehrkanaligen Daten anhand konfigurierbarer Schwellwerte.	FB_CMA_DiscreteClassification
Envelope [▶ 315]	Berechnet die Einhüllende eines Zeitsignals.	FB_CMA_Envelope
Envelope Spectrum [▶ 316]	Berechnet das Einhüllendenspektrum eines Zeitsignals.	FB_CMA_EnvelopeSpectrum
Fatigue Analysis [▶ 317]	Lebensdaueranalyse und Schädigungsrechnung.	FB_CMA_RainflowCounting FB_CMA_MeanStressCorrection FB_CMA_MinersRule
Instantaneous Frequency [▶ 319]	Berechnung der Momentanfrequenz eines Zeitsignals.	FB_CMA_InstantaneousFrequency
Instantaneous Phase [▶ 313]	Berechnung der Momentanphase eines Zeitsignals.	FB_CMA_InstantaneousPhase
Integrated RMS [▶ 319]	Berechnet (optional integrierte) RMS Werte für ein- und mehrkanalige reell-wertige Zeitreihen.	FB_CMA_IntegratedRMS
Magnitude Spectrum [▶ 321]	Berechnet das Magnitudenspektrum (auch Betragsspektrum oder Amplitudenspektrum genannt) eines reell-wertigen Eingangssignals.	FB_CMA_MagnitudeSpectrum
Multi Band RMS [▶ 322]	Berechnet RMS Werte für ein- und mehrkanalige reell-wertige Zeitreihen für konfigurierbarer Frequenzbänder.	FB_CMA_MultiBandRMS
Power Spectrum [▶ 323]	Berechnung des Leistungsspektrums eines reell-wertigen Eingangssignals sowie optionale Skalierung zu Dezibel.	FB_CMA_PowerSpectrum
RMS [▶ 324]	Berechnet für ein- und mehrkanalige reell-wertige Signale den zeitlichen RMS Wert.	FB_CMA_RMS
Spike Energy Spectrum [▶ 324]	Analyse von Spitzenenergie hochfrequenter Signalanteile.	FB_CMA_SpikeEnergySpectrum
Vibration Assessment [▶ 326]	Schwingungsbeurteilung reell-wertiger Eingangssignale.	FB_CMA_VibrationAssessment
Watch Upper Thresholds [▶ 327]	Konfigurierbare Schwellwertüberwachung von mehrkanaligen Daten.	FB_CMA_WatchUpperThresholds

6.4.14.2.1.1 Instantaneous Phase



Berechnung der Momentanphase eines Zeitsignals.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_InstantaneousPhase

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

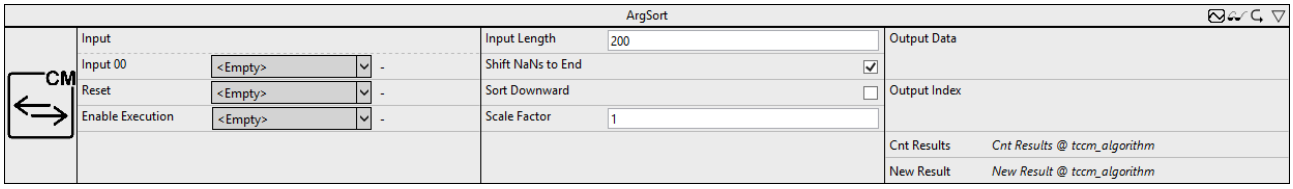
Konfigurationsoptionen

- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.

Phase Unwrap Method: Definiert die verwendete Methode zum phase-unwrapping bezüglich der Phase in Vielfachen von 2 PI (siehe E_CM_UnwrapMethod).

- **Phase Threshold**: Grenzwert zur Berechnung der Momentanphase. Der Wert bezieht sich auf die Einhüllende des Signals. Interpretation: Bei zu geringem Signalpegel ist die Berechnung der Phase numerisch zu unsicher und nicht verlässlich auswertbar. Als Phase wird dann eine 0 ausgegeben.

6.4.14.2.1.2 ArgSort



Sortiert die eingehenden Argumente.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_ArgSort

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

Konfigurationsoptionen

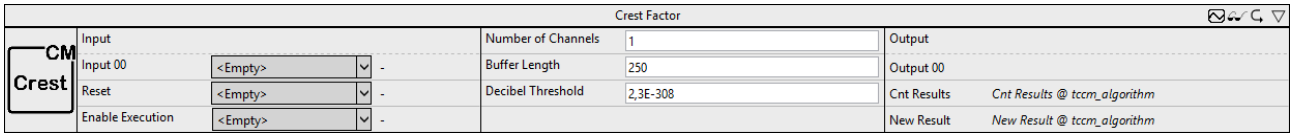
- **Input Length**: Ist die Länge des Eingangs-Arrays.
- **Shift NaNs to End**: Ist ein Flag, mit dem gewählt werden kann, ob NaNs an das Ende des Ausgangs-Arrays geschoben werden.
- **Sort Downward**: Ist ein Flag, mit dem gewählt werden kann, ob die Daten aufsteigend (`FALSE`) oder absteigend (`TRUE`) sortiert werden sollen.
- **Scale Factor**: Dieser Parameter kann genutzt werden, um sich anstelle der Index-Position (Scale Factor = 1) beispielsweise direkt die Amplituden mit zugehörigen Frequenzen anzeigen zu lassen.

Ausgangswerte

- **Output Data**: Ausgangsarray mit identischer Länge des Eingangsarrays, welches die wahlweise aufsteigend oder absteigend sortierten Eingangswerte enthält.
- **Output Index**: Ausgangsarray mit identischer Länge des Eingangsarrays, welches die zu den sortierten Werten gehörenden (skalierbaren) Indizes im Eingangsarray beinhaltet.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.

- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.3 Crest Factor



Berechnet für ein- und mehrkanalige Zeitreihen den Crest-Faktor für jeden Kanal.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_CrestFactor

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

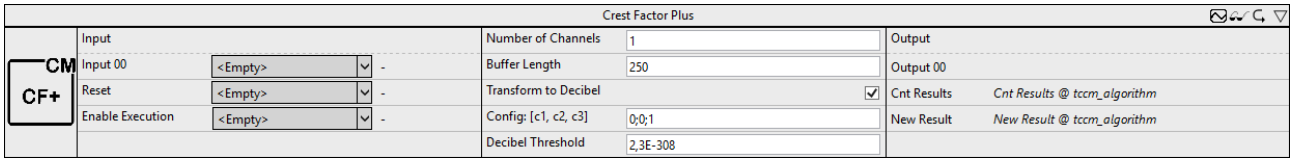
Konfigurationsoptionen

- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Buffer Length**: Ist die Zahl der im internen Puffer gehaltenen Eingangswerte je Kanal.
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)

Ausgangswerte

- **Output**: Crest Faktor des zugehörigen Eingangsdaten-Streams.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.4 Crest Factor Plus



Berechnet für ein- und mehrkanalige Zeitreihen den Crest-Faktor für jeden Kanal.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_CrestFactorPlus

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

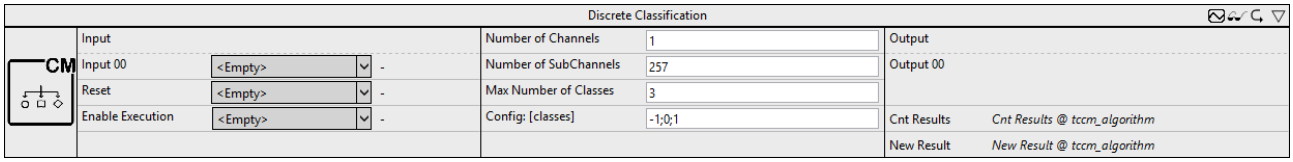
Konfigurationsoptionen

- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Buffer Length**: Ist die Zahl der im internen Puffer gehaltenen Eingangswerte je Kanal.
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Config: [c1, c2, c3]:** Konfigurierbare Gewichte der maximalen Amplitude (c1), dem RMS Wert (c2) und des Crest Faktors (c3).
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)

Ausgangswerte

- **Output**: Crest Faktor Plus des zugehörigen Eingangsdaten-Streams bezüglich der konfigurierbaren Gewichte.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.5 Discrete Classification



Klassifikation von mehrkanaligen Daten anhand konfigurierbarer Schwellwerte.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_CMA_DiscreteClassification

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

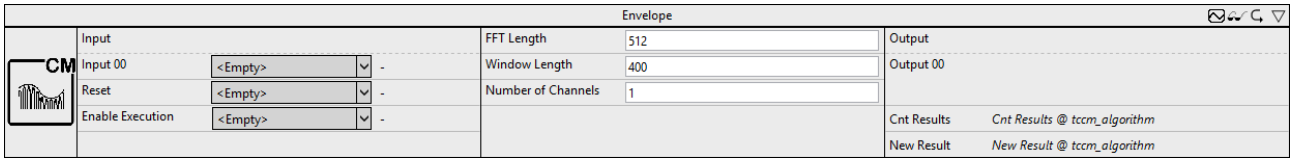
Konfigurationsoptionen

- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Number of SubChannels**: Definiert die Anzahl der unabhängigen Unterkanäle. Dieser Wert wird automatisch der Länge des verknüpften Eingangsarrays angepasst. Falls mehrere Kanäle genutzt werden, müssen alle Längen übereinstimmen.
- **Max Number of Classes**: Definiert die maximale Anzahl an Klassen, die konfiguriert werden. Dieser Wert muss mindestens eins sein.
- **Config: [classes]**: Definition der konfigurierbaren Schwellwerte.

Ausgangswerte

- **Output**: Ausgangsarray mit identischer Länge des Eingangsarrays, welches die Klassifikationen der Einträge enthält.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.6 Envelope



Berechnet die Einhüllende eines Zeitsignals.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_CMA_Envelope

Für die zeitsynchrone Darstellung und Weiterverarbeitung beachte den Abschnitt Timeshift [▶ 316].

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

Konfigurationsoptionen

- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.

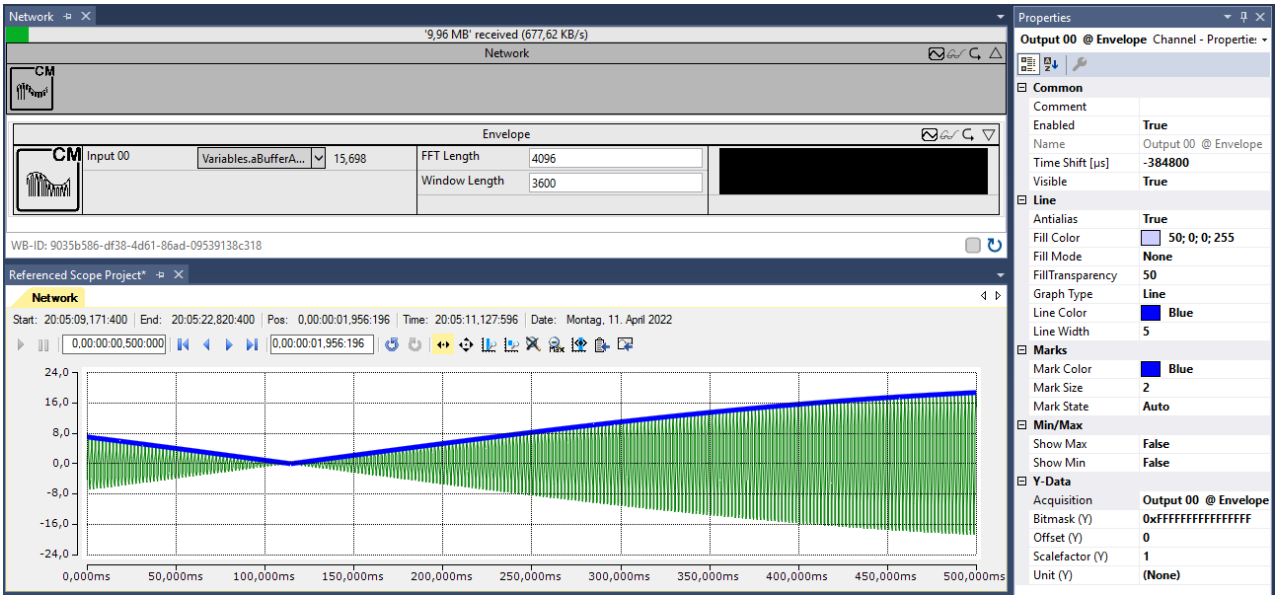
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.

Ausgangswerte

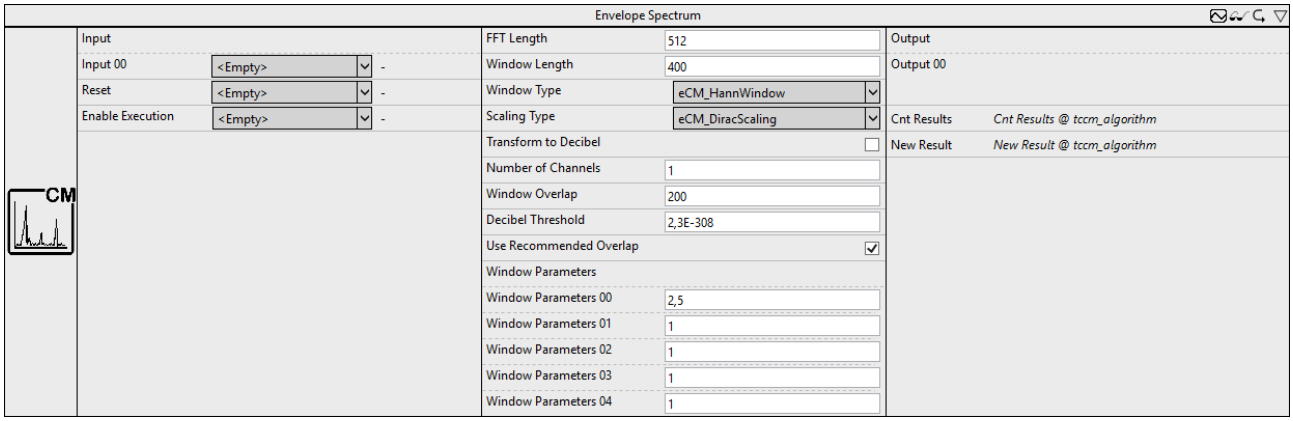
- **Output**: Ausgangsarray mit halber Fensterlänge, welches die berechnete Einhüllende enthält.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

Time shift

Aufgrund der Verwendung der Welch Methode (50% Überlappung) für die Analyse sowie einer segmentierten Faltung des analytischen Signals in der Synthese werden vier vollständige Datenpuffer der Länge `nWindowLength/2` benötigt um ein valides Ausgangssignal zu berechnen. Ferner ist das Ergegnis um `(nFFTLentgth + nWindowLength)/2` Samples verzögert. Dies lässt sich mittels geeigneter Wahl des Time Shift am Kanal des Outputs korrigieren.



6.4.14.2.1.7 Envelope Spectrum



Berechnet das Einhüllendenspektrum eines Zeitsignals.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_EnvelopeSpectrum

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

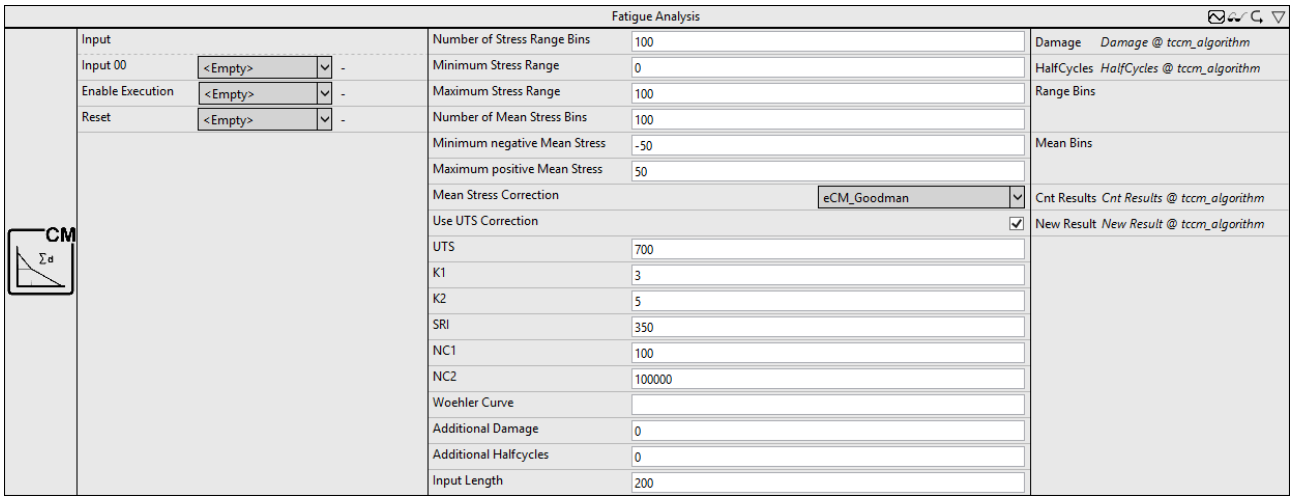
Konfigurationsoptionen

- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Window Type**: Definiert die verwendete Fensterfunktion (vom Typ E_CM_WindowType). Ein guter Standardwert ist der Fenstertyp `eCM_HannWindow`.
- **Scaling Type**: ermöglicht eine Auswahl der verwendeten Skalierung (vom Typ E_CM_ScalingType), falls eine absolute Skalierung benötigt wird. Standardwert ist `eCM_DiracScaling`. Bei der Auswahl der Skalierung sollte die Art des Signals, entweder deterministische Signale oder breitbandige Signale mit stochastischen Anteil, berücksichtigt werden; Beide Arten erfordern unterschiedliche Skalierungen.
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Window Overlap**: Definiert die Anzahl der überlappenden Samples. Diese muss größer oder gleich null sein.
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)
- **Use Recommended Overlap**: Falls selektiert, wird eine empfohlene Überlappung intern berechnet (siehe F_CM_CalculateRecommendedOverlap).
- **Window Parameters**: Beinhaltet die freien Parameter ausgewählter Fensterfunktionen. Bei der Verwendung von `eCM_KaiserWindow` definiert der erste Eintrag den Parameter beta, wird das `eCM_FlatTopWindow` verwendet, werden alle Parameter genutzt. Siehe Abschnitt Fensterfunktionen.

Ausgangswerte

- **Output**: Ausgangsarray der Länge `N / 2 + 1` (bei FFT Länge `N`), welches die berechneten Spektrallinien des Einhüllendenspektrums enthält.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.8 Fatigue Analysis



Lebensdaueranalyse und Schädigungsrechnung zur Abschätzung des Ermüdungsprozesses von überwachten Bauteilen.

Die Dokumentationen der zugehörigen SPS-Funktionsbausteine finden Sie hier: FB_CMA_RainflowCounting, FB_CMA_MeanStressCorrection und FB_CMA_MinersRule

Für die Berechnung der Wöhlerkurve wird die Funktion F_CM_CalculateWoehlerCurve auf Basis von fiktiven Materialparametern durchgeführt.

Nähere Erläuterungen finden Sie in der Dokumentation des Condition Monitoring im Kapitel Anwendungskonzepte, Abschnitt Lebensdaueranalyse und Schädigungsrechnung.

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

Konfigurationsoptionen

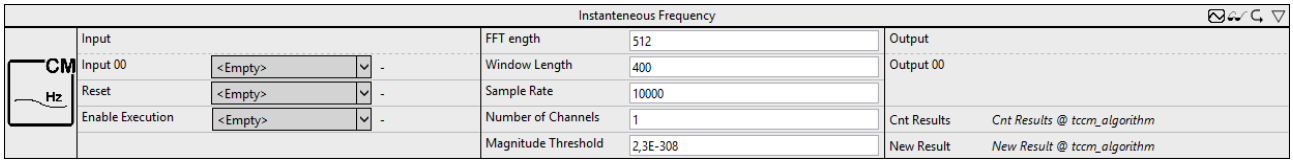
- **Number of Stress Range Bins**: Definiert die Anzahl der Bins für das Stresslevel. Die Gesamtzahl der Bins ist „Bins + 2“, der Wert muss mindestens eins sein. Das erste Bin jeder Zeile enthält die Stresslevels kleiner oder gleich dem definierten Minimum, analog fallen alle Werte größer als dem definierten Maximum in das letzte Bin jeder Zeile.
- **Minimum Stress Range**: Definiert die untere Schranke des Stresslevels. Alle Werte, die kleiner oder gleich diesem Wert sind, werden in einem Bin akkumuliert.
- **Maximum Stress Range**: Definiert die obere Schranke des Stresslevels. Werte, die das Maximum übersteigen, werden in das letzte Bin gezählt.
- **Number of Mean Stress Bins**: Definiert die Anzahl der Bins für die Mittelwerte. Der Wert muss mindestens eins sein. Zwei gesonderte Bins werden in der ersten/letzten Spalte der Halfcycle Count Matrix hinzugefügt, in der Mittelwerte kleiner oder gleich bzw. größer als die definierten Schranken gehalten werden.
- **Minimum negative Mean Stress**: Definiert die untere Schranke der Mittelwerte. Alle Werte, die kleiner oder gleich diesem Wert sind, werden in einem Bin akkumuliert.
- **Maximum positive Mean Stress**: Definiert die obere Schranke der Mittelwerte. Werte, die das Maximum übersteigen, werden in das letzte Bin gezählt.
- **Mean Stress Correction**: Definiert den zu verwendenden Korrekturmodus. Es stehen die Korrektur nach Goodman und Gerber zur Verfügung sowie die Möglichkeit keine Korrektur durchzuführen.
- **Use UTS Correction**: Falls aktiv, wird eine UTS Korrektur der Stress-Matrix durchgeführt.
- **UTS**: Definiert die Zugfestigkeit des überwachten Materials.
- **K1**: Definiert den Gradienten der Wöhlerkurve im Bereich `N = 1..NC1`.
- **K2**: Definiert den Gradienten der Wöhlerkurve startend bei `N >= NC2`.
- **SRI**: Definiert den „Stress Range Intercept“.
- **NC1**: Definiert den Übergangspunkt für die UTS Korrektur.
- **NC2**: Definiert den Übergangspunkt zwischen K1 und K2.
- **Woehler Curve**: Definition der Wöhlerkurve.
- **Additional Damage**: Definiert den konstanten Schaden ab Beginn. Der Gesamtschaden berechnet sich somit aus der Summe aus fDamage und dem akkumulierten Schaden bezüglich der konfigurierten Wöhlerkurve.
- **Additional Halfcycles**: Definiert den initialen Wert der gezählten Halbzyklen zu Beginn. Die gesamte Anzahl an Zyklen berechnet sich aus der Summe aus nCycles und der aktuellen Anzahlt der Eingangsdaten.
- **Input Length**: Definiert die Länge des Eingangsdatenarrays für die Blockverarbeitung.

Ausgangswerte

- **Damage**: Gibt den berechneten Ermüdungsschaden bezüglich der definierten Wöhlerkurve an.
- **HalfCycles**: Gibt die Anzahl der gezählten Halbzyklen an.
- **Range Bins**: Ausgangsarray der Länge „Number of Stress Range Bins“ mit aufsummierten Halbzyklen entlang des mittleren Stresses.
- **Mean Bins**: Ausgangsarray der Länge „Number of Mean Stress Bins“ mit aufsummierten Halbzyklen entlang des absoluten Stresses.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.

- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.9 Instantaneous Frequency



Berechnung der Momentanfrequenz eines Zeitsignals.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_InstantaneousFrequency

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

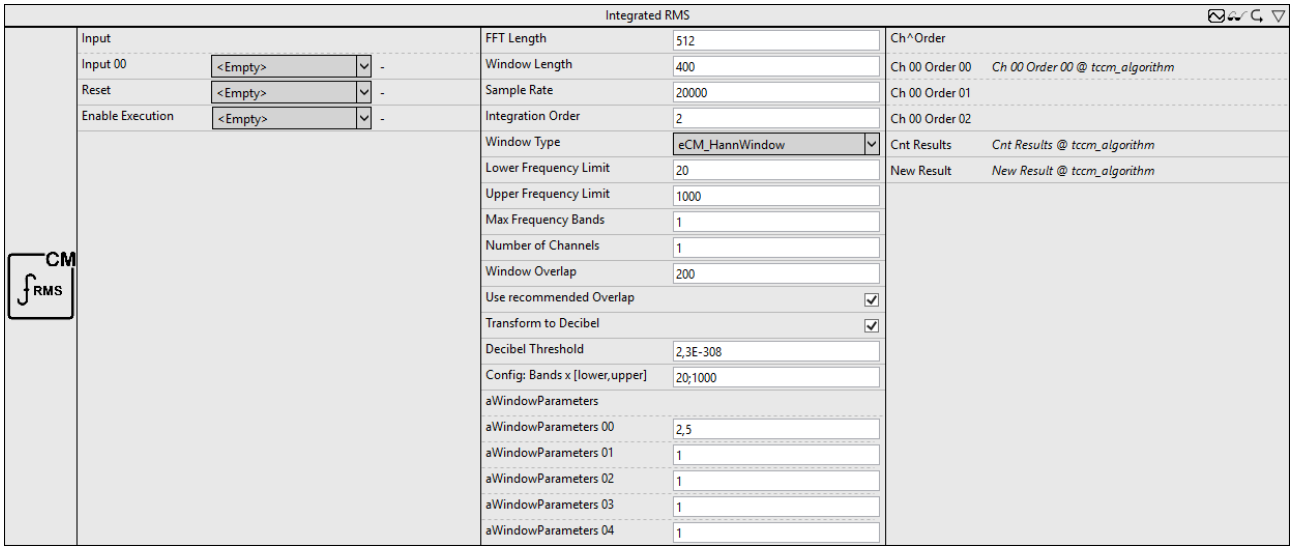
Konfigurationsoptionen

- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Sample Rate**: Abtastrate des eingehenden Zeitsignals. Der Wert wird genutzt zur Skalierung des Ergebnisses in Hz.
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Magnitude Threshold**: Definiert den Grenzwert für die numerische Berechenbarkeit der Momentanfrequenz.

Ausgangswerte

- **Output**: Ausgangsarray der halben Fensterlänge, welches die berechnete Momentanfrequenz enthält.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.10 Integrated RMS



Berechnet (optional integrierte) RMS Werte für ein- und mehrkanalige reell-wertige Zeitreihen.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_IntegratedRMS

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

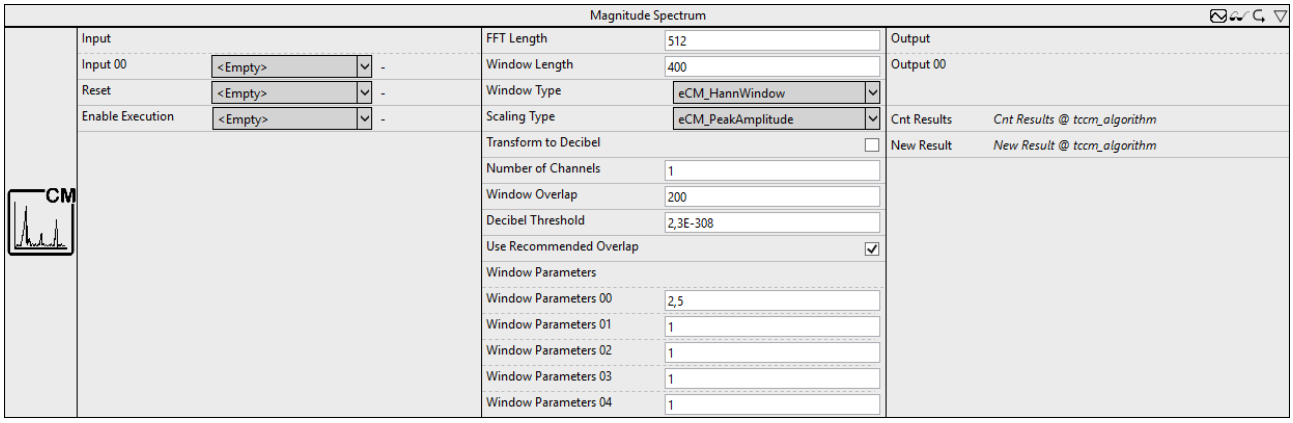
Konfigurationsoptionen

- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Sample Rate**: Abtastrate des eingehenden Zeitsignals. Der Wert wird genutzt zur Skalierung des Ergebnisses in Hz.
- **Max Integration Order**: Ist die maximale Ordnung der Integration. Dies muss eine ganze Zahl zwischen null und zwei sein. Die Zahl der ermittelten Werte pro Kanal ist (`Order+1`).
- **Window Type**: Definiert die verwendete Fensterfunktion (vom Typ E_CM_WindowType). Ein guter Standardwert ist der Fenstertyp `eCM_HannWindow`.
- **Lower Frequency Limit**: Untere Grenze des berücksichtigten Frequenzintervalls. Die untere Grenzfrequenz muss mindestens so groß sein wie die Abtastrate geteilt durch die FFT-Länge.
- **Upper Frequency Limit**: Obere Grenze des berücksichtigten Frequenzintervalls. Die obere Grenzfrequenz darf höchstens so groß sein wie die halbe Abtastrate und muss größer als die untere Grenzfrequenz sein.
- **Max Frequency Bands**: Dieser Wert gibt die maximale Anzahl an Frequenzbändern an, für die der RMS-Wert berechnet wird.
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Window Overlap**: Definiert die Anzahl der überlappenden Samples. Diese muss größer oder gleich null sein.
- **Use Recommended Overlap**: Falls selektiert, wird eine empfohlene Überlappung intern berechnet (siehe F_CM_CalculateRecommendedOverlap).
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)
- **Config: MaxBands x [fmin, fmax]**: Definition der konfigurierbaren unteren und oberen Schranke der Frequenzbänder.
- **Window Parameters**: Beinhaltet die freien Parameter ausgewählter Fensterfunktionen. Bei der Verwendung von `eCM_KaiserWindow` definiert der erste Eintrag den Parameter beta, wird das `eCM_FlatTopWindow` verwendet, werden alle Parameter genutzt. Siehe Abschnitt Fensterfunktionen.

Ausgangswerte

- **Output**: Ausgangsarrays entsprechend der Anzahl der Frequenzbänder mit Kanal- und ordnungsspezifischen RMS-Werten.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.11 Magnitude Spectrum



Berechnet das Magnitudenspektrum (auch Betragsspektrum oder Amplitudenspektrum genannt) eines reellwertigen Eingangssignals.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_MagnitudeSpectrum

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

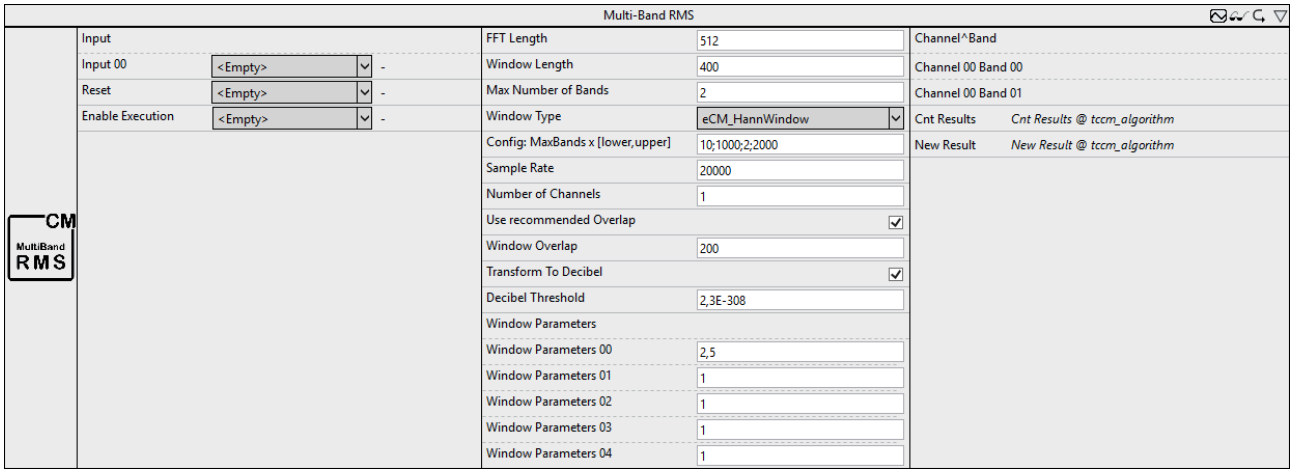
Konfigurationsoptionen

- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Window Type**: Definiert die verwendete Fensterfunktion (vom Typ E_CM_WindowType). Ein guter Standardwert ist der Fenstertyp `eCM_HannWindow`.
- **Scaling Type**: ermöglicht eine Auswahl der verwendeten Skalierung (vom Typ E_CM_ScalingType), falls eine absolute Skalierung benötigt wird. Standardwert ist `eCM_DiracScaling`. Bei der Auswahl der Skalierung sollte die Art des Signals, entweder deterministische Signale oder breitbandige Signale mit stochastischen Anteil, berücksichtigt werden; Beide Arten erfordern unterschiedliche Skalierungen.
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Window Overlap**: Definiert die Anzahl der überlappenden Samples. Diese muss größer oder gleich null sein.
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)
- **Use Recommended Overlap**: Falls selektiert, wird eine empfohlene Überlappung intern berechnet (siehe F_CM_CalculateRecommendedOverlap).
- **Window Parameters**: Beinhaltet die freien Parameter ausgewählter Fensterfunktionen. Bei der Verwendung von `eCM_KaiserWindow` definiert der erste Eintrag den Parameter beta, wird das `eCM_FlatTopWindow` verwendet, werden alle Parameter genutzt. Siehe Abschnitt Fensterfunktionen.

Ausgangswerte

- **Output**: Ausgangsarray der Länge `N / 2 + 1` (bei FFT Länge `N`), welches die berechneten Spektrallinien des Magnitudenspektrums enthält.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.12 Multi Band RMS



Berechnet RMS-Werte für ein- und mehrkanalige reell-wertige Zeitreihen für konfigurierbarer Frequenzbänder.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_CMA_MultiBandRMS

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

Konfigurationsoptionen

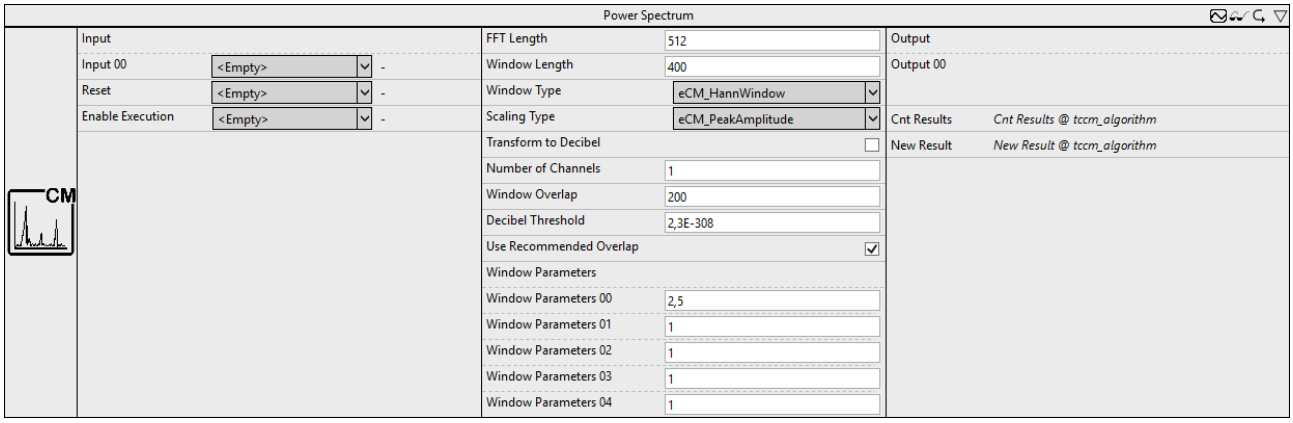
- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Max Number of Bands**: Dieser Wert gibt die maximale Anzahl an Frequenzbändern an, für die der RMS-Wert berechnet wird.
- **Window Type**: Definiert die verwendete Fensterfunktion (vom Typ E_CM_WindowType). Ein guter Standardwert ist der Fenstertyp `eCM_HannWindow`.
- **Config: MaxBands x [fmin, fmax]**: Definition der konfigurierbaren unteren und oberen Schranke der Frequenzbänder.
- **Sample Rate**: Abtastrate des eingehenden Zeitsignals. Der Wert wird genutzt zur Skalierung des Ergebnisses in Hz.
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Use Recommended Overlap**: Falls selektiert, wird eine empfohlene Überlappung intern berechnet (siehe F_CM_CalculateRecommendedOverlap).
- **Window Overlap**: Definiert die Anzahl der überlappenden Samples. Diese muss größer oder gleich null sein.
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)
- **Window Parameters**: Beinhaltet die freien Parameter ausgewählter Fensterfunktionen. Bei der Verwendung von `eCM_KaiserWindow` definiert der erste Eintrag den Parameter beta, wird das `eCM_FlatTopWindow` verwendet, werden alle Parameter genutzt. Siehe Abschnitt Fensterfunktionen.

Ausgangswerte

- **Output**: Kanal- und bandspezifische RMS-Werte.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.

- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.13 Power Spectrum



Berechnung des Leistungsspektrums eines reell-wertigen Eingangssignals sowie optionale Skalierung zu Dezibel.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_CMA_PowerSpectrum

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

Konfigurationsoptionen

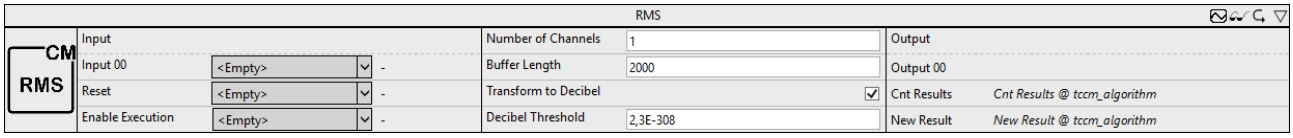
- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Window Type**: Definiert die verwendete Fensterfunktion (vom Typ E_CM_WindowType). Ein guter Standardwert ist der Fenstertyp `eCM_HannWindow`.
- **Scaling Type**: ermöglicht eine Auswahl der verwendeten Skalierung (vom Typ E_CM_ScalingType), falls eine absolute Skalierung benötigt wird. Standardwert ist `eCM_DiracScaling`. Bei der Auswahl der Skalierung sollte die Art des Signals, entweder deterministische Signale oder breitbandige Signale mit stochastischen Anteil, berücksichtigt werden; Beide Arten erfordern unterschiedliche Skalierungen.
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Window Overlap**: Definiert die Anzahl der überlappenden Samples. Diese muss größer oder gleich null sein.
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)
- **Use Recommended Overlap**: Falls selektiert, wird eine empfohlene Überlappung intern berechnet (siehe F_CM_CalculateRecommendedOverlap).
- **Window Parameters**: Beinhaltet die freien Parameter ausgewählter Fensterfunktionen. Bei der Verwendung von `eCM_KaiserWindow` definiert der erste Eintrag den Parameter beta, wird das `eCM_FlatTopWindow` verwendet, werden alle Parameter genutzt. Siehe Abschnitt Fensterfunktionen.

Ausgangswerte

- **Output**: Ausgangsarray der Länge `N / 2 + 1` (bei FFT Länge `N`), welches die berechneten Spektrallinien des Magnitudenspektrums enthält.

- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.14 RMS



Berechnet für ein- und mehrkanalige reell-wertige Signale den zeitlichen RMS-Wert.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_CMA_RMS

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

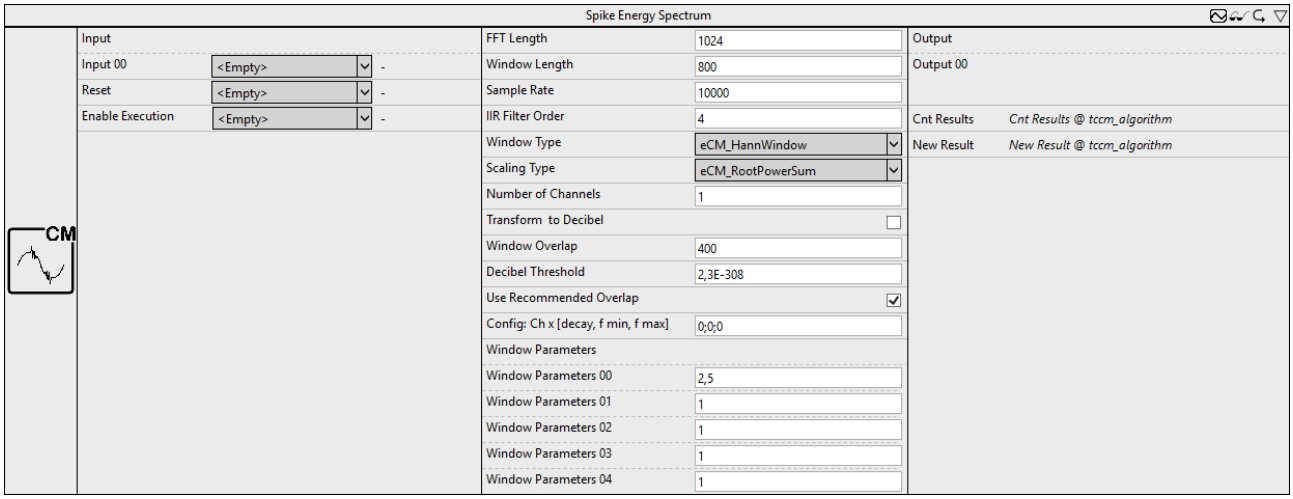
Konfigurationsoptionen

- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Buffer Length**: Ist die Zahl der im internen Puffer gehaltenen Eingangswerte je Kanal.
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)

Ausgangswerte

- **Output**: RMS-Wert des zugehörigen Eingangsdaten-Streams.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.15 Spike Energy Spectrum



Analyse von Spitzenenergie hochfrequenter Signalanteile.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier:
FB_CMA_SpikeEnergySpectrum

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

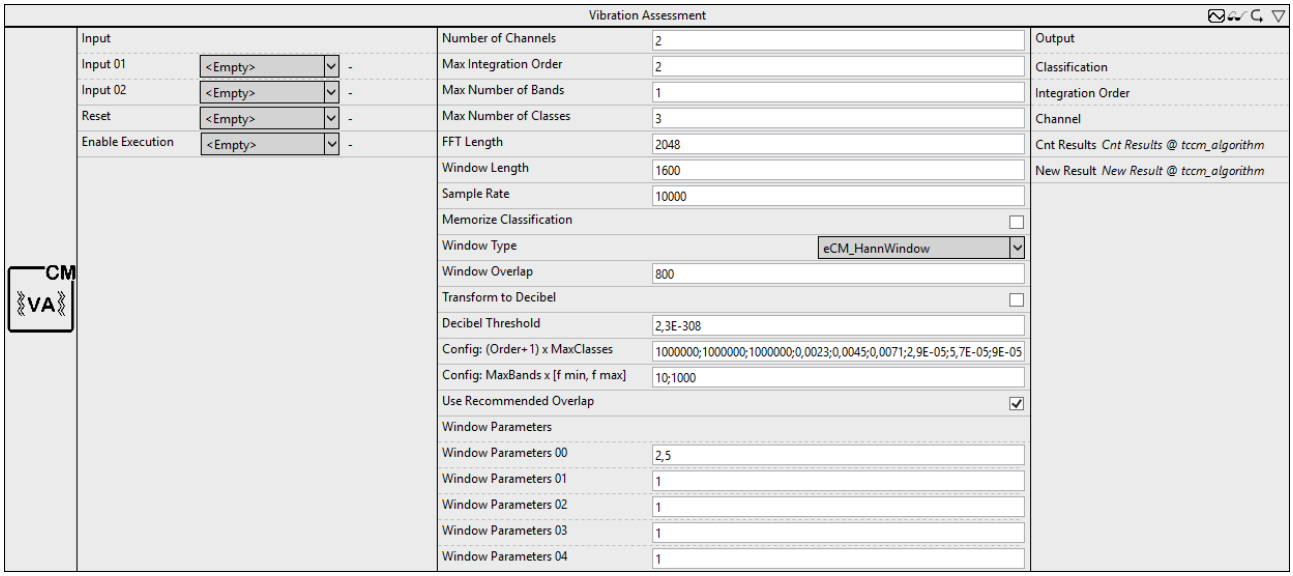
Konfigurationsoptionen

- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Sample Rate**: Abtastrate des eingehenden Zeitsignals. Der Wert wird genutzt zur Skalierung des Ergebnisses in Hz.
- **IIR Filter Order**: Definiert die Ordnung des verwendeten IIR-Filters.
- **Window Type**: Definiert die verwendete Fensterfunktion (vom Typ E_CM_WindowType). Ein guter Standardwert ist der Fenstertyp `eCM_HannWindow`.
- **Scaling Type**: ermöglicht eine Auswahl der verwendeten Skalierung (vom Typ E_CM_ScalingType), falls eine absolute Skalierung benötigt wird. Standardwert ist `eCM_DiracScaling`. Bei der Auswahl der Skalierung sollte die Art des Signals, entweder deterministische Signale oder breitbandige Signale mit stochastischen Anteil, berücksichtigt werden; Beide Arten erfordern unterschiedliche Skalierungen.
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Window Overlap**: Definiert die Anzahl der überlappenden Samples. Diese muss größer oder gleich null sein.
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)
- **Use Recommended Overlap**: Falls selektiert, wird eine empfohlene Überlappung intern berechnet (siehe F_CM_CalculateRecommendedOverlap).
- **Config: Ch x [decay, fmin, fmax]**: Definition der konfigurierbaren Parameter: Die Abklingzeit sowie das betrachtete Frequenzband (durch untere und obere Grenzen). Die Abklingzeit sollte optimalerweise so gewählt sein, dass die Spitzenenergie vollständig abklingen kann, d.h. 'Abklingzeit' > 1 / 'Fehlerfrequenz'.
- **Window Parameters**: Beinhaltet die freien Parameter ausgewählter Fensterfunktionen. Bei der Verwendung von `eCM_KaiserWindow` definiert der erste Eintrag den Parameter beta, wird das `eCM_FlatTopWindow` verwendet, werden alle Parameter genutzt. Siehe Abschnitt Fensterfunktionen.

Ausgangswerte

- **Output**: Ausgangsarray der Länge `N / 2 + 1` (bei FFT Länge `N`), welches die berechneten Spektrallinien des Spitzenenergiespektrums enthält.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.16 Vibration Assessment



Schwingungsbeurteilung reell-wertiger Eingangssignale in Anlehnung an ISO 10816-3.

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_CMA_VibrationAssessment

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

Konfigurationsoptionen

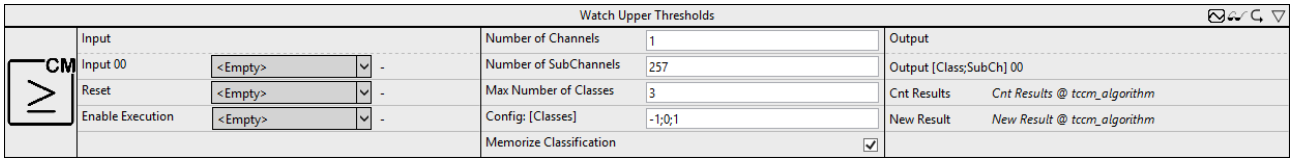
- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Max Integration Order**: Ist die maximale Ordnung der Integration. Dies muss eine ganze Zahl zwischen null und zwei sein. Die Zahl der ermittelten Werte pro Kanal ist (`Order+1`).
- **Max Number of Bands**: Dieser Wert gibt die maximale Anzahl an Frequenzbändern an, für die der RMS-Wert berechnet wird.
- **Max Number of Classes**: Definiert die maximale Anzahl an Klassen, die konfiguriert werden. Dieser Wert muss mindestens eins sein.
- **FFT Length**: Ist die Länge der FFT. Sie muss größer als eins und eine ganzzahlige Potenz von zwei sein.
- **Window Length**: Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Sample Rate**: Abtastrate des eingehenden Zeitsignals. Der Wert wird genutzt zur Skalierung des Ergebnisses in Hz.
- **Memorize Classification**: Wenn angewählt, berechnet der Baustein die Nummer der höchsten Kategorie und des entsprechenden Kanals bei jedem Schritt neu. Andernfalls werden die Ergebniswerte bei Überschreitung eines Grenzwerts so lange gespeichert, bis ein Reset ausgeführt wird oder ein Kanal eine höhere Kategorie erreicht.
- **Window Type**: Definiert die verwendete Fensterfunktion (vom Typ E_CM_WindowType). Ein guter Standardwert ist der Fenstertyp `eCM_HannWindow`.
- **Window Overlap**: Definiert die Anzahl der überlappenden Samples. Diese muss größer oder gleich null sein.
- **Transform to Decibel**: Ist ein boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x -> 20 * log10(x).
- **Decibel Threshold**: Ist ein sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt. (Zweck ist die Vermeidung von Wertbereichsfehlern. Der Logarithmus von Null ist nicht definiert und strebt für den Grenzwert kleiner Argumente gegen minus unendlich. Entsprechendes gilt für das Argument der Zahl Null, arg(0). Der kleinste mögliche Wert ist 2.3e-308)

- **Config: (Order+1) x MaxClasses**: Definition der konfigurierbaren Schwellwerte im Bezug auf die Integrationsordnung.
- **Config: MaxBands x [fmin, fmax]**: Definition der konfigurierbaren unteren und oberen Schranke der Frequenzbänder.
- **Use Recommended Overlap**: Falls selektiert, wird eine empfohlene Überlappung intern berechnet (siehe F_CM_CalculateRecommendedOverlap).
- **Window Parameters**: Beinhaltet die freien Parameter ausgewählter Fensterfunktionen. Bei der Verwendung von `eCM_KaiserWindow` definiert der erste Eintrag den Parameter beta, wird das `eCM_FlatTopWindow` verwendet, werden alle Parameter genutzt. Siehe Abschnitt Fensterfunktionen.

Ausgangswerte

- **Output**: Das Ergebnis ist ein eindimensionales Array, welches für jedes Frequenzband drei Werte bereithält, die höchste berechnete Klassifikation (im Bereich -1..‘Max Number of Classes‘) sowie die zugehörige Integrationsordnung (im Bereich 0..‘Max Integration Order‘) und den Kanal (im Bereich 1..‘Number of Channels‘).
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.4.14.2.1.17 Watch Upper Thresholds



Konfigurierbare Schwellwertüberwachung von mehrkanaligen Daten.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier:
FB_CMA_WatchUpperThresholds

Die Dokumentation der TwinCAT 3 Condition Monitoring SPS-Bibliothek finden Sie hier: Übersicht.

Konfigurationsoptionen

- **Number of Channels**: Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.
- **Number of SubChannels**: Definiert die Anzahl der unabhängigen Unterkanäle. Dieser Wert wird automatisch der Länge des verknüpften Eingangsarrays angepasst. Falls mehrere Kanäle genutzt werden, müssen alle Längen übereinstimmen.
- **Max Number of Classes**: Definiert die maximale Anzahl an Klassen, die konfiguriert werden. Dieser Wert muss mindestens eins sein.
- **Config: [classes]**: Definition der konfigurierbaren Schwellwerte.
- **Memorize Classification**: Wenn angewählt, berechnet der Baustein die Nummer der höchsten Kategorie und des entsprechenden Kanals bei jedem Schritt neu. Andernfalls werden die Ergebniswerte bei Überschreitung eines Grenzwerts so lange gespeichert, bis ein Reset ausgeführt wird oder ein Kanal eine höhere Kategorie erreicht.

Ausgangswerte

- **Output**: Ausgangsarray der Länge zwei für jeden Eingangsdaten-Stream. Die Werte des Tupels beschreiben die Klassifikation sowie den Unterkanal, in dem diese Klasse identifiziert wurde.
- **Cnt Results**: Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result**: Ist ein boolscher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

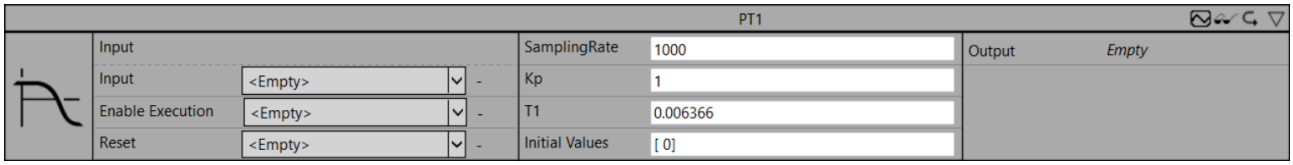
6.4.14.2.2 Filter

Die Bibliothek **TwinCAT Filter** bietet Algorithmen zur Realisierung von Digitalfiltern.

Übersicht der Algorithmen

Algorithmus	Beschreibung	SPS Funktionsbaustein
PT1 [▶ 328]	Realisierung eines Verzögerungsglieds 1. Ordnung (PT1-Glied).	FB_FTR_PT1
IIRCoeff [▶ 329]	Realisierung eines IIR-Filters (Infinitive-Impulse-Response) über Filterkoeffizienten.	FB_FTR_IIRCoeff
IIRSpec [▶ 329]	Realisierung eines IIR-Filters (Infinitive-Impulse-Response) über Filterspezifikationen.	FB_FTR_IIRSpec
PT2 [▶ 330]	Realisierung eines Verzögerungsglieds 2. Ordnung (PT2-Glied).	FB_FTR_PT2
PT3 [▶ 330]	Realisierung eines Verzögerungsglieds 3. Ordnung (PT3-Glied).	FB_FTR_PT3
PTn [▶ 331]	Realisierung eines Verzögerungsglieds n-ter Ordnung mit gleichen Zeitkonstanten.	FB_FTR_PTn
IIRSos [▶ 331]	Realisierung eines IIR-Filters (Infinitive-Impulse-Response) über Filterkoeffizienten.	FB_FTR_IIRSos
Notch [▶ 331]	Realisierung eines Bandstopp-Filters mit schmaler Bandbreite.	FB_FTR_Notch
LeadLag [▶ 332]	Realisierung eines Phasenkorrekturglieds 1. Ordnung.	FB_FTR_LeadLag
PT2oscillation [▶ 332]	Realisierung eines schwingungsfähigen Verzögerungsglieds 2. Ordnung.	FB_FTR_PT2oscillation
PTt [▶ 333]	Realisierung eines Totzeitglieds (PTt-Glied).	FB_FTR_PTt
Median [▶ 333]	Realisierung eines Median Filters.	FB_FTR_Median
ActualValue [▶ 333]	Realisierung einer Plausibilitätskontrolle und Filterung einer gemessenen Eingangsgröße.	FB_FTR_ActualValue
Gaussian [▶ 334]	Realisierung eines Gauß-Filters.	FB_FTR_Gaussian

6.4.14.2.2.1 PT1



Realisierung eines Verzögerungsglieds 1. Ordnung (PT1-Glied).

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_PT1

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

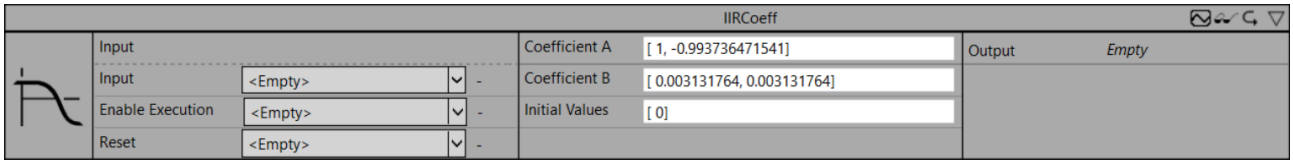
Konfigurationsparameter

- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **Kp**: Verstärkungsfaktor (größer Null).
- **T1**: Zeitkonstante T_1 in Sekunden (größer Null).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.2 IIRCoeff



Realisierung eines Infinite-Impulse-Response-Filter (IIR).

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_IIRCoeff

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

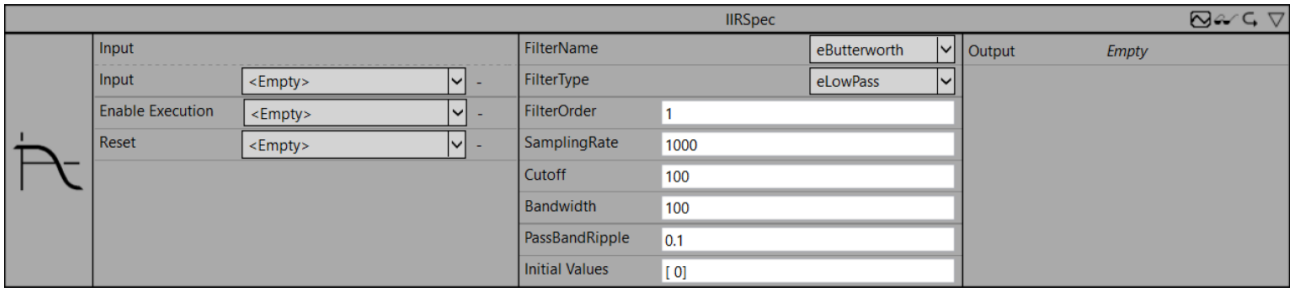
Konfigurationsparameter

- **Coefficient A**: Die Koeffizienten a_k (Nenner) $[a_0, a_1, a_2, \ldots, a_N]$ sind frei wählbar.
- **Coefficient B**: Die Koeffizienten b_k (Zähler) $[b_0, b_1, b_2, \ldots, b_M]$ sind frei wählbar.
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.3 IIRSpec



Realisierung eines Infinite-Impulse-Response-Filter (IIR).

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_FTR_IIRSpec

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

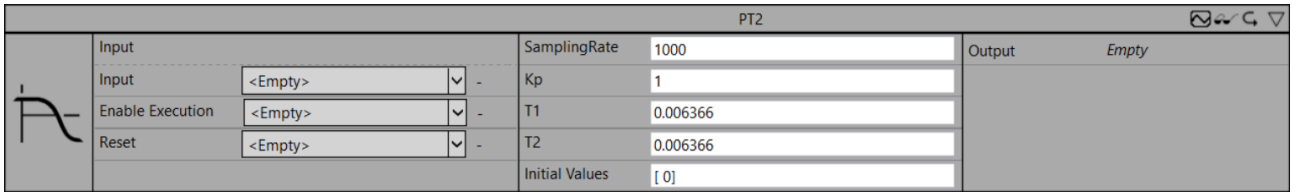
Konfigurationsparameter

- **FilterName**: Beschreibt die Filterrealisierung (Butterworth, Chebyshev).
- **FilterType**: Beschreibt den Filtertyp (Hochpass, Tiefpass, Bandpass, Bandstopp).
- **FilterOrder**: Filterordnung (max. 20 für Hoch- und Tiefpass, max. 10 für Bandpass und Bandstopp).
- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **Cutoff**: Grenzfrequenz in Hz (größer 0 und kleiner SamplingRate/2).
- **Bandwidth**: Bandbreite in Hz bezüglich Bandpass und Bandstopp.
- **PassBandRipple**: Welligkeit des Amplitudengangs im Durchlassbereich des Filters in dB (größer 0).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.4 PT2



Realisierung eines Verzögerungsglieds 2. Ordnung (PT2-Glied).

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_PT2

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

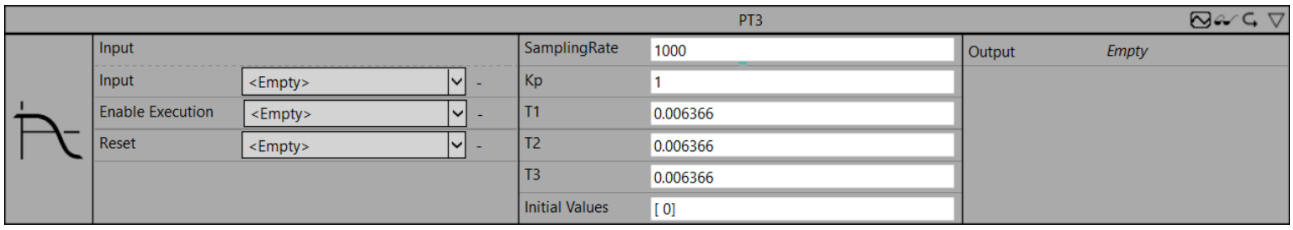
Konfigurationsparameter

- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **Kp**: Verstärkungsfaktor (größer Null).
- **T1**: Zeitkonstante T_1 in Sekunden (größer Null).
- **T2**: Zeitkonstante T_2 in Sekunden (größer Null).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.5 PT3



Realisierung eines Verzögerungsglieds 3. Ordnung (PT3-Glied).

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_PT3

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

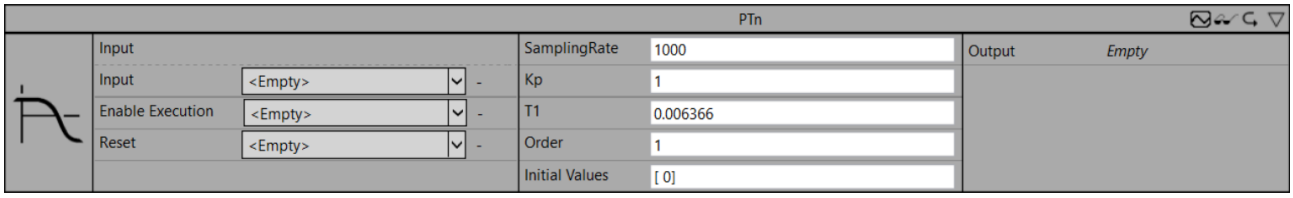
Konfigurationsparameter

- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **Kp**: Verstärkungsfaktor (größer Null).
- **T1**: Zeitkonstante T_1 in Sekunden (größer Null).
- **T2**: Zeitkonstante T_2 in Sekunden (größer Null).
- **T3**: Zeitkonstante T_3 in Sekunden (größer Null).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.6 PTn



Realisierung eines Verzögerungsglieds n-ter Ordnung mit gleichen Zeitkonstanten.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_PTn

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

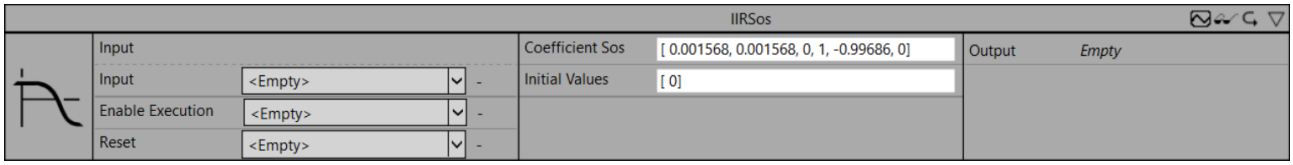
Konfigurationsparameter

- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **Kp**: Verstärkungsfaktor (größer Null).
- **T1**: Zeitkonstante T_1 in Sekunden (größer Null).
- **Order**: Ordnung des Filters (1..10).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.7 IIRSos



Realisierung eines Infinite-Impulse-Response-Filter (IIR).

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_IIRSos.

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

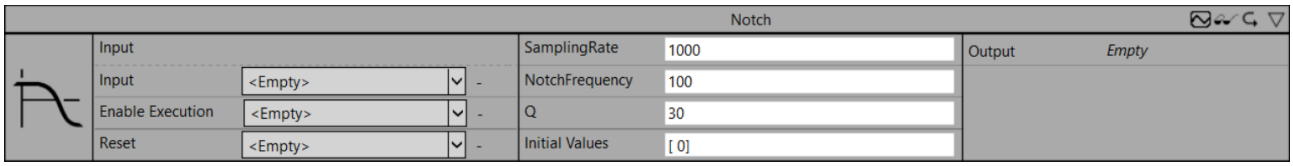
Konfigurationsparameter

- **Coefficient Sos**: Die Koeffizienten [b_{01}, b_{11}, b_{21}, a_{01}, a_{11}, a_{21}, b_{02}, b_{12}, b_{22}, a_{02}, a_{12}, a_{22},…, b_{0M}, b_{1M}, b_{2M}, a_{0M}, a_{1M}, a_{2M}] sind frei wählbar.
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.8 Notch



Realisierung eines Bandstopp-Filters mit schmaler Bandbreite.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_Notch

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

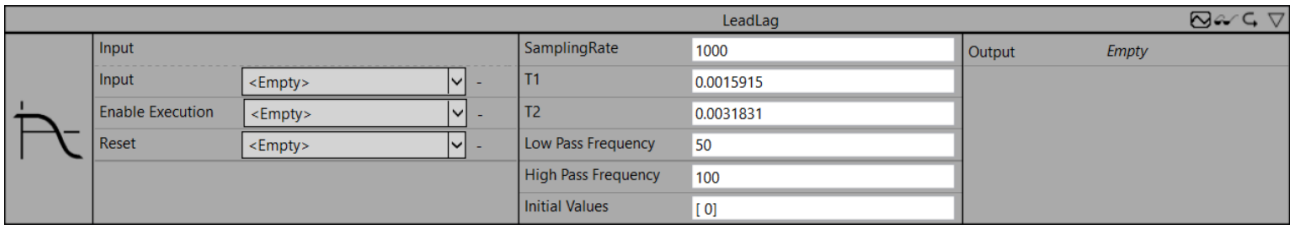
Konfigurationsparameter

- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **NotchFrequency**: Notchfrequenz in Hz (größer 0 und kleiner SamplingRate/2).
- **Q**: Q-Faktor = Notchfrequenz/Bandbreite (größer Null).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.9 LeadLag



Realisierung eines Phasenkorrekturglieds 1. Ordnung.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_LeadLag.

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

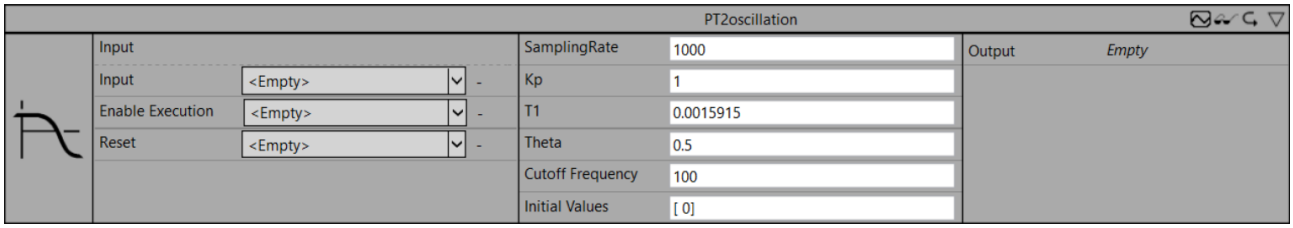
Konfigurationsparameter

- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **T1**: Zeitkonstante T_1 in Sekunden (größer Null).
- **T2**: Zeitkonstante T_2 in Sekunden (größer Null).
- **Low Pass Frequency (optional)**: Tiefpassfrequenz in Hz (größer 0 und kleiner SamplingRate/2).
- **High Pass Frequency (optional)**: Hochpassfrequenz in Hz (größer 0 und kleiner SamplingRate/2).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.10 PT2oscillation



Realisierung eines schwingungsfähigen Verzögerungsglieds 2. Ordnung.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_PT2oscillation

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

Konfigurationsparameter

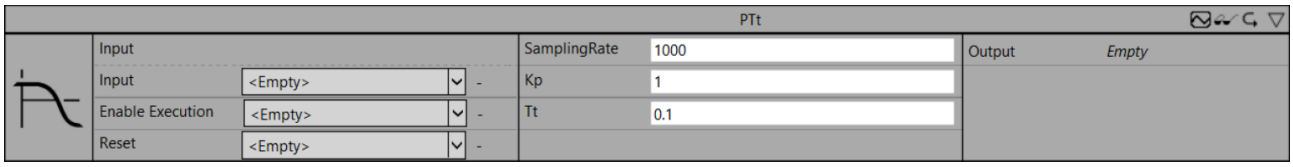
- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **Kp**: Verstärkungsfaktor (größer Null).
- **T1**: Zeitkonstante T_1 in Sekunden (größer Null).

- **Theta**: Dämpfungsfaktor (größer Null).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.11 PTt



Realisierung eines Totzeitglieds (PTt-Glied).

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_PTt

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

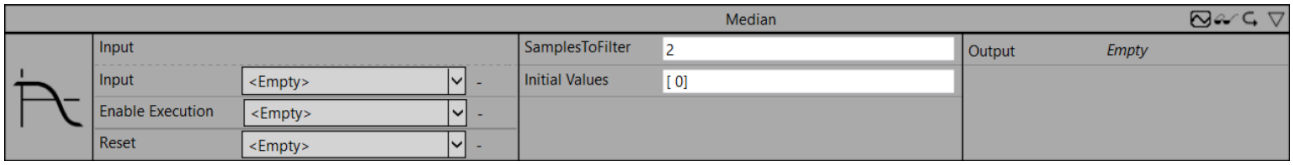
Konfigurationsparameter

- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **Kp**: Verstärkungsfaktor (größer Null).
- **Tt**: Totzeit T_t in Sekunden (größer Null).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.12 Median



Realisierung eines Median-Filters.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_Median

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

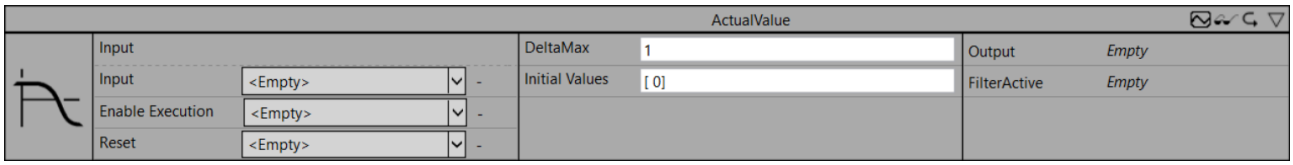
Konfigurationsparameter

- **SamplesToFilter**: Anzahl der Samples zur Bildung des gleitenden Mittelwerts (oft als Fenstergröße bezeichnet).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.2.13 ActualValue



Realisierung einer Plausibilitätskontrolle und Filterung einer gemessenen Eingangsgröße.

Die Dokumentation des zugehörigen SPS-Funktionsbausteins finden Sie hier: FB_FTR_ActualValue

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

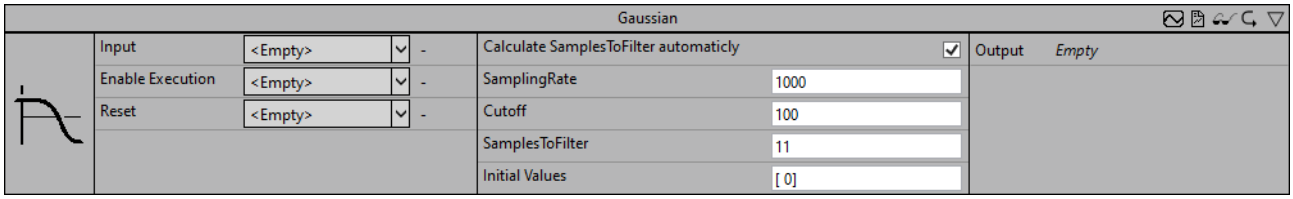
Konfigurationsparameter

- **DeltaMax**: Maximale Differenz zweier aufeinander folgender Eingangswerte (größer oder gleich Null).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.
- **FilterActive**: Zeigt an, welche Elemente des Eingangssignals verändert wurden.

6.4.14.2.2.14 Gaussian



Realisierung eines Infinite-Impulse-Response-Filter (IIR).

Die Dokumentation des zugehörigen SPS Funktionsbausteins finden Sie hier: FB_FTR_IIRSpec

Die Dokumentation der TwinCAT 3 Filter SPS-Bibliothek finden Sie hier: Übersicht

Konfigurationsparameter

- **Calculate SamplesToFilter automaticly**: Ist ein boolescher Wert, der angibt, ob SamplesToFilter automatisch berechnet werden soll.
- **SamplingRate**: Abtastrate f_s in Hz (größer Null).
- **Cutoff**: Grenzfrequenz in Hz (größer 0 und kleiner SamplingRate/2).
- **SamplesToFilter**: Anzahl der Samples zur Bildung des gleitenden Mittelwerts (oft als Fenstergröße bezeichnet).
- **Initial Values (optional)**: Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output**: Manipuliertes Ausgangssignal.

6.4.14.2.3 Power Monitoring

Beckhoff bietet in seinem I/O-Portfolio verschiedene Klemmen zur Realisierung einer Energienetzanalyse und -überwachung. Sogenannte Power-Measurement-Klemmen wie die EL3403 oder EL3413 liefern dabei in einem Ein- oder Drei-Phasen-Netz direkt RMS-Werte (Effektivwerte) für Strom und Spannung sowie Wirk-, Blind- und Scheinleistung. Mit den Power-Monitoring-Klemmen EL3773 und EL3783 können sogar die Rohdaten von Strom und Spannung aufgezeichnet werden, sodass Ereignisse im Netz noch genauer detektiert und die Berechnung beliebiger Werte in der Steuerung selbst ausgeführt werden können.

Die TwinCAT 3 Power Monitoring Function ist eine SPS-Bibliothek zur Auswertung von Strom- und Spannungsrohdaten, die von Power-Monitoring-Klemmen bereitgestellt werden.

Die Bibliothek stellt Funktionsbausteine zur Berechnung von RMS-Werten (Effektivwerten) für Strom, Spannung und Leistung zur Verfügung. Diese können als Momentan- oder Durchschnittswert ausgegeben werden. Zusätzlich stehen am Funktionsbaustein auch Maximal- und Minimalwerte zur Verfügung. Frequenzen und Frequenzspektren können wie die Harmonischen im Netz und deren Belastung in Form der Total Harmonic Distortion (THD) bestimmt werden. Des Weiteren bietet die Bibliothek Funktionsbausteine an, mit denen die Frequenzen und Drehfelder ermittelt werden können.

Allgemeine Algorithmen

Algorithmus	Beschreibung	SPS-Funktionsbaustein
Scaling [▶ 336]	Skalierung der Eingangsdaten von Strom und Spannung	FB_PMA_Scaling
Scaling EL3773 [▶ 337]	Skalierung der Eingangsdaten von Strom und Spannung, die als Eingänge von EtherCAT-Klemmen des Typs EL3773 bereitgestellt werden.	FB_PMA_Scaling_EL3773
Scaling EL3783 [▶ 337]	Skalierung der Eingangsdaten von Strom und Spannung, die als Eingänge von EtherCAT-Klemmen des Typs EL3783 bereitgestellt werden.	FB_PMA_Scaling_EL3783

Einphasige Algorithmen

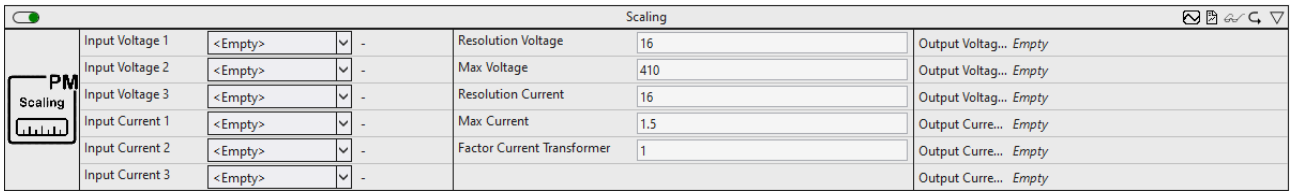
Algorithmus	Beschreibung	SPS-Funktionsbaustein
Frequency Period 1Ph [▶ 338]	Berechnung der Grundfrequenz des Eingangssignals.	FB_PMA_Frequency_Period_1Ph
Basic Values Period 1Ph [▶ 339]	Berechnung von Analysewerten für den zeitlichen Signalverlauf von Strom und Spannung.	FB_PMA_BasicValues_Period_1Ph
Power Values Period 1Ph [▶ 340]	Berechnung von Leistungskennwerten des angeschlossenen Verbrauchers auf Basis der Signalperiode.	FB_PMA_PowerValues_Period_1Ph
Harmonics Period 1Ph [▶ 341]	Berechnung der Harmonischen von Strom und Spannung auf Basis der Signalperiode.	FB_PMA_Harmonics_Period_1Ph
Spectrum 1Ph [▶ 342]	Berechnung der Betragsspektren der Strom- und Spannungswerte.	FB_PMA_Spectrum_1Ph
Spectrum Quantiles 1Ph [▶ 343]	Berechnung der Betragsspektren der Strom- und Spannungswerte. Zusätzlich können p-Quantile der Verteilung des Spektrums berechnet werden	FB_PMA_Spectrum_Quantiles_1Ph
Power Values 1Ph [▶ 344]	Berechnung von Leistungskennwerten des angeschlossenen Verbrauchers auf Basis von Berechnungen im Frequenzbereich.	FB_PMA_PowerValues_1Ph
Harmonics 1Ph [▶ 345]	Berechnung der Harmonischen von Strom und Spannung.	FB_PMA_Harmonics_1Ph

Dreiphasige Algorithmen

Algorithmus	Beschreibung	SPS-Funktionsbaustein
Frequency Period 3Ph [▶ 346]	Berechnung der Grundfrequenz des Eingangssignals.	FB_PMA_Frequency_Period_3Ph
Basic Values Period 3Ph [▶ 347]	Berechnung von Analysewerten für den zeitlichen Signalverlauf von Strom und Spannung.	FB_PMA_BasicValues_Period_3Ph
Power Values Period 3Ph [▶ 348]	Berechnung von Leistungskennwerten des angeschlossenen Verbrauchers auf Basis der Signalperiode.	FB_PMA_PowerValues_Period_3Ph
Harmonics Period 3Ph [▶ 350]	Berechnung der Harmonischen von Strom und Spannung auf Basis der Signalperiode.	FB_PMA_Harmonics_Period_3Ph
Spectrum 3Ph [▶ 351]	Berechnung der Betragsspektren der Strom- und Spannungswerte.	FB_PMA_Spectrum_3Ph
Spectrum Quantiles 3Ph [▶ 352]	Berechnung der Betragsspektren der Strom- und Spannungswerte. Zusätzlich können p-Quantile der Verteilung des Spektrums berechnet werden	FB_PMA_Spectrum_Quantiles_3Ph
Power Values 3Ph [▶ 353]	Berechnung von Leistungskennwerten des angeschlossenen Verbrauchers auf Basis von Berechnungen im Frequenzbereich.	FB_PMA_PowerValues_3Ph
Harmonics 3Ph [▶ 355]	Berechnung der Harmonischen von Strom und Spannung.	FB_PMA_Harmonics_3Ph

6.4.14.2.3.1 Allgemein

6.4.14.2.3.1.1 Scaling



Skalierung der Eingangsdaten von Strom und Spannung, die beispielsweise als Eingänge von EtherCAT-Klemmen bereitgestellt werden.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Scaling.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Resolution Voltage:** Auflösung der Spannung in Bit. Wertebereich: 1-32.
- **Max Voltage:** Gibt die maximale Amplitude der Eingangsspannung an.
- **Resolution Current:** Auflösung des Stroms in Bit. Wertebereich: 1-32.
- **Max Current:** Gibt die maximale Amplitude des Eingangsstroms an.
- **Factor Current Transformer:** Faktor des Stromwandlers. Dieser berechnet sich aus der Division von Eingangsstrom und Ausgangsstrom.

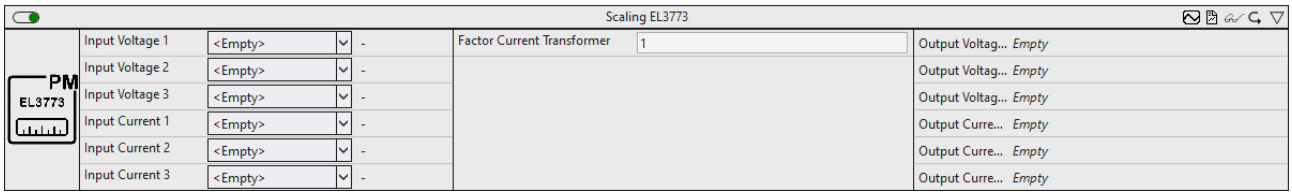
Eingangswerte

- **Input Voltage 1 … Input Voltage 3:** Rohwerte der Spannung der einzelnen Phasen.
- **Input Current 1 … Input Current 3:** Rohwerte des Stroms der einzelnen Phasen.

Ausgangswerte

- **Output Voltage 1 … Input Voltage 3:** Skalierte Spannungswerte der einzelnen Phasen.
- **Output Current 1 … Input Current 3:** Skalierte Stromwerte der einzelnen Phasen.

6.4.14.2.3.1.2 Scaling EL3773



Skalierung der Eingangsdaten von Strom und Spannung, die als Eingänge von EtherCAT-Klemmen des Typs EL3773 bereitgestellt werden.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Scaling_EL3773.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Factor Current Transformer:** Faktor des Stromwandlers. Dieser berechnet sich aus der Division von Eingangsstrom und Ausgangsstrom.

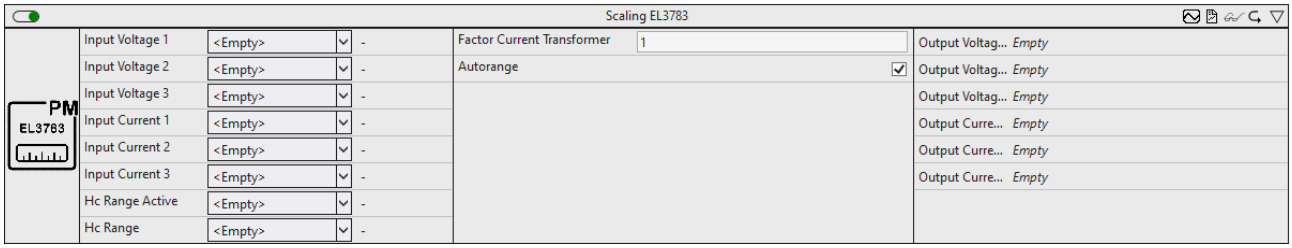
Eingangswerte

- **Input Voltage 1 … Input Voltage 3:** Rohwerte der Spannung der einzelnen Phasen.
- **Input Current 1 … Input Current 3:** Rohwerte des Stroms der einzelnen Phasen.

Ausgangswerte

- **Output Voltage 1 … Input Voltage 3:** Skalierte Spannungswerte der einzelnen Phasen.
- **Output Current 1 … Input Current 3:** Skalierte Stromwerte der einzelnen Phasen.

6.4.14.2.3.1.3 Scaling EL3783



Skalierung der Eingangsdaten von Strom und Spannung, die als Eingänge von EtherCAT-Klemmen des Typs EL3783 bereitgestellt werden.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Scaling_EL3783.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Factor Current Transformer:** Faktor des Stromwandlers. Dieser berechnet sich aus der Division von Eingangsstrom und Ausgangsstrom.
- **Autorange:** Automatische Bereichsauswahl in der EtherCAT-Klemme aktiviert.

Eingangswerte

- **Input Voltage 1 … Input Voltage 3:** Rohwerte der Spannung der einzelnen Phasen.
- **Input Current 1 … Input Current 3:** Rohwerte des Stroms der einzelnen Phasen.
- **Hc Range Active:** *TRUE*, wenn der Auto-Range-Modus an der Klemme aktiv ist.
- **Hc Range:** Die aktuellen Messbereichsinformationen der EL3783 im Auto-Range-Modus.
- **Use 5A Range (optional):** Wenn der Wert *TRUE* ist, wird der 5A-Messbereich der EL3783 genutzt. Wenn er *FALSE* ist, wird der 1A-Messbereich verwendet. Nicht kombinierbar mit dem Auto-Range-Modus.

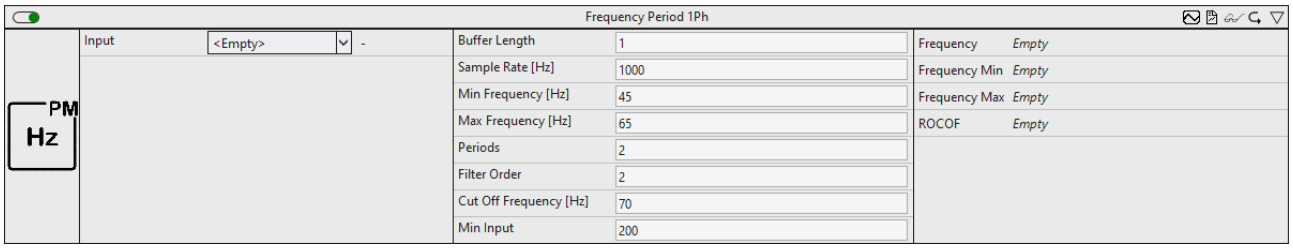
Ausgangswerte

- **Output Voltage 1 … Input Voltage 3:** Skalierte Spannungswerte der einzelnen Phasen.
- **Output Current 1 … Input Current 3:** Skalierte Stromwerte der einzelnen Phasen.

6.4.14.2.3.2 Einphasig

6.4.14.2.3.2.1 Basierend auf der Signalperiode

6.4.14.2.3.2.1.1 Frequency Period 1Ph



Das *Frequency Period*-Modul berechnet die Grundfrequenz des Eingangssignals. Dafür wird das Signal zunächst mit einem Butterworth-Tiefpassfilter gefiltert. Anschließend werden aus den gefilterten Werten die Nulldurchgänge des Eingangssignals ermittelt und aus deren Differenz die Frequenz berechnet. Die Ergebnisse beziehen sich, abhängig von der Konfiguration, auf eine oder mehrere Perioden. Die statistischen Ergebnisse beziehen sich auf die gesamte Laufzeit bzw. den Zeitpunkt, an dem zuletzt das Zurücksetzen der statistischen Ergebnisse erfolgt ist.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Frequency_Period_1Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Buffer Length:** Länge des Eingangspuffers.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Min Frequency [Hz]**: Minimal zu erwartende Messfrequenz.
- **Max Frequency [Hz]:** Maximal zu erwartende Messfrequenz.
- **Periods:** Anzahl an Perioden, die Einfluss auf die Berechnung haben. (Periodenlänge = Samplerate/ Frequenz).
- **Filter Order:** Ordnung des Tiefpassfilters. Bei der Einstellung muss die Stabilität des Filters berücksichtigt werden. Nur Werte bis zur zehnten Ordnung sind zulässig.
- **Cut Off Frequency [Hz]:** Grenzfrequenz des Tiefpassfilters.
- **Min Input:** Mindesteingangsgröße (RMS) über eine Periode. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.
- **ROCOF Avg Window Order:** Fenstergröße des gleitenden Mittelwertes für die Berechnung des ROCOF.

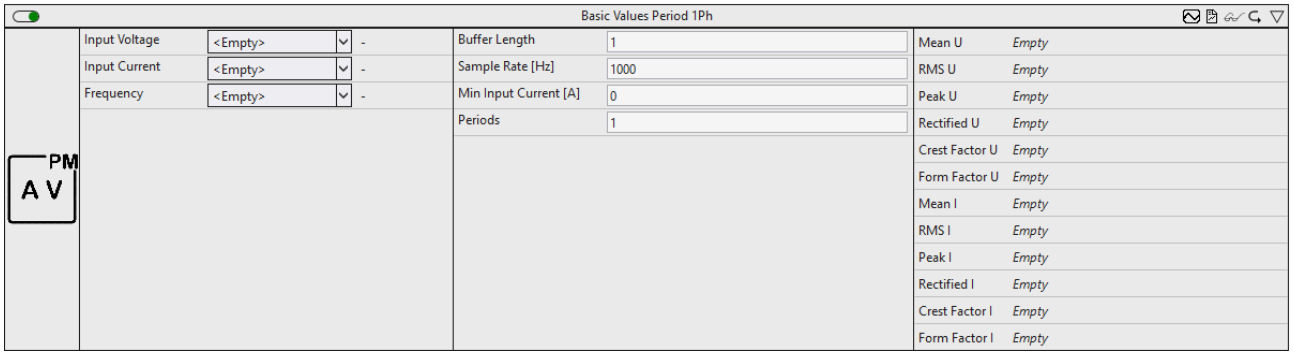
Eingangswerte

- **Input:** Eingangswerte von Strom oder Spannung.
- **Reset Statistics (optional):** *TRUE* setzt die Min- und Max-Werte der Ausgänge zurück.

Ausgangswerte

- **Frequency:** Frequenz, die durch zwei oder mehr Nulldurchgänge ermittelt wurde.
- **Frequency Min:** Kleinster aufgetretener Wert der berechneten Frequenz. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **Frequency Max:** Größter aufgetretener Wert der berechneten Frequenz. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **ROCOF:** Rate of change of frequency (ROCOF).

6.4.14.2.3.2.1.2 Basic Values Period 1Ph



Das *Basic Values Period*-Modul berechnet Analysewerte für den zeitlichen Signalverlauf von Strom und Spannung. Dazu gehören Mittelwert, RMS-Wert, Spitzenwert, Gleichrichtwert, Crest-Faktor sowie Formfaktor jeweils für Strom und Spannung. Die Ergebnisse beziehen sich auf eine konfigurierbare Anzahl von Signalperioden. Die Periodendauer bezieht sich auf die zu Periodenbeginn angegebene Frequenz am Eingang. Die optionalen statistischen Ergebnisse beziehen sich auf die gesamte Laufzeit bzw. den Zeitpunkt, an dem zuletzt das Zurücksetzen der statistischen Ergebnisse erfolgt ist.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_BasicValues_Period_1Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Buffer Length:** Länge des Eingangspuffers.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Min Input Current [A]:** Mindesteingangsgröße (RMS) des Stroms. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.
- **Periods:** Anzahl an Perioden, die Einfluss auf die Berechnung haben. (Periodenlänge = Samplerate/ Frequenz).

Eingangswerte

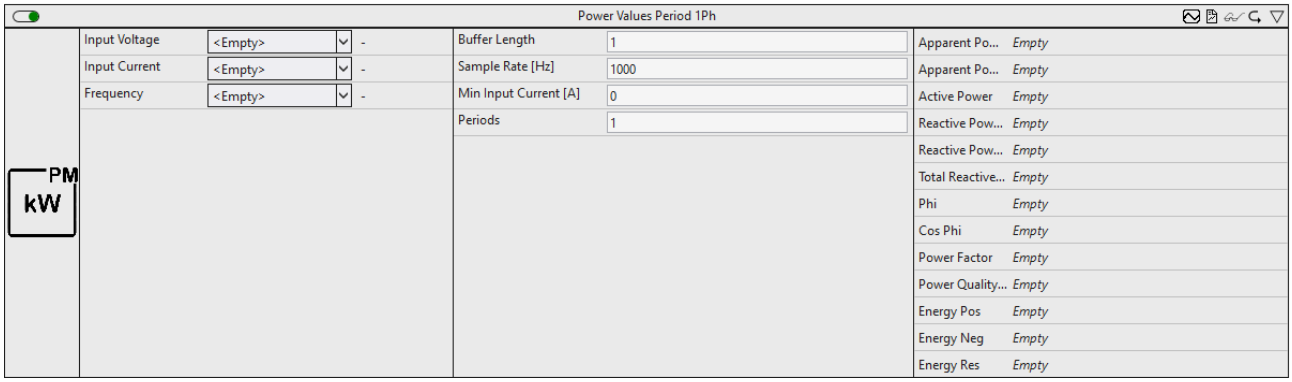
- **Frequency:** Aktuelle Frequenz des Eingangssignals. Wird verwendet, um zu Beginn einer Periode die Periodenlänge zu ermitteln. Der Ausgang des Moduls *Frequency Period1 Ch* kann verwendet werden.
- **Input Voltage:** Eingangswerte der Spannung.
- **Input Current:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Mean U:** Mittelwert der Spannung über n Perioden

- **RMS U:** Effektivwert der Spannung über n Perioden.
- **Peak U:** Spitzenwert der Spannung über n Perioden.
- **Rectified U:** Gleichrichtwert der Spannung über n Perioden.
- **Crest Factor U:** Scheitelfaktor (Crest-Faktor) der Spannung (Spitzenwert/Effektivwert).
- **Form Factor U:** Formfaktor der Spannung (Effektivwert/Gleichrichtwert).
- **Mean I:** Mittelwert des Stroms über n Perioden
- **RMS I:** Effektivwert des Stroms über n Perioden.
- **Peak I:** Spitzenwert des Stroms über n Perioden.
- **Rectified I:** Gleichrichtwert des Stroms über n Perioden.
- **Crest Factor I:** Scheitelfaktor (Crest-Faktor) des Stroms (Spitzenwert/Effektivwert).
- **Form Factor I:** Formfaktor des Stroms (Effektivwert/Gleichrichtwert).

6.4.14.2.3.2.1.3 Power Values Period 1Ph



Das *Power Values Period*-Modul berechnet die Leistungswerte des angeschlossenen Verbrauchers. Dazu gehören auch die Grundschwingungskomponenten und der Phasenverschiebungswinkel. Hierfür werden neben dem Signalverlauf im Zeitbereich nur die ersten Harmonischen des Eingangssignals zur Berechnung herangezogen. Der Vorteil dieser Algorithmen ist die hohe Dynamik der Berechnungen. Die Ergebnisse beziehen sich auf eine konfigurierbare Anzahl von Signalperioden. Die Periodendauer bezieht sich auf die zu Periodenbeginn angegebene Frequenz am Eingang. Die statistischen Ergebnisse beziehen sich auf die gesamte Laufzeit bzw. den Zeitpunkt, an dem zuletzt das Zurücksetzen der statistischen Ergebnisse erfolgt ist.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_PowerValues_Period_1Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Buffer Length:** Länge des Eingangspuffers.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Min Input Current [A]:** Mindesteingangsgröße (RMS) des Stroms. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.
- **Periods:** Anzahl an Perioden, die Einfluss auf die Berechnung haben. (Periodenlänge = Samplerate/ Frequenz).

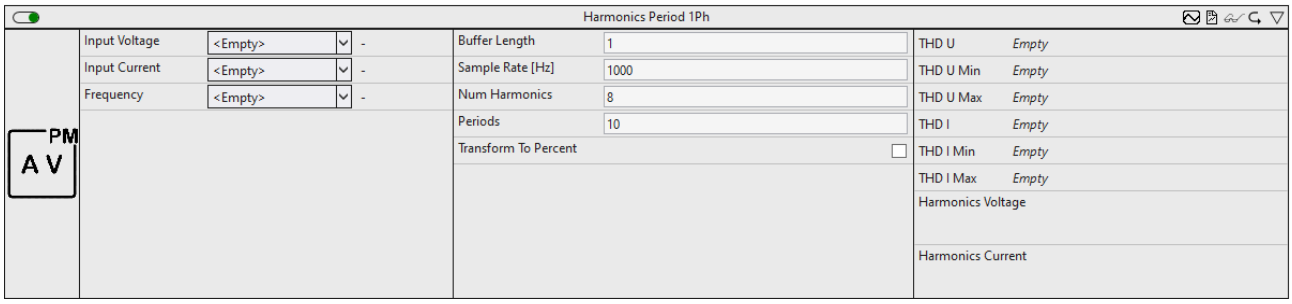
Eingangswerte

- **Frequency:** Aktuelle Frequenz des Eingangssignals. Wird verwendet, um zu Beginn einer Periode die Periodenlänge zu ermitteln. Der Ausgang des Moduls *Frequency Period1 Ch* kann verwendet werden.
- **Input Voltage:** Eingangswerte der Spannung.
- **Input Current:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Apparent Power:** Gesamt-Scheinleistung.
- **Apparent Power 1:** Grundschwingungs-Scheinleistung.
- **Active Power:** Wirkleistung.
- **Reactive Power d:** Verzerrungsblindleistung.
- **Reactive Power 1:** Grundschwingungs-Verschiebungs-Blindleistung.
- **Total Reactive Power:** Gesamt-Blindleistung.
- **Phi:** Phasenverschiebungswinkel.
- **Cos Phi:** CosPhi (Wirkleistung/Grundschwingungs-Scheinleistung).
- **Power Factor:** Leistungsfaktor (Wirkleistung/Gesamt-Scheinleistung).
- **Power Quality Factor:** Stellt die Qualität der Spannungsversorgung vereinfacht in einem Wertebereich zwischen 0 und 1 dar. Miteinbezogen wird die Frequenz, der Effektivwert der Spannung sowie der THD der Spannung.
- **Energy Pos:** Energie in positiver Richtung.
- **Energy Neg:** Energie in negativer Richtung.
- **Energy Res:** Resultierende Energie.

6.4.14.2.3.2.1.4 Harmonics Period 1Ph



Das *Harmonics Period*-Modul berechnet die Harmonischen von Strom und Spannung. Zusätzlich wird aus den Harmonischen der THD der Eingangsgrößen berechnet. Im Gegensatz zum Modul *Harmonics* beziehen sich die Ergebnisse auf eine konfigurierbare Anzahl von Signalperioden. Die Periodendauer bezieht sich auf die zu Periodenbeginn angegebene Frequenz am Eingang. Die statistischen Ergebnisse beziehen sich auf die gesamte Laufzeit bzw. den Zeitpunkt, an dem zuletzt das Zurücksetzen der statistischen Ergebnisse erfolgt ist.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Harmonics_Period_1Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Buffer Length:** Länge des Eingangspuffers.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Min Input Current [A]:** Mindesteingangsgröße (RMS) des Stroms. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.
- **Periods:** Anzahl an Perioden, die Einfluss auf die Berechnung haben. (Periodenlänge = Samplerate/Frequenz).
- **Num Harmonics:** Anzahl der zu berechnender Harmonischen.
- **Transform To Percent:** Boolescher Wert der angibt, ob das Ergebnis in Prozent ausgegeben werden soll. Dabei entspricht die erste Harmonische 100%.

Eingangswerte

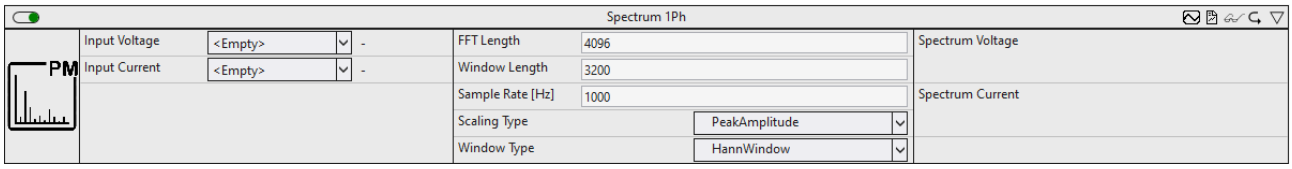
- **Frequency:** Aktuelle Frequenz des Eingangssignals. Wird verwendet, um zu Beginn einer Periode die Periodenlänge zu ermitteln. Der Ausgang des Moduls *Frequency Period1 Ch* kann verwendet werden.
- **Input Voltage:** Eingangswerte der Spannung.
- **Input Current:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **THD U:** THD der Spannung. Die Ausgabe erfolgt in Prozent.
- **THD U Min:** Kleinster aufgetretener Wert von *THD U*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD U Max:** Größter aufgetretener Wert von *THD U*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD I:** THD des Stroms. Die Ausgabe erfolgt in Prozent.
- **THD I Min:** Kleinster aufgetretener Wert von *THD I*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD I Max:** Größter aufgetretener Wert von *THD I*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **Harmonics Voltage:** Harmonische der Spannung.
- **Harmonics Current:** Harmonische des Stroms.

6.4.14.2.3.2.2 Basierend auf dem Frequenzbereich

6.4.14.2.3.2.2.1 Spectrum 1Ph



Das *Spectrum*-Modul berechnet die Betragsspektren der Strom- und Spannungswerte. Diese sind für eine Analyse der Eingangssignale im Frequenzbereich geeignet. Die Größe des Eingangspuffers entspricht der halben Fensterlänge.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Spectrum_1Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **FFT Length:** Länge der FFT. Sie muss größer als Eins und eine ganzzahlige Potenz von Zwei sein.
- **Window Length:** Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein. Sie darf nicht größer als *FFT Length* sein.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Scaling Type:** Ermöglicht eine Auswahl der verwendeten Skalierung, falls eine absolute Skalierung benötigt wird. Siehe: E_PMA_ScalingType
- **Window Type:** Definiert die verwendete Fensterfunktion. Ein guter Standardwert ist der Fenstertyp *HannWindow*. Die Fensterung kann durch die Verwendung des Fenstertyps *RectangularWindow* abgeschaltet werden. Siehe: E_PMA_WindowType
- **Transform To Decibel (optional):** Boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x → 20 * log10(x).

- **Decibel Threshold (optional):** Sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt, da der Logarithmus von Null mathematisch nicht definiert ist. Der kleinste mögliche Wert ist 3.75e-324.

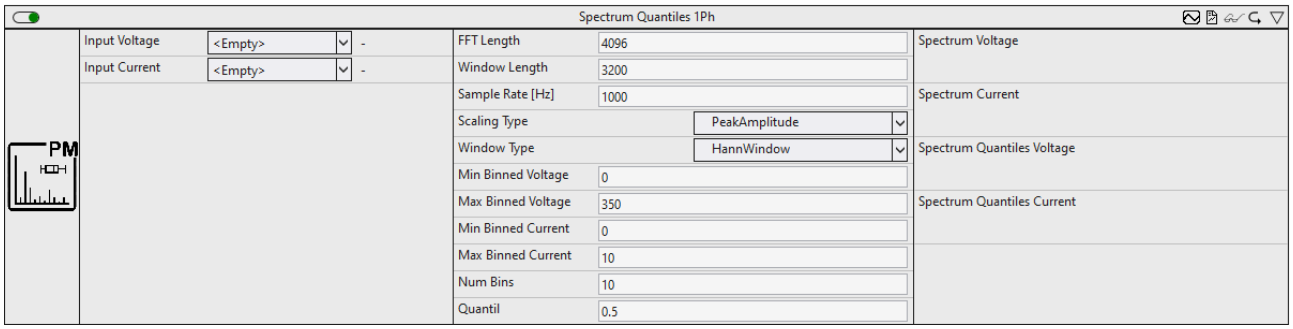
Eingangswerte

- **Input Voltage:** Eingangswerte der Spannung.
- **Input Current:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Spectrum Voltage:** Betragsspektrum der Spannungswerte.
- **Spectrum Current:** Betragsspektrum der Stromwerte.

6.4.14.2.3.2.2.2 Spectrum Quantiles 1Ph



Das *Spectrum Quantiles*-Modul berechnet die Betragsspektren der Strom- und Spannungswerte wie das *Spectrum*-Modul. Zusätzlich können p-Quantile der Verteilung des Spektrums berechnet werden.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Spectrum_Quantiles_1Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **FFT Length:** Länge der FFT. Sie muss größer als Eins und eine ganzzahlige Potenz von Zwei sein.
- **Window Length:** Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein. Sie darf nicht größer als *FFT Length* sein.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Scaling Type:** Ermöglicht eine Auswahl der verwendeten Skalierung, falls eine absolute Skalierung benötigt wird. Siehe: E_PMA_ScalingType
- **Window Type:** Definiert die verwendete Fensterfunktion. Ein guter Standardwert ist der Fenstertyp *HannWindow*. Die Fensterung kann durch die Verwendung des Fenstertyps *RectangularWindow* abgeschaltet werden. Siehe: E_PMA_WindowType
- **Transform To Decibel (optional):** Boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x → 20 * log10(x).
- **Decibel Threshold (optional):** Sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt, da der Logarithmus von Null mathematisch nicht definiert ist. Der kleinste mögliche Wert ist 3.75e-324.
- **Min Binned Voltage:** Unterer Grenzwert, für den die Ausgangsdaten der Spektrumsberechnung der Spannung in den regulären Histogramm-Bins gezählt werden.
- **Max Binned Voltage:** Oberer Grenzwert, für den die Ausgangsdaten der Spektrumsberechnung der Spannung in den regulären Histogramm-Bins gezählt werden. *Max Binned Voltage* muss größer sein als *Min Binned Voltage*.

- **Min Binned Current:** Unterer Grenzwert, für den die Ausgangsdaten der Spektrumsberechnung des Stroms in den regulären Histogramm-Bins gezählt werden.
- **Max Binned Current:** Oberer Grenzwert, für den die Ausgangsdaten der Spektrumsberechnung des Stroms in den regulären Histogramm-Bins gezählt werden. *Max Binned Current* muss größer sein als *Min Binned Current*.
- **Num Bins:** Anzahl der Bins eines Histogramms. Sie muss mindestens eins sein. In vielen Fällen sind Werte zwischen zehn und zwanzig eine sinnvolle Wahl. Die beiden speziellen Bins für Werte, die unterhalb von *Min Binned* bzw. oberhalb von *Max Binned* liegen sind in diesem Wert nicht mit enthalten.
- **Quantil:** Gibt die Quantilgrenze an. Sie muss zwischen 0.0 und 1.0 liegen. Beispielsweise entspricht 0.2 dem 20%-Quantil.

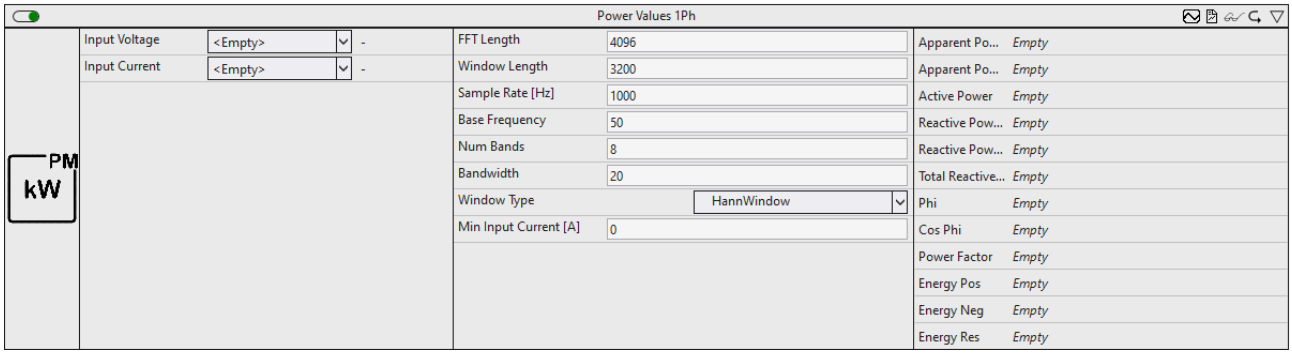
Eingangswerte

- **Input Voltage:** Eingangswerte der Spannung.
- **Input Current:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Spectrum Voltage:** Betragsspektrum der Spannungswerte.
- **Spectrum Current:** Betragsspektrum der Stromwerte.
- **Spectrum Quantiles Voltage:** P-Quantile der Verteilung von *Spectrum Voltage*
- **Spectrum Quantiles Current:** P-Quantile der Verteilung von *Sprectrum Current*

6.4.14.2.3.2.2.3 Power Values 1Ph



Das *Power Values*-Modul berechnet die Leistungswerte des angeschlossenen Verbrauchers. Dazu gehören auch die Grundschwingungskomponenten und der Phasenverschiebungswinkel. Intern werden die einzelnen Harmonischen und deren Phasenlage für die Berechnungen ermittelt.

Alternativ kann das *Power Values Period*-Modul für die Berechnung eingesetzt werden. Dieser verwendet einfachere Berechnungsverfahren für eine erhöhte Dynamik.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_PowerValues_1Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **FFT Length:** Länge der FFT. Sie muss größer als Eins und eine ganzzahlige Potenz von Zwei sein.
- **Window Length:** Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein. Sie darf nicht größer als *FFT Length* sein.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Base Frequency**: Frequenz der ersten Harmonischen (Grundschwingung) in Hz.
- **Num Bands**: Gibt die Anzahl an Bändern an, für die der RMS berechnet wird.

- **Bandwidth**: Gesamte Bandweite eines einzelnen RMS-Bandes.
- **Window Type:** Definiert die verwendete Fensterfunktion. Ein guter Standardwert ist der Fenstertyp *HannWindow*. Die Fensterung kann durch die Verwendung des Fenstertyps *RectangularWindow* abgeschaltet werden. Siehe: E_PMA_WindowType
- **Min Input Current [A]:** Mindesteingangsgröße (RMS) des Stroms. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.

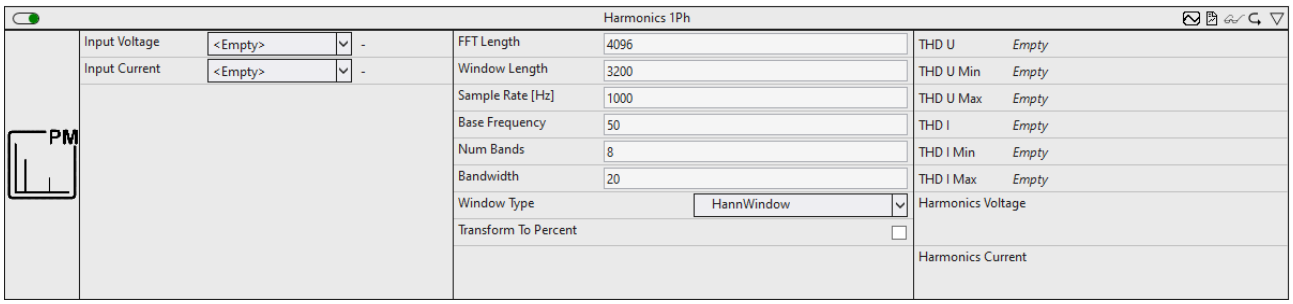
Eingangswerte

- **Input Voltage:** Eingangswerte der Spannung.
- **Input Current:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Apparent Power:** Gesamt-Scheinleistung.
- **Apparent Power 1:** Grundschwingungs-Scheinleistung.
- **Active Power:** Wirkleistung.
- **Reactive Power d:** Verzerrungsblindleistung.
- **Reactive Power 1:** Grundschwingungs-Verschiebungs-Blindleistung.
- **Total Reactive Power:** Gesamt-Blindleistung.
- **Phi:** Phasenverschiebungswinkel.
- **Cos Phi:** CosPhi (Wirkleistung/Grundschwingungs-Scheinleistung).
- **Power Factor:** Leistungsfaktor (Wirkleistung/Gesamt-Scheinleistung).
- **Power Quality Factor:** Stellt die Qualität der Spannungsversorgung vereinfacht in einem Wertebereich zwischen 0 und 1 dar. Miteinbezogen wird die Frequenz, der Effektivwert der Spannung sowie der THD der Spannung.
- **Energy Pos:** Energie in positiver Richtung.
- **Energy Neg:** Energie in negativer Richtung.
- **Energy Res:** Resultierende Energie.

6.4.14.2.3.2.2.4 Harmonics 1Ph



Das *Harmonics*-Modul berechnet die Harmonischen von Strom und Spannung. Zusätzlich wird aus den Harmonischen der THD der Eingangsgrößen berechnet.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Harmonics_1Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Num Harmonics:** Anzahl der zu berechnender Harmonischen.
- **Transform To Percent:** Boolescher Wert der angibt, ob das Ergebnis in Prozent ausgegeben werden soll. Dabei entspricht die erste Harmonische 100%.

- **FFT Length:** Länge der FFT. Sie muss größer als Eins und eine ganzzahlige Potenz von Zwei sein.
- **Window Length:** Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein. Sie darf nicht größer als *FFT Length* sein.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Base Frequency**: Frequenz der ersten Harmonischen (Grundschwingung) in Hz.
- **Num Bands**: Gibt die Anzahl an Bändern an, für die der RMS berechnet wird.
- **Bandwidth**: Gesamte Bandweite eines einzelnen RMS-Bandes.
- **Window Type:** Definiert die verwendete Fensterfunktion. Ein guter Standardwert ist der Fenstertyp *HannWindow*. Die Fensterung kann durch die Verwendung des Fenstertyps *RectangularWindow* abgeschaltet werden. Siehe: E_PMA_WindowType
- **Transform To Percent:** Boolescher Wert, der angibt, ob das Ergebnis in Prozent ausgegeben werden soll. Dabei entspricht die erste Harmonische 100%.

Eingangswerte

- **Input Voltage:** Eingangswerte der Spannung.
- **Input Current:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

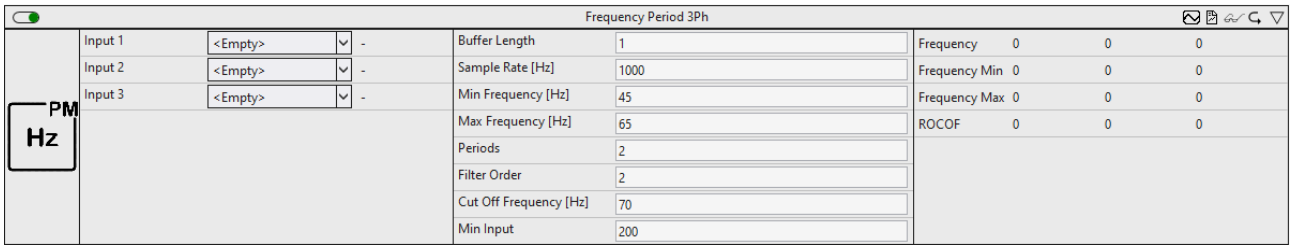
Ausgangswerte

- **THD U:** THD der Spannung. Die Ausgabe erfolgt in Prozent.
- **THD U Min:** Kleinster aufgetretener Wert von *THD U*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD U Max:** Größter aufgetretener Wert von *THD U*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD I:** THD des Stroms. Die Ausgabe erfolgt in Prozent.
- **THD I Min:** Kleinster aufgetretener Wert von *THD I*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD I Max:** Größter aufgetretener Wert von *THD I*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **Harmonics Voltage:** Harmonische der Spannung.
- **Harmonics Current:** Harmonische des Stroms.

6.4.14.2.3.3 Dreiphasig

6.4.14.2.3.3.1 Basierend auf der Signalperiode

6.4.14.2.3.3.1.1 Frequency Period 3Ph



Das *Frequency Period*-Modul berechnet die Grundfrequenz des Eingangssignals. Dafür wird das Signal zunächst mit einem Butterworth-Tiefpassfilter gefiltert. Anschließend werden aus den gefilterten Werten die Nulldurchgänge des Eingangssignals ermittelt und aus deren Differenz die Frequenz berechnet. Die Ergebnisse beziehen sich, abhängig von der Konfiguration, auf eine oder mehrere Perioden. Die statistischen Ergebnisse beziehen sich auf die gesamte Laufzeit bzw. den Zeitpunkt, an dem zuletzt das Zurücksetzen der statistischen Ergebnisse erfolgt ist.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Frequency_Period_3Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Buffer Length:** Länge des Eingangspuffers.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Min Frequency [Hz]**: Minimal zu erwartende Messfrequenz.
- **Max Frequency [Hz]:** Maximal zu erwartende Messfrequenz.
- **Periods:** Anzahl an Perioden, die Einfluss auf die Berechnung haben. (Periodenlänge = Samplerate/ Frequenz).
- **Filter Order:** Ordnung des Tiefpassfilters. Bei der Einstellung muss die Stabilität des Filters berücksichtigt werden. Nur Werte bis zur zehnten Ordnung sind zulässig.
- **Cut Off Frequency [Hz]:** Grenzfrequenz des Tiefpassfilters.
- **Min Input:** Mindesteingangsgröße (RMS) über eine Periode. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.
- **ROCOF Avg Window Order:** Fenstergröße des gleitenden Mittelwertes für die Berechnung des ROCOF.

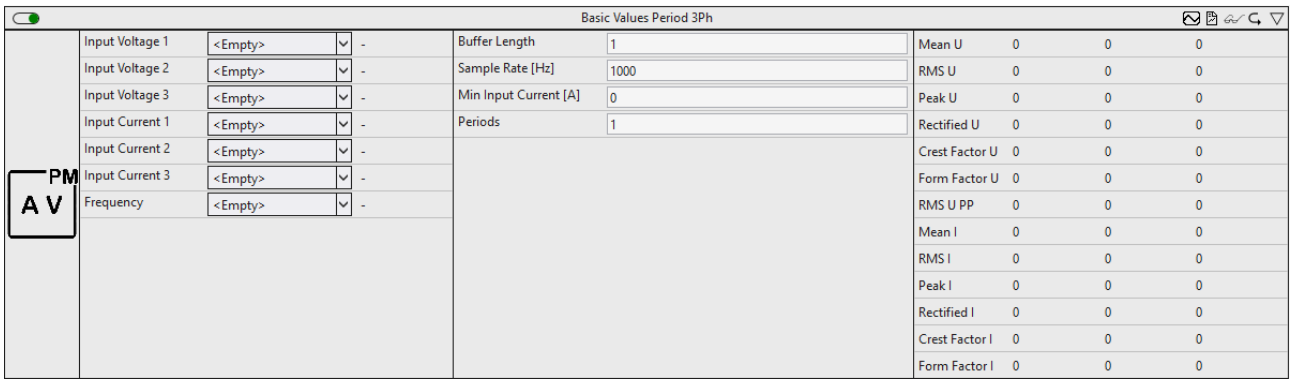
Eingangswerte

- **Input 1 .. Input 3:** Eingangswerte von Strom oder Spannung.
- **Reset Statistics (optional):** *TRUE* setzt die Min- und Max-Werte der Ausgänge zurück.

Ausgangswerte

- **Frequency:** Frequenz, die durch zwei oder mehr Nulldurchgänge ermittelt wurde.
- **Frequency Min:** Kleinster aufgetretener Wert der berechneten Frequenz. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **Frequency Max:** Größter aufgetretener Wert der berechneten Frequenz. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **ROCOF:** Rate of change of frequency (ROCOF).

6.4.14.2.3.3.1.2 Basic Values Period 3Ph



Das *Basic Values Period*-Modul berechnet Analysewerte für den zeitlichen Signalverlauf von Strom und Spannung. Dazu gehören Mittelwert, RMS-Wert, Spitzenwert, Gleichrichtwert, Crest-Faktor sowie Formfaktor jeweils für Strom und Spannung. Die Ergebnisse beziehen sich auf eine konfigurierbare Anzahl von Signalperioden. Die Periodendauer bezieht sich auf die zu Periodenbeginn angegebene Frequenz am Eingang. Die optionalen statistischen Ergebnisse beziehen sich auf die gesamte Laufzeit bzw. den Zeitpunkt, an dem zuletzt das Zurücksetzen der statistischen Ergebnisse erfolgt ist.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_BasicValues_Period_3Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Buffer Length:** Länge des Eingangspuffers.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Min Input Current [A]:** Mindesteingangsgröße (RMS) des Stroms. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.
- **Periods:** Anzahl an Perioden, die Einfluss auf die Berechnung haben. (Periodenlänge = Samplerate/ Frequenz).

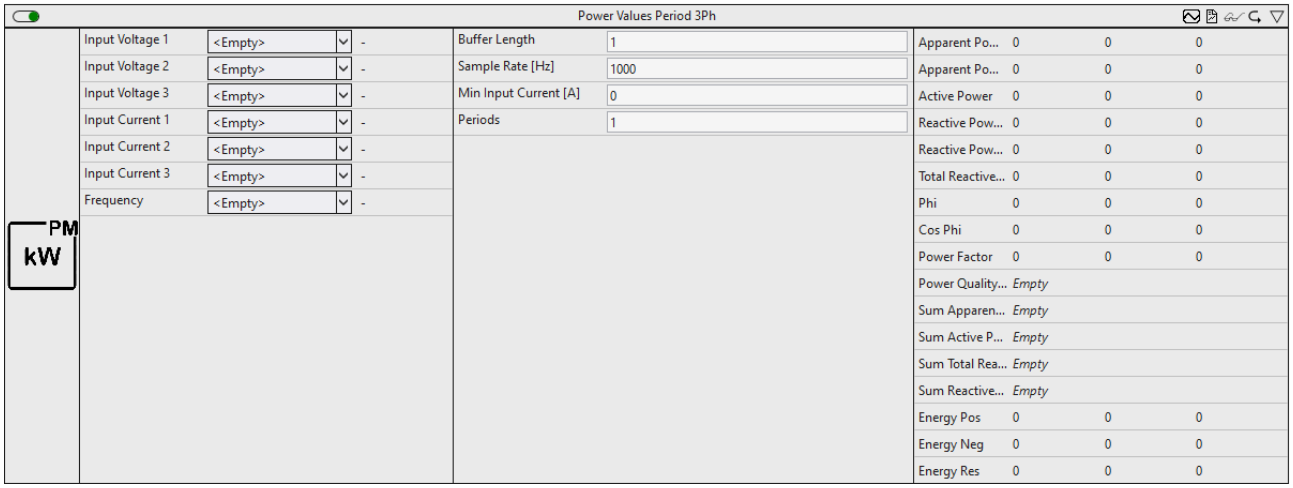
Eingangswerte

- **Frequency:** Aktuelle Frequenz des Eingangssignals. Wird verwendet, um zu Beginn einer Periode die Periodenlänge zu ermitteln. Der Ausgang des Moduls *Frequency Period1 Ch* kann verwendet werden.
- **Input Voltage 1 .. Input Voltage 3:** Eingangswerte der Spannung.
- **Input Current 1 .. Input Current 3:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Mean U:** Mittelwert der Spannung über n Perioden
- **RMS U:** Effektivwert der Spannung über n Perioden.
- **Peak U:** Spitzenwert der Spannung über n Perioden.
- **Rectified U:** Gleichrichtwert der Spannung über n Perioden.
- **Crest Factor U:** Scheitelfaktor (Crest-Faktor) der Spannung (Spitzenwert/Effektivwert).
- **Form Factor U:** Formfaktor der Spannung (Effektivwert/Gleichrichtwert).
- **Mean I:** Mittelwert des Stroms über n Perioden
- **RMS I:** Effektivwert des Stroms über n Perioden.
- **Peak I:** Spitzenwert des Stroms über n Perioden.
- **Rectified I:** Gleichrichtwert des Stroms über n Perioden.
- **Crest Factor I:** Scheitelfaktor (Crest-Faktor) des Stroms (Spitzenwert/Effektivwert).
- **Form Factor I:** Formfaktor des Stroms (Effektivwert/Gleichrichtwert).

6.4.14.2.3.3.1.3 Power Values Period 3Ph



Das *Power Values Period*-Modul berechnet die Leistungswerte des angeschlossenen Verbrauchers. Dazu gehören auch die Grundschwingungskomponenten und der Phasenverschiebungswinkel. Hierfür werden neben dem Signalverlauf im Zeitbereich nur die ersten Harmonischen des Eingangssignals zur Berechnung herangezogen. Der Vorteil dieser Algorithmen ist die hohe Dynamik der Berechnungen. Die Ergebnisse beziehen sich auf eine konfigurierbare Anzahl von Signalperioden. Die Periodendauer bezieht sich auf die zu Periodenbeginn angegebene Frequenz am Eingang. Die statistischen Ergebnisse beziehen sich auf die gesamte Laufzeit bzw. den Zeitpunkt, an dem zuletzt das Zurücksetzen der statistischen Ergebnisse erfolgt ist.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_PowerValues_Period_3Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Buffer Length:** Länge des Eingangspuffers.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Min Input Current [A]:** Mindesteingangsgröße (RMS) des Stroms. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.
- **Periods:** Anzahl an Perioden, die Einfluss auf die Berechnung haben. (Periodenlänge = Samplerate/ Frequenz).

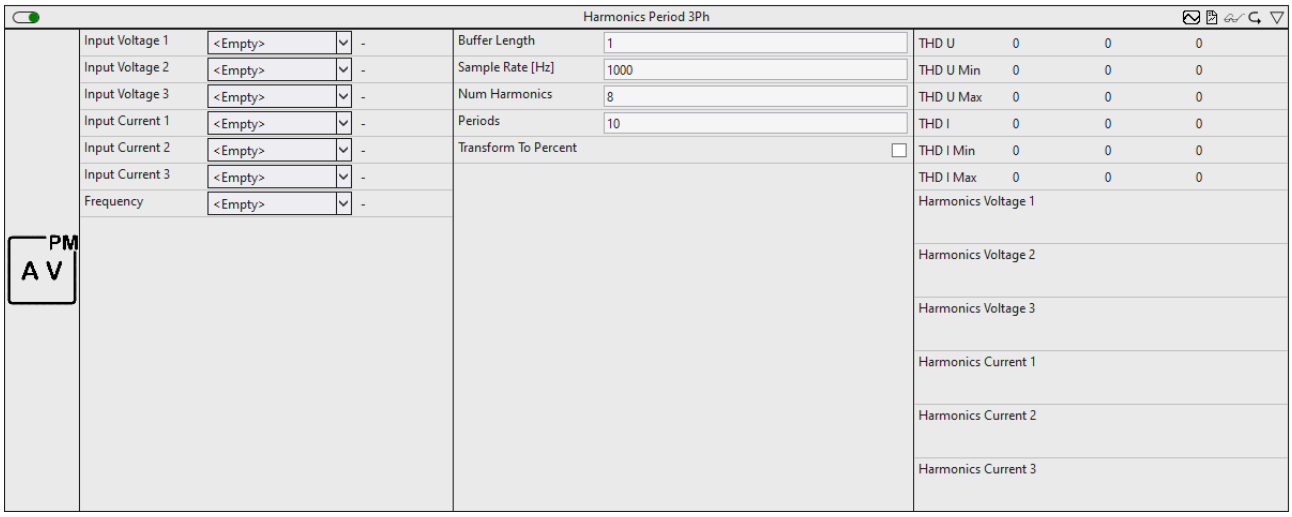
Eingangswerte

- **Frequency:** Aktuelle Frequenz des Eingangssignals. Wird verwendet, um zu Beginn einer Periode die Periodenlänge zu ermitteln. Der Ausgang des Moduls *Frequency Period1 Ch* kann verwendet werden.
- **Input Voltage 1 .. Input Voltage 3:** Eingangswerte der Spannung.
- **Input Current 1 .. Input Current 3:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Apparent Power:** Gesamt-Scheinleistung.
- **Apparent Power 1:** Grundschwingungs-Scheinleistung.
- **Active Power:** Wirkleistung.
- **Reactive Power d:** Verzerrungsblindleistung.
- **Reactive Power 1:** Grundschwingungs-Verschiebungs-Blindleistung.
- **Total Reactive Power:** Gesamt-Blindleistung.
- **Phi:** Phasenverschiebungswinkel.
- **Cos Phi:** CosPhi (Wirkleistung/Grundschwingungs-Scheinleistung).
- **Power Factor:** Leistungsfaktor (Wirkleistung/Gesamt-Scheinleistung).
- **Power Quality Factor:** Stellt die Qualität der Spannungsversorgung vereinfacht in einem Wertebereich zwischen 0 und 1 dar. Miteinbezogen wird die Frequenz, der Effektivwert der Spannung sowie der THD der Spannung.
- **Energy Pos:** Energie in positiver Richtung.
- **Energy Neg:** Energie in negativer Richtung.
- **Energy Res:** Resultierende Energie.
- **Sum Apparent Power:** Summe der Gesamt-Scheinleistung aller Phasen.
- **Sum Active Power:** Summe der Wirkleistung aller Phasen.
- **Sum Total Reactive Power:** Summe der Gesamt- Blindleistung aller Phasen.
- **Sum Reactive Power 1 Res:** Summe der Beträge der Grundschwingungs-Verschiebungs-Blindleistung aller Phasen.

6.4.14.2.3.3.1.4 Harmonics Period 3Ph



Das *Harmonics Period*-Modul berechnet die Harmonischen von Strom und Spannung. Zusätzlich wird aus den Harmonischen der THD der Eingangsgrößen berechnet. Im Gegensatz zum Modul *Harmonics* beziehen sich die Ergebnisse auf eine konfigurierbare Anzahl von Signalperioden. Die Periodendauer bezieht sich auf die zu Periodenbeginn angegebene Frequenz am Eingang. Die statistischen Ergebnisse beziehen sich auf die gesamte Laufzeit bzw. den Zeitpunkt, an dem zuletzt das Zurücksetzen der statistischen Ergebnisse erfolgt ist.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Harmonics_Period_3Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Buffer Length:** Länge des Eingangspuffers.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Min Input Current [A]:** Mindesteingangsgröße (RMS) des Stroms. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.
- **Periods:** Anzahl an Perioden, die Einfluss auf die Berechnung haben. (Periodenlänge = Samplerate/ Frequenz).
- **Num Harmonics:** Anzahl der zu berechnender Harmonischen.
- **Transform To Percent:** Boolescher Wert der angibt, ob das Ergebnis in Prozent ausgegeben werden soll. Dabei entspricht die erste Harmonische 100%.

Eingangswerte

- **Frequency:** Aktuelle Frequenz des Eingangssignals. Wird verwendet, um zu Beginn einer Periode die Periodenlänge zu ermitteln. Der Ausgang des Moduls *Frequency Period1 Ch* kann verwendet werden.
- **Input Voltage 1 .. Input Voltage 3:** Eingangswerte der Spannung.
- **Input Current 1 .. Input Current 3:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

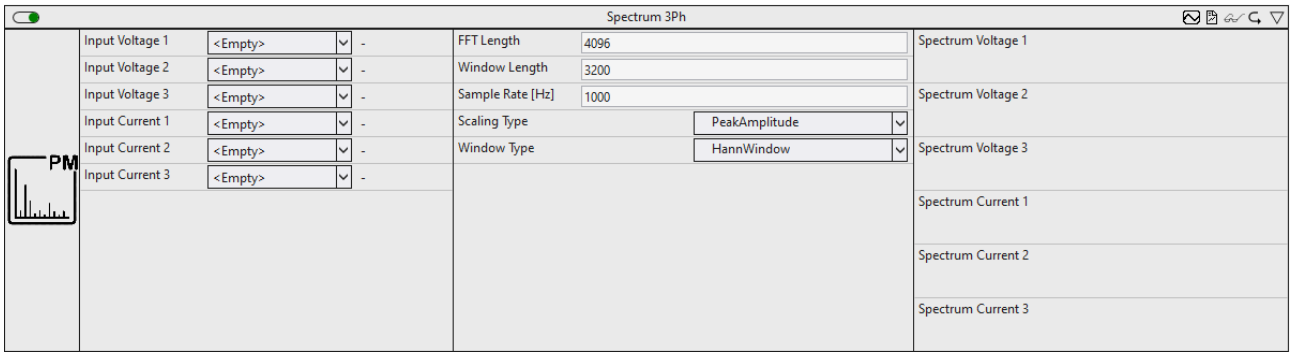
Ausgangswerte

- **THD U:** THD der Spannung. Die Ausgabe erfolgt in Prozent.
- **THD U Min:** Kleinster aufgetretener Wert von *THD U*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD U Max:** Größter aufgetretener Wert von *THD U*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD I:** THD des Stroms. Die Ausgabe erfolgt in Prozent.

- **THD I Min:** Kleinster aufgetretener Wert von *THD I*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD I Max:** Größter aufgetretener Wert von *THD I*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **Harmonics Voltage:** Harmonische der Spannung.
- **Harmonics Current:** Harmonische des Stroms.

6.4.14.2.3.3.2 Basierend auf dem Frequenzbereich

6.4.14.2.3.3.2.1 Spectrum 3Ph



Das *Spectrum*-Modul berechnet die Betragsspektren der Strom- und Spannungswerte. Diese sind für eine Analyse der Eingangssignale im Frequenzbereich geeignet. Die Größe des Eingangspuffers entspricht der halben Fensterlänge.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Spectrum_3Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **FFT Length:** Länge der FFT. Sie muss größer als Eins und eine ganzzahlige Potenz von Zwei sein.
- **Window Length:** Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein. Sie darf nicht größer als *FFT Length* sein.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Scaling Type:** Ermöglicht eine Auswahl der verwendeten Skalierung, falls eine absolute Skalierung benötigt wird. Siehe: E_PMA_ScalingType
- **Window Type:** Definiert die verwendete Fensterfunktion. Ein guter Standardwert ist der Fenstertyp *HannWindow*. Die Fensterung kann durch die Verwendung des Fenstertyps *RectangularWindow* abgeschaltet werden. Siehe: E_PMA_WindowType
- **Transform To Decibel (optional):** Boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x → 20 * log10(x).
- **Decibel Threshold (optional):** Sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt, da der Logarithmus von Null mathematisch nicht definiert ist. Der kleinste mögliche Wert ist 3.75e-324.

Eingangswerte

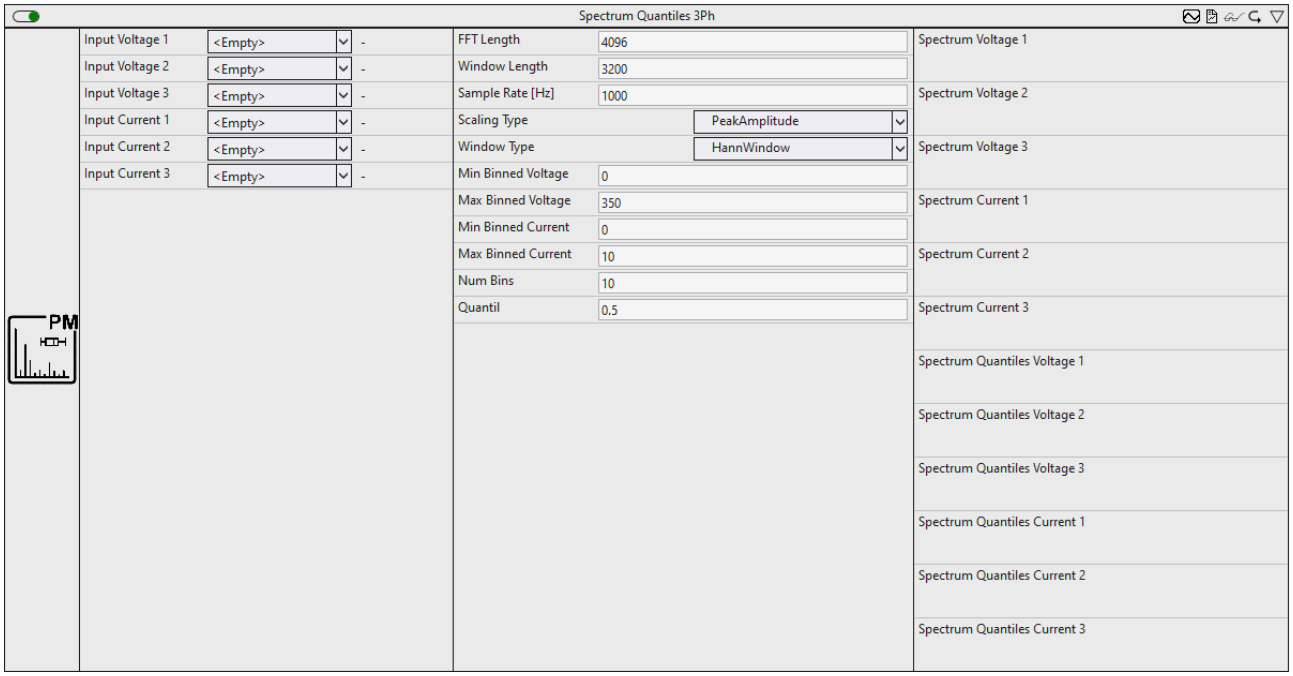
- **Input Voltage 1 .. Input Voltage 3:** Eingangswerte der Spannung.
- **Input Current 1 .. Input Current 3:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Spectrum Voltage 1 .. Spectrum Voltage 3:** Betragsspektrum der Spannungswerte.

- **Spectrum Current 1 .. Spectrum Current 3:** Betragsspektrum der Stromwerte.

6.4.14.2.3.3.2.2 Spectrum Quantiles 3Ph



Das *Spectrum Quantiles*-Modul berechnet die Betragsspektren der Strom- und Spannungswerte wie das *Spectrum*-Modul. Zusätzlich können p-Quantile der Verteilung des Spektrums berechnet werden.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Spectrum_Quantiles_3Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **FFT Length:** Länge der FFT. Sie muss größer als Eins und eine ganzzahlige Potenz von Zwei sein.
- **Window Length:** Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein. Sie darf nicht größer als *FFT Length* sein.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Scaling Type:** Ermöglicht eine Auswahl der verwendeten Skalierung, falls eine absolute Skalierung benötigt wird. Siehe: E_PMA_ScalingType
- **Window Type:** Definiert die verwendete Fensterfunktion. Ein guter Standardwert ist der Fenstertyp *HannWindow*. Die Fensterung kann durch die Verwendung des Fenstertyps *RectangularWindow* abgeschaltet werden. Siehe: E_PMA_WindowType
- **Transform To Decibel (optional):** Boolescher Wert, der angibt, ob das Ergebnis der FFT in die Dezibel-Skala transformiert werden soll, entsprechend der Transformation x → 20 * log10(x).
- **Decibel Threshold (optional):** Sehr kleiner Fließkommawert größer als Null. Werte, die kleiner als diese Zahl sind, werden vor einer Transformation in die Dezibel-Skala durch diesen Wert ersetzt, da der Logarithmus von Null mathematisch nicht definiert ist. Der kleinste mögliche Wert ist 3.75e-324.
- **Min Binned Voltage:** Unterer Grenzwert, für den die Ausgangsdaten der Spektrumsberechnung der Spannung in den regulären Histogramm-Bins gezählt werden.
- **Max Binned Voltage:** Oberer Grenzwert, für den die Ausgangsdaten der Spektrumsberechnung der Spannung in den regulären Histogramm-Bins gezählt werden. *Max Binned Voltage* muss größer sein als *Min Binned Voltage*.
- **Min Binned Current:** Unterer Grenzwert, für den die Ausgangsdaten der Spektrumsberechnung des Stroms in den regulären Histogramm-Bins gezählt werden.

- **Max Binned Current:** Oberer Grenzwert, für den die Ausgangsdaten der Spektrumsberechnung des Stroms in den regulären Histogramm-Bins gezählt werden. *Max Binned Current* muss größer sein als *Min Binned Current*.
- **Num Bins:** Anzahl der Bins eines Histogramms. Sie muss mindestens eins sein. In vielen Fällen sind Werte zwischen zehn und zwanzig eine sinnvolle Wahl. Die beiden speziellen Bins für Werte, die unterhalb von *Min Binned* bzw. oberhalb von *Max Binned* liegen sind in diesem Wert nicht mit enthalten.
- **Quantil:** Gibt die Quantilgrenze an. Sie muss zwischen 0.0 und 1.0 liegen. Beispielsweise entspricht 0.2 dem 20%-Quantil.

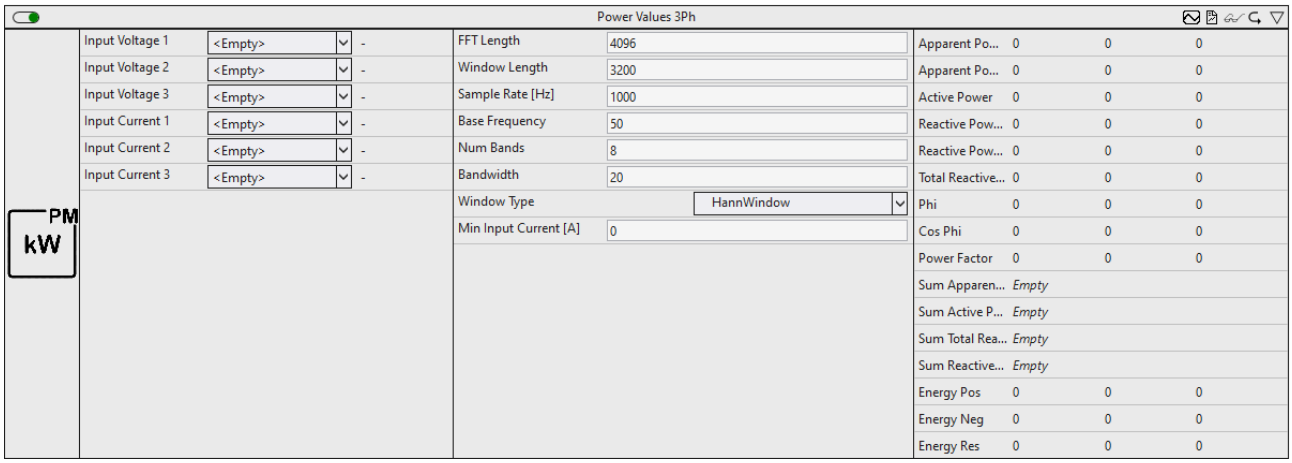
Eingangswerte

- **Input Voltage 1 .. Input Voltage 3:** Eingangswerte der Spannung.
- **Input Current 1 .. Input Current 3:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Spectrum Voltage 1 .. Spectrum Voltage 3:** Betragsspektrum der Spannungswerte.
- **Spectrum Current 1 .. Spectrum Current 3:** Betragsspektrum der Stromwerte
- **Spectrum Quantiles Voltage 1 .. Spectrum Quantiles Voltage 3:** P-Quantile der Verteilung von *Spectrum Voltage*
- **Spectrum Quantiles Current 1 .. Spectrum Quantiles Current 3:** P-Quantile der Verteilung von *Sprectrum Current*

6.4.14.2.3.3.2.3 Power Values 3Ph



Das *Power Values*-Modul berechnet die Leistungswerte des angeschlossenen Verbrauchers. Dazu gehören auch die Grundschwingungskomponenten und der Phasenverschiebungswinkel. Intern werden die einzelnen Harmonischen und deren Phasenlage für die Berechnungen ermittelt.

Alternativ kann das *Power Values Period*-Modul für die Berechnung eingesetzt werden. Dieser verwendet einfachere Berechnungsverfahren für eine erhöhte Dynamik.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_PowerValues_3Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **FFT Length:** Länge der FFT. Sie muss größer als Eins und eine ganzzahlige Potenz von Zwei sein.
- **Window Length:** Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein. Sie darf nicht größer als *FFT Length* sein.

- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Base Frequency**: Frequenz der ersten Harmonischen (Grundschwingung) in Hz.
- **Num Bands**: Gibt die Anzahl an Bändern an, für die der RMS berechnet wird.
- **Bandwidth**: Gesamte Bandweite eines einzelnen RMS-Bandes.
- **Window Type:** Definiert die verwendete Fensterfunktion. Ein guter Standardwert ist der Fenstertyp *HannWindow*. Die Fensterung kann durch die Verwendung des Fenstertyps *RectangularWindow* abgeschaltet werden. Siehe: E_PMA_WindowType
- **Min Input Current [A]:** Mindesteingangsgröße (RMS) des Stroms. Dieser Parameter verhindert die Berechnung bei zu kleinen Eingangswerten.

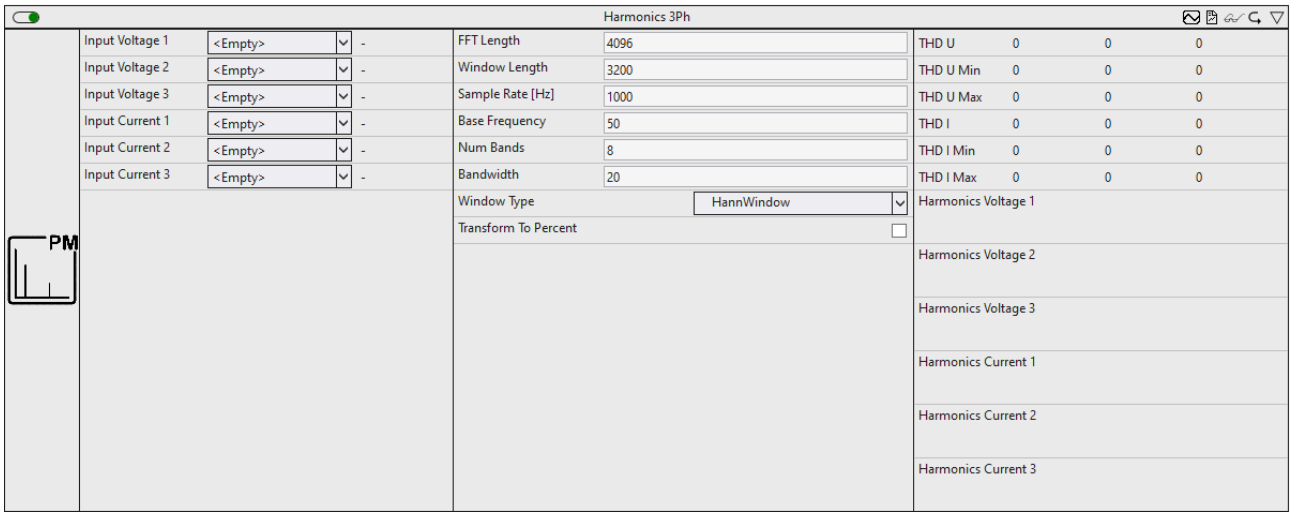
Eingangswerte

- **Input Voltage 1 .. Input Voltage 3:** Eingangswerte der Spannung.
- **Input Current 1 .. Input Current 3:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **Apparent Power:** Gesamt-Scheinleistung.
- **Apparent Power 1:** Grundschwingungs-Scheinleistung.
- **Active Power:** Wirkleistung.
- **Reactive Power d:** Verzerrungsblindleistung.
- **Reactive Power 1:** Grundschwingungs-Verschiebungs-Blindleistung.
- **Total Reactive Power:** Gesamt-Blindleistung.
- **Phi:** Phasenverschiebungswinkel.
- **Cos Phi:** CosPhi (Wirkleistung/Grundschwingungs-Scheinleistung).
- **Power Factor:** Leistungsfaktor (Wirkleistung/Gesamt-Scheinleistung).
- **Power Quality Factor:** Stellt die Qualität der Spannungsversorgung vereinfacht in einem Wertebereich zwischen 0 und 1 dar. Miteinbezogen wird die Frequenz, der Effektivwert der Spannung sowie der THD der Spannung.
- **Energy Pos:** Energie in positiver Richtung.
- **Energy Neg:** Energie in negativer Richtung.
- **Energy Res:** Resultierende Energie.
- **Sum Apparent Power:** Summe der Gesamt-Scheinleistung aller Phasen.
- **Sum Active Power:** Summe der Wirkleistung aller Phasen.
- **Sum Total Reactive Power:** Summe der Gesamt- Blindleistung aller Phasen.
- **Sum Reactive Power 1 Res:** Summe der Beträge der Grundschwingungs-Verschiebungs-Blindleistung aller Phasen.

6.4.14.2.3.3.2.4 Harmonics 3Ph



Das *Harmonics*-Modul berechnet die Harmonischen von Strom und Spannung. Zusätzlich wird aus den Harmonischen der THD der Eingangsgrößen berechnet.

Im Beckhoff Information System finden Sie die Dokumentation des zugehörigen SPS-Funktionsbausteins FB_PMA_Harmonics_3Ph.

Im Beckhoff Information System finden Sie die Dokumentation der Function TF3650 TwinCAT 3 Power Monitoring mit Beschreibungen der zugehörigen SPS-Bibliothek.

Konfigurationsparameter

- **Num Harmonics:** Anzahl der zu berechnender Harmonischen.
- **Transform To Percent:** Boolescher Wert der angibt, ob das Ergebnis in Prozent ausgegeben werden soll. Dabei entspricht die erste Harmonische 100%.
- **FFT Length:** Länge der FFT. Sie muss größer als Eins und eine ganzzahlige Potenz von Zwei sein.
- **Window Length:** Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein. Sie darf nicht größer als *FFT Length* sein.
- **Sample Rate [Hz]:** Abtastrate (Samples pro Sekunde) des Eingangssignals.
- **Base Frequency**: Frequenz der ersten Harmonischen (Grundschwingung) in Hz.
- **Num Bands**: Gibt die Anzahl an Bändern an, für die der RMS berechnet wird.
- **Bandwidth**: Gesamte Bandweite eines einzelnen RMS-Bandes.
- **Window Type:** Definiert die verwendete Fensterfunktion. Ein guter Standardwert ist der Fenstertyp *HannWindow*. Die Fensterung kann durch die Verwendung des Fenstertyps *RectangularWindow* abgeschaltet werden. Siehe: E_PMA_WindowType
- **Transform To Percent:** Boolescher Wert, der angibt, ob das Ergebnis in Prozent ausgegeben werden soll. Dabei entspricht die erste Harmonische 100%.

Eingangswerte

- **Input Voltage 1 .. Input Voltage 3:** Eingangswerte der Spannung.
- **Input Current 1 .. Input Current 3:** Eingangswerte des Stroms.
- **Reset Statistics (optional):** `TRUE` setzt die Min- und Max--Werte der Ausgänge zurück.

Ausgangswerte

- **THD U:** THD der Spannung. Die Ausgabe erfolgt in Prozent.
- **THD U Min:** Kleinster aufgetretener Wert von *THD U*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD U Max:** Größter aufgetretener Wert von *THD U*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.

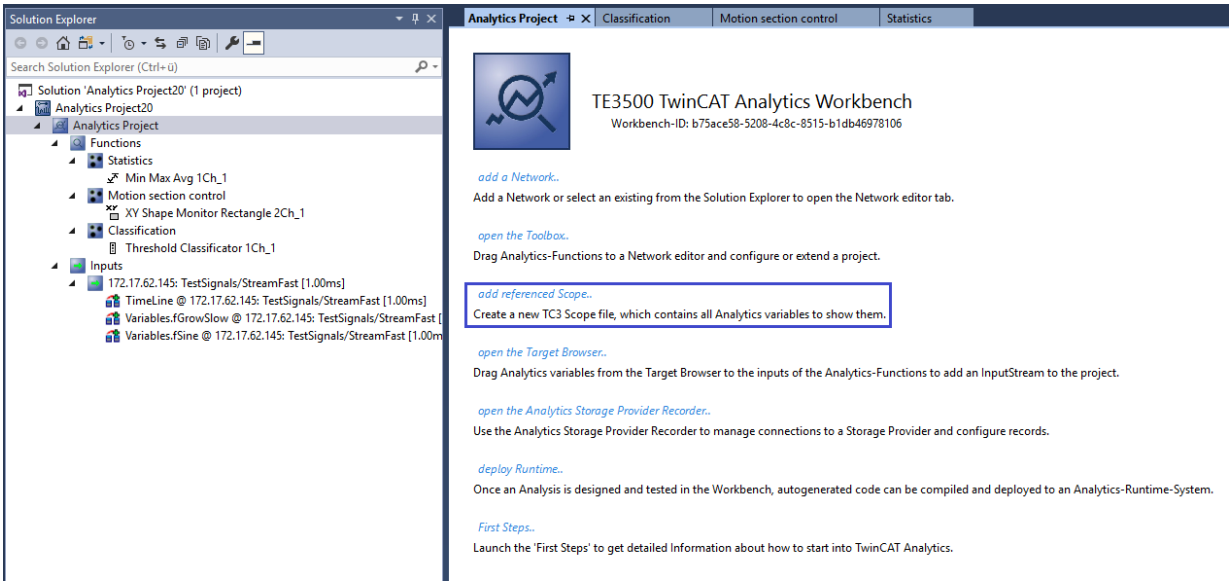
- **THD I:** THD des Stroms. Die Ausgabe erfolgt in Prozent.
- **THD I Min:** Kleinster aufgetretener Wert von *THD I*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **THD I Max:** Größter aufgetretener Wert von *THD I*. Kann über den Eingang *Reset Statistics* zurückgesetzt werden.
- **Harmonics Voltage:** Harmonische der Spannung.
- **Harmonics Current:** Harmonische des Stroms.

6.5 Interaktion mit Scope

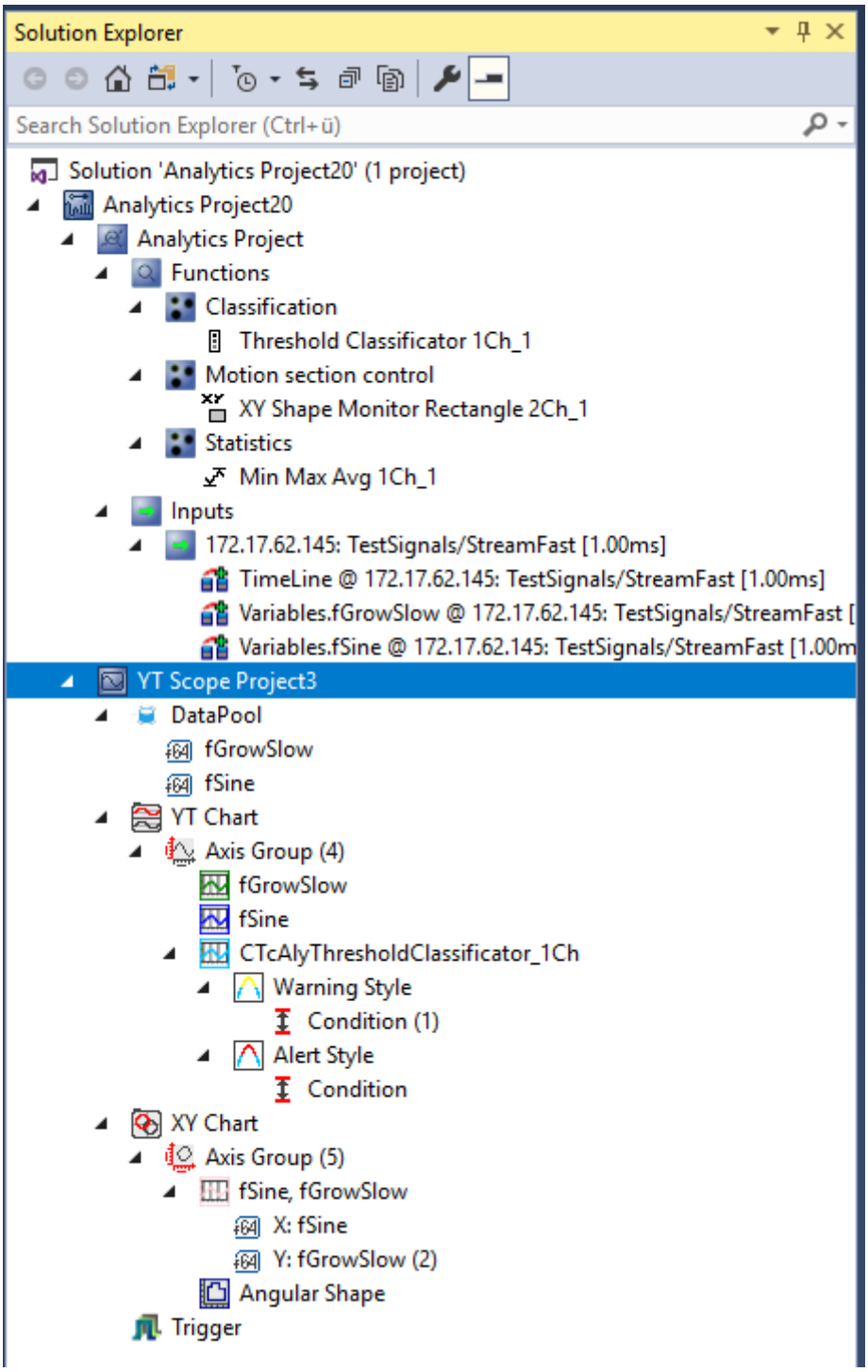
Die TwinCAT Analytics Engineering-Tools bieten eine einfache Interaktion zwischen Analytics-Konfigurator und Scope View. Sie ermöglichen es Ihnen, signifikante Werte im Datenstrom zu markieren und diese mit anderen Prozessdaten innerhalb des genauen Zyklus zu untersuchen. Zudem ist es möglich, die Ergebnisdaten des Algorithmus, wie einen durchschnittlichen oder maximalen Wert, im Scope View anzuzeigen.

✓ Wechseln Sie nach der Konfiguration der Analyse zur Startseite **Analytics Project**.

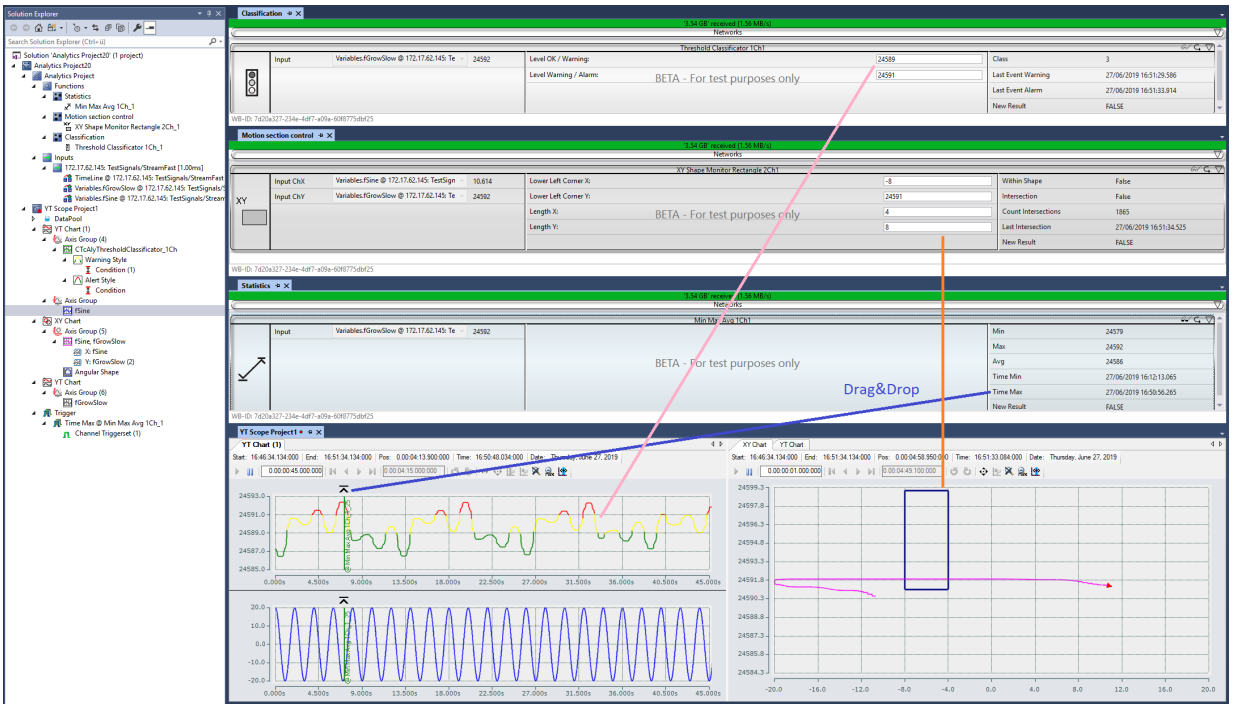
1. Klicken Sie auf **add referenced Scope…**



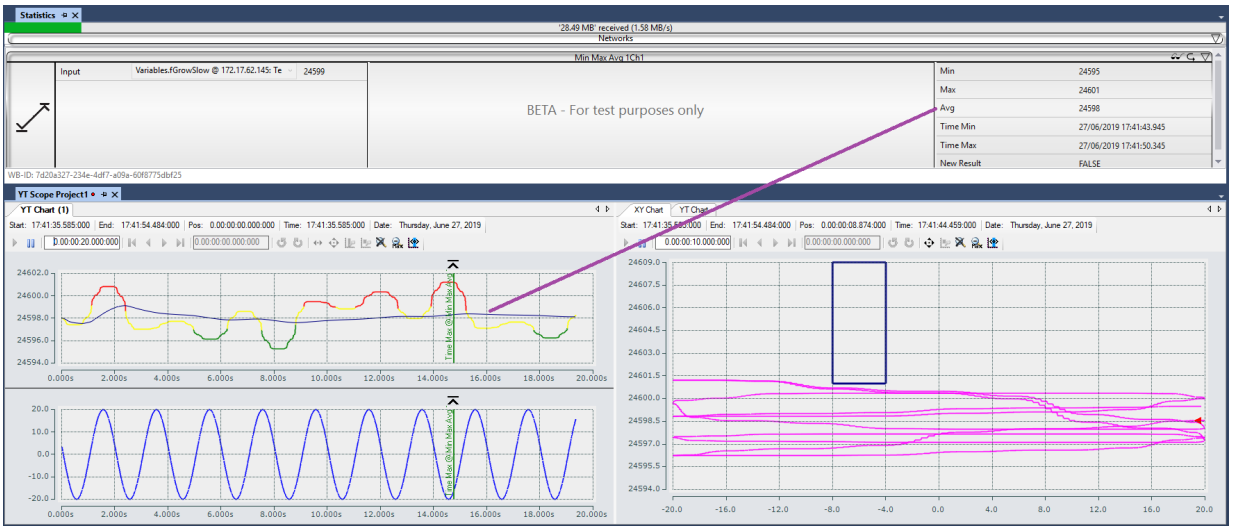
⇨ Dem Projekt wird automatisch eine geeignete Scope-Konfiguration hinzugefügt.



2. Es gibt zahlreiche Möglichkeiten, die Daten der verschiedenen Algorithmen zu visualisieren. Jegliche Zeitstempel eines Algorithmusausgangs können Sie per Drag-and-Drop in die Grafiken ziehen. Der Scope kennzeichnet die Position des Ereignisses mit einer farbigen Markierung (siehe blaue Linie im Bild unten).

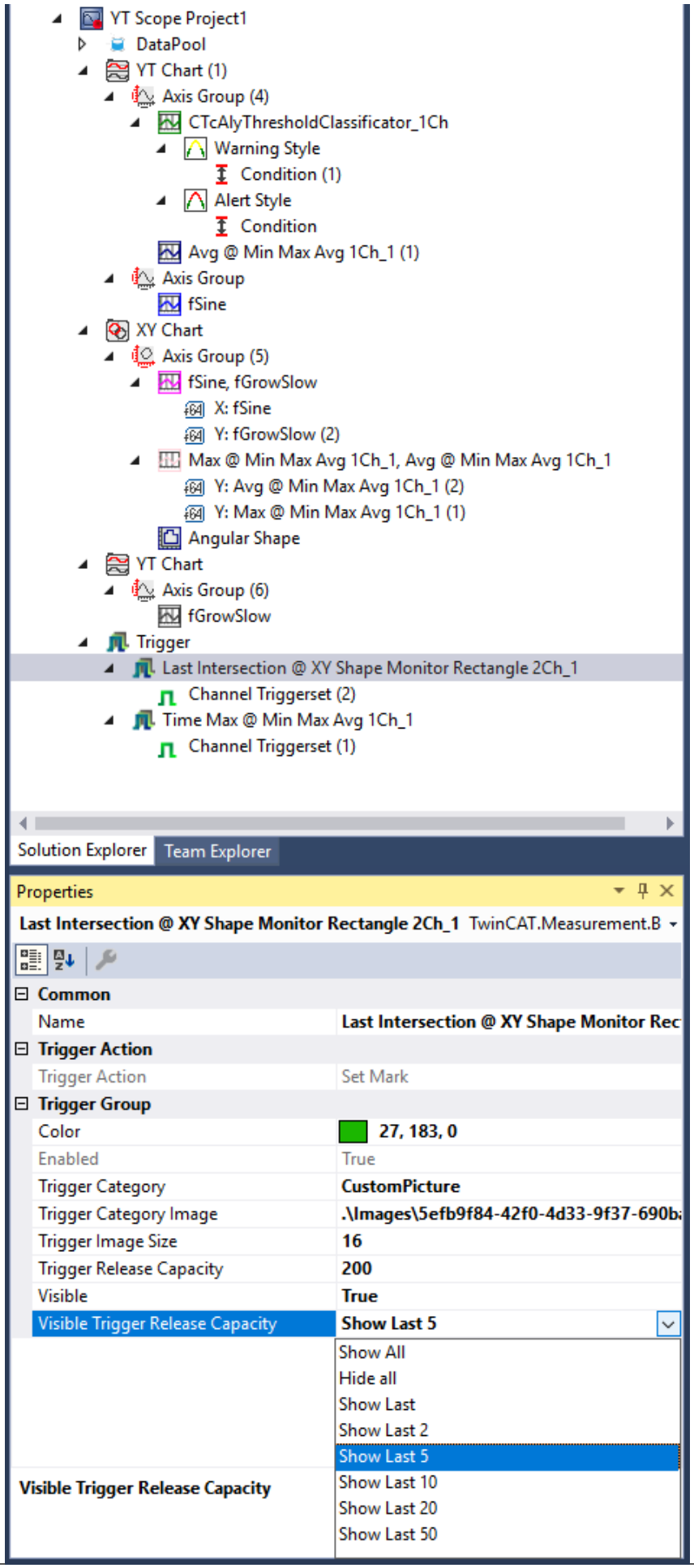


3. Für jede Art von Shape-Algorithmus erstellt der SW ein XY-Diagramm in der Scope-Konfiguration einschließlich einer Form mit der gegebenen Definition (siehe orangefarbene Linie im Bild oben). Für den Threshold-Klassifizierungsalgorithmus erfolgt im Scope View zudem automatisch ein dynamischer Farbwechsel von der Kanalfarbe zu gelb (Warnung) und zu rot (Alarm) (rosafarbene Linie).
4. Zusätzlich können Sie die Ergebniswerte des Algorithmus vor oder während der Aufzeichnung per Drag-and-Drop in ein Diagramm ziehen, um einen neuen Kanal hinzuzufügen, der z. B. den durchschnittlichen, minimalen oder maximalen Wert anzeigt.



5. Klicken Sie auf die entsprechende Trigger-Gruppe im Scope View-Baum um die Anzahl der angezeigten Ereignisse/Marker zu steuern.

6. Im Fenster Properties können Sie eine Anzahl für Visible Trigger Release Capacity festgelegen.



⇨ Sie können wählen zwischen:

- All
- Hide All
- Show Last
- Show Last 2
- Show Last 5
- Show Last 10
- Show Last 20
- Show Last 50

Scope-Konfiguration gespeichert im Netzwerktemplate

Eine erstellte Scope-Konfiguration kann zusammen mit dem dazugehörigen Netzwerk in einem Netzwerktemplate abgespeichert werden [▶ 360], um bei einer erneuten Verwendung eines Netzwerks automatisch die gleiche Scope-Konfiguration zu erhalten.

Zeitverhältnis zwischen Analytics-Konfiguration und Scope View

Beachten Sie, dass sich die Aufzeichnungszeit im Scope View von der in Analytics unterscheiden kann. Insbesondere, wenn der Ringpuffer im Scope View ausgewählt wird, könnte es möglich sein, dass sich einige signifikante Werte Ihrer Analyse in einer vergangenen Scope-Aufzeichnung befinden!

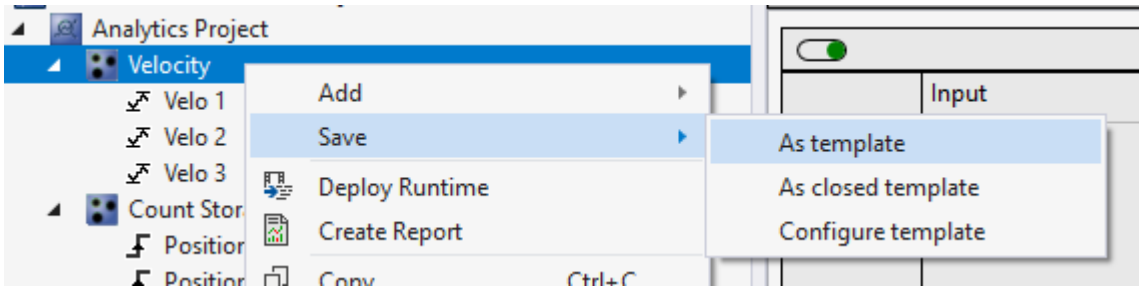
6.5.1 Scope Konfiguration im Netzwerk Template

In einem Netzwerktemplate lässt sich neben der Konfiguration eines Netzwerks auch die Konfiguration des verbundenen Scope Projektes abspeichern. Hierbei können jedoch nur die Bestandteile des Scopes mitgespeichert werden, welche auch eine Verbindung in das ausgewählte Netzwerk besitzen.

Je nachdem, ob die abzuspeichernde Scope-Konfiguration noch manuell angepasst werden soll, kann das Abspeichern des Templates über zwei Wege stattfinden.

Speichern ohne manuelle Konfiguration

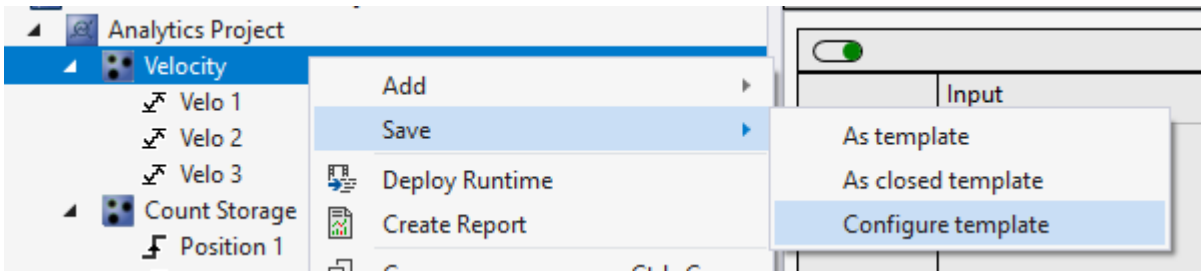
Wenn die Scope-Konfiguration ohne eine manuelle Anpassung im Template gespeichert werden soll, wählen Sie die Kontextmenü-Einträge **Save-As template** und **Save-As closed template.**



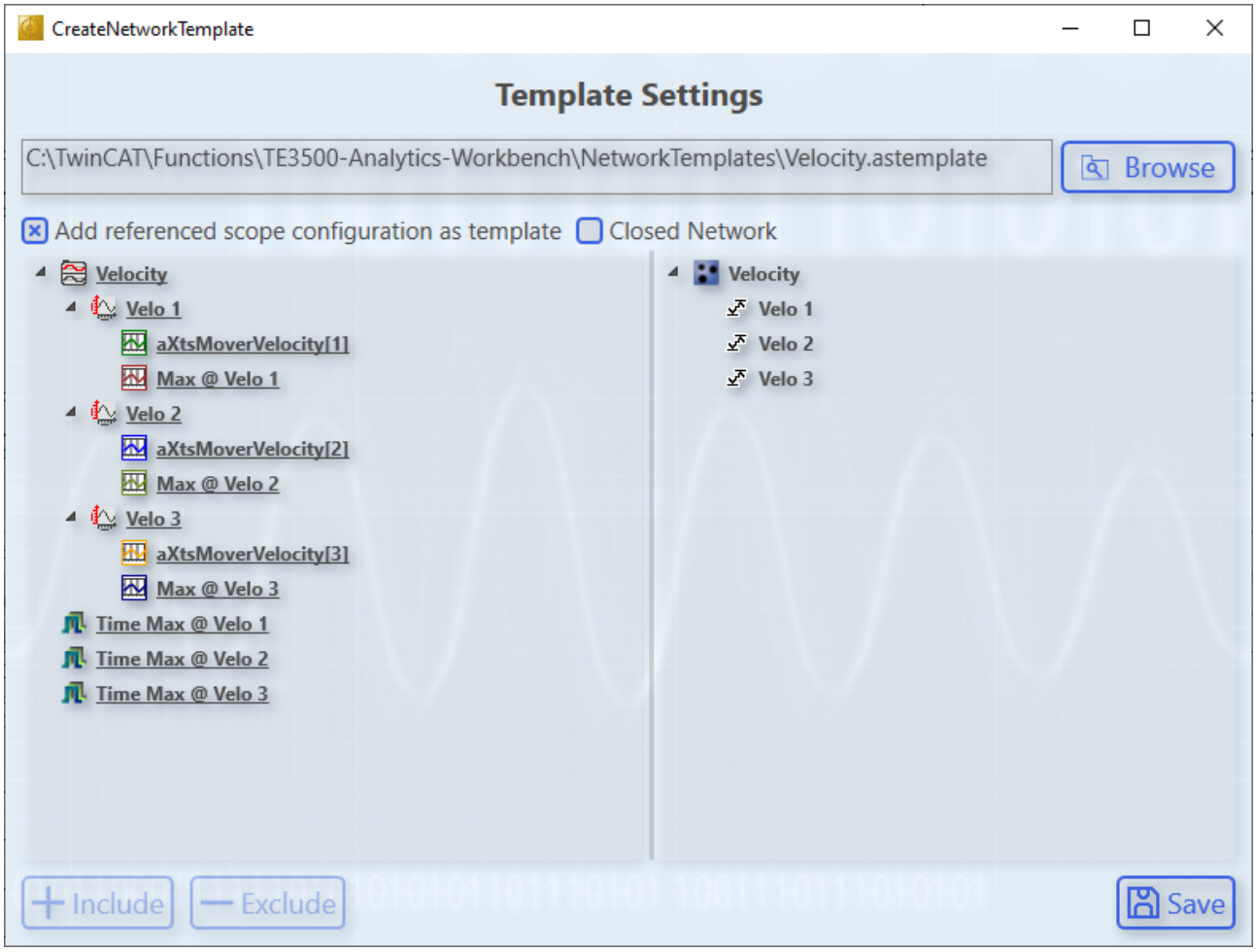
Legen Sie im nachfolgenden Dialog den Speicherort und den Namen des Templates fest. An der Stelle können Sie über dem Dateityp **Save as type** einstellen, ob die Scope-Konfiguration mit abgespeichert werden soll (**Scope Network Template**) oder ob nur das Netzwerk (**Network Template**) abgespeichert werden soll.

Speichern mit manueller Konfiguration

Wenn die Scope-Konfiguration vor dem Speichern kontrolliert und gegebenenfalls angepasst werden soll, wählen Sie im Kontextmenü den untersten Eintrag **Configure template**.



Zum Kontrollieren und Anpassen der Scope-Konfiguration öffnet sich ein neues Fenster, in welchem alle Scope-Komponenten auf der linken Seite und alle Analytics-Komponenten auf der rechten Seite dargestellt werden:

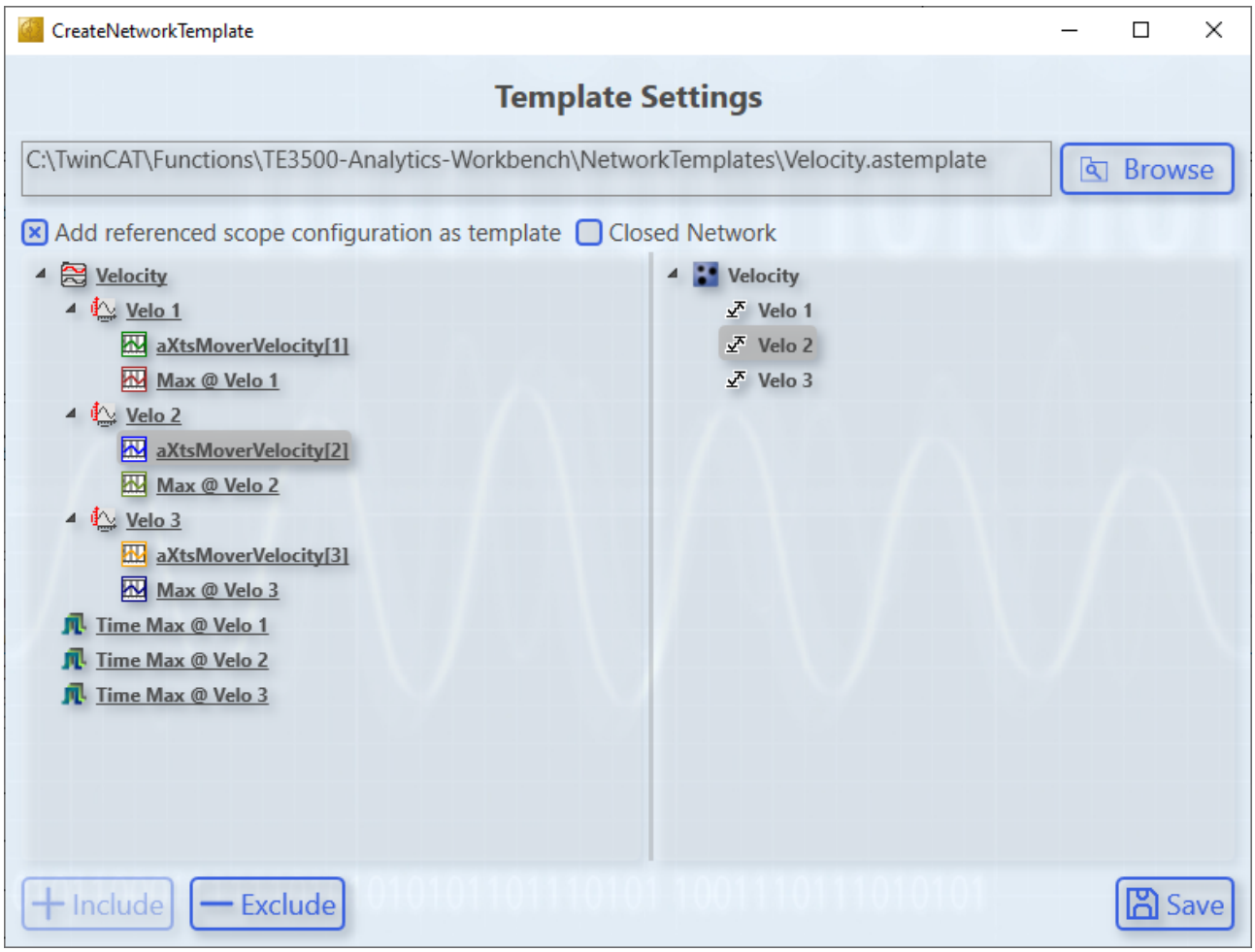


In der oberen Zeile können Sie den Speicherort und den Namen des Templates einstellen. Direkt darunter können Sie generelle Einstellungen vornehmen, welche das Template betreffen.

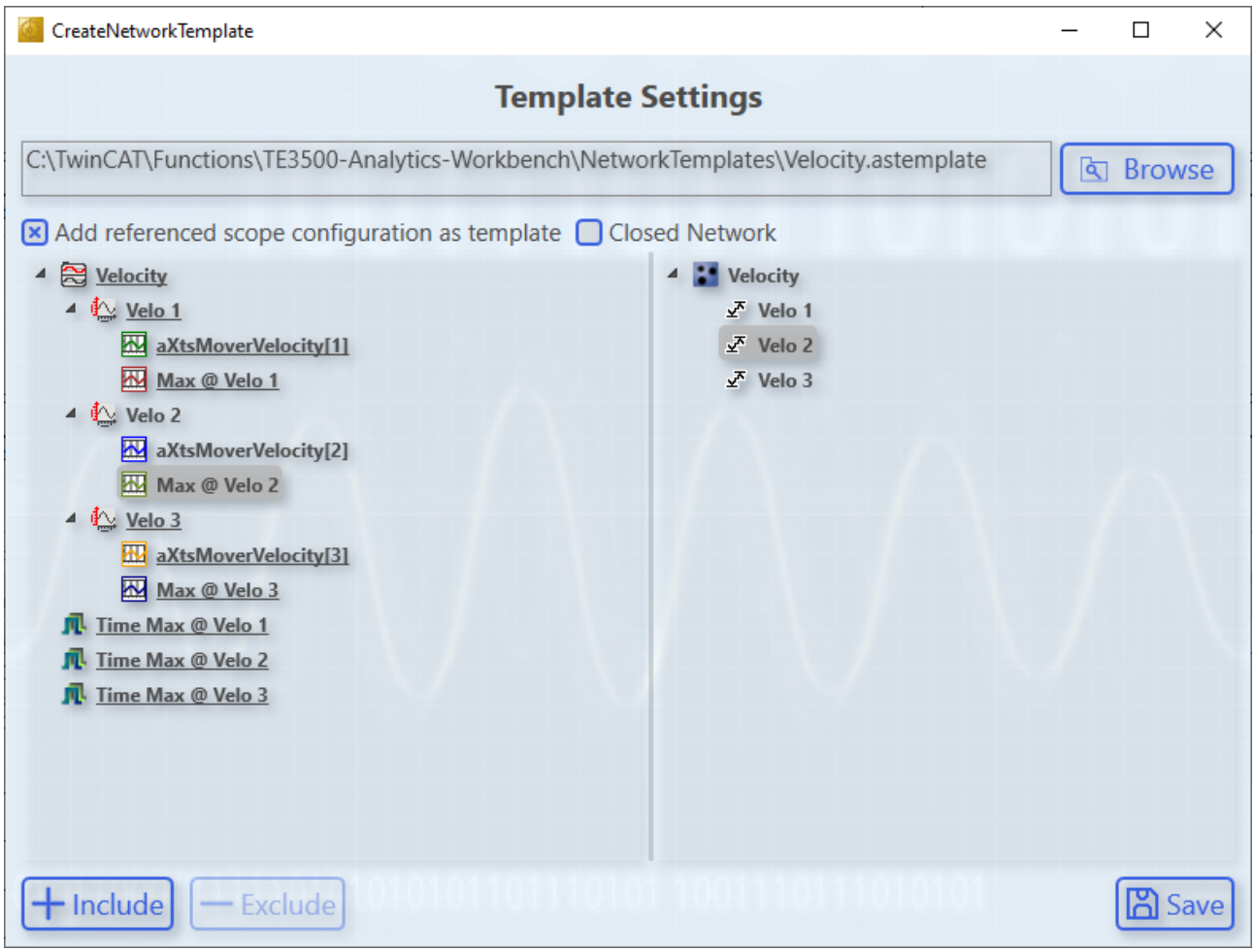
- **Add referenced scope configuration as template**
  Aktivieren Sie diese Einstellung, um die Scope-Konfiguration mit im Template abzuspeichern.
- **Closed Network**
  Diese Einstellung legt fest, in welchem Format das Template [▶ 70] abgespeichert werden soll. Diese Einstellung ist nur sichtbar, wenn in dem Netzwerk mindestens ein Netzwerk-Input und ein Netzwerk-Output konfiguriert sind.

In den beiden Baumansichten darunter werden die Konfigurationen angezeigt, um die Scope-Konfiguration gegebenenfalls noch zu bearbeiten. Hier können Sie Scope-Komponenten aus dem Template ausschließen aber auch wieder hinzufügen.
Alle Komponenten, welche zu dem Template hinzugefügt werden, sind in der Ansicht unterstrichen.

Wenn in dem Scope-Baum Elemente ausgewählt werden, werden auch alle Analytics-Komponenten ausgewählt, welche eine direkte Abhängigkeit zu der Scope-Komponente haben.



In dem oben abgebildeten Beispiel wird durch die Auswahl von **aXtsMoverVelocity[2]** automatisch das Modul **Velo 2** ausgewählt, da der Kanal die Input-Variable des Moduls anzeigt.
Ist eine Scope-Komponente ausgewählt, welche noch unterstrichen ist und somit in dem Template abgespeichert wird, können Sie diese mit der Schaltfläche **Exclude** aus dem Template herausnehmen.
Wenn Sie ein Element aus dem Template herausnehmen, werden auch automatisch alle Elemente, welche darunter liegen, herausgenommen. Zusätzlich wird das darüberliegende Element herausgenommen, wenn jedes Element darunter auch herausgenommen wurde.



Identisch zum Herausnehmen von Elementen, können Sie Scope-Komponenten auswählen, um diese mit der Schaltfläche **Include** wieder zu dem Template hinzuzufügen.
Hierbei ist zu beachten, dass ebenfalls automatisch alle darüberliegenden Elemente dem Template wieder hinzugefügt werden, auch wenn diese vorher abgewählt wurden.

Hinzufügen von Scope-Konfigurationen aus dem Template

Um ein Template mit der Scope-Konfiguration zu verwenden, müssen Sie das Template zu dem Analytics-Projekt hinzugefügen. Wenn es zu dem Projekt bereits ein referenziertes Scope-Projekt gibt, wird die Template-Konfiguration direkt dort hinzugefügt. Existiert noch kein referenziertes Scope-Projekt, wird die Template-Konfiguration beim späteren Erzeugen des Scope-Projekts hinzugefügt.
Bevor die Konfiguration zu dem Projekt hinzugefügt wird, öffnet sich eine Abfrage, bei welcher Sie das automatische Hinzufügen auch abbrechen können.

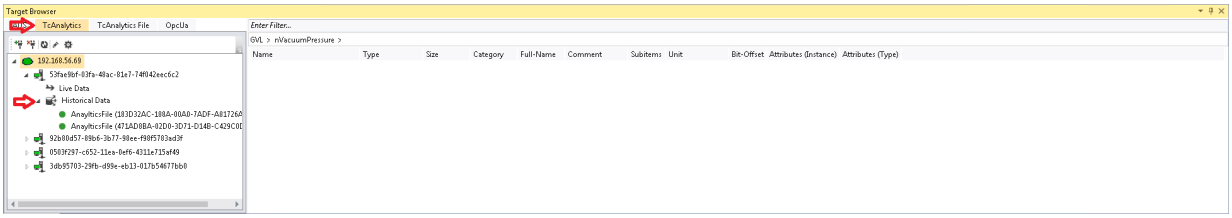
6.6 Arbeit mit historischen Daten

Historische Daten können mit der Analytics Workbench oder dem Analytics Service Tool analysiert werden. Um Ihre aufgezeichneten Daten anzuzeigen, benötigen Sie den TwinCAT Target Browser.

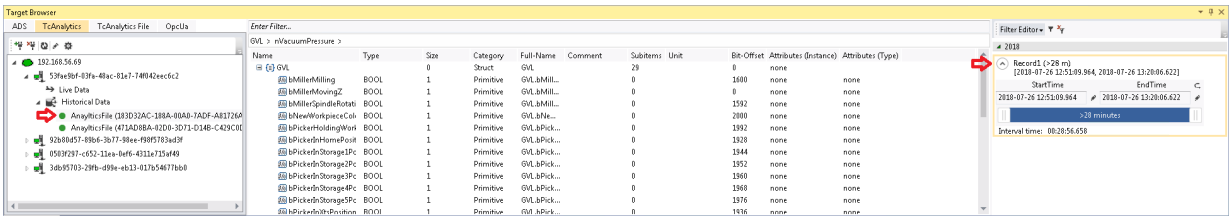
Auswahl der Daten aus dem TwinCAT Target Browser

Die historischen Daten können direkt aus dem Target Browser zu einem Eingang eines Analysealgorithmus gezogen werden.

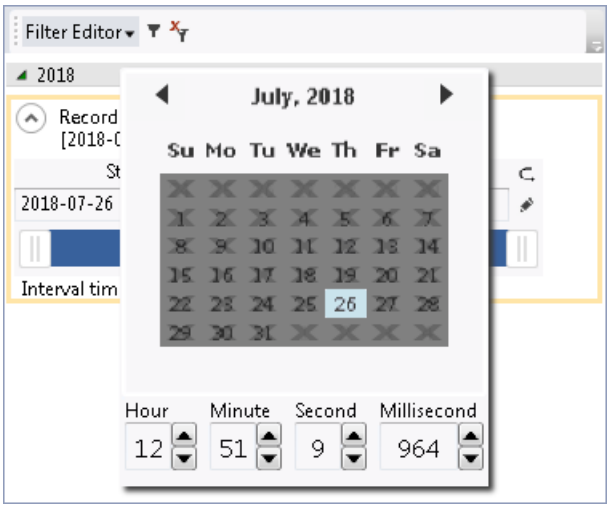
1. Zunächst müssen Sie in der linken Ecke des Target Browsers auf **TcAnalytics** klicken. Dort können Sie Ihren konfigurierten Broker sehen, der die Live- und historischen Daten Ihrer verschiedenen Geräte auflistet. Dies sollte wie in der folgenden Abbildung aussehen.



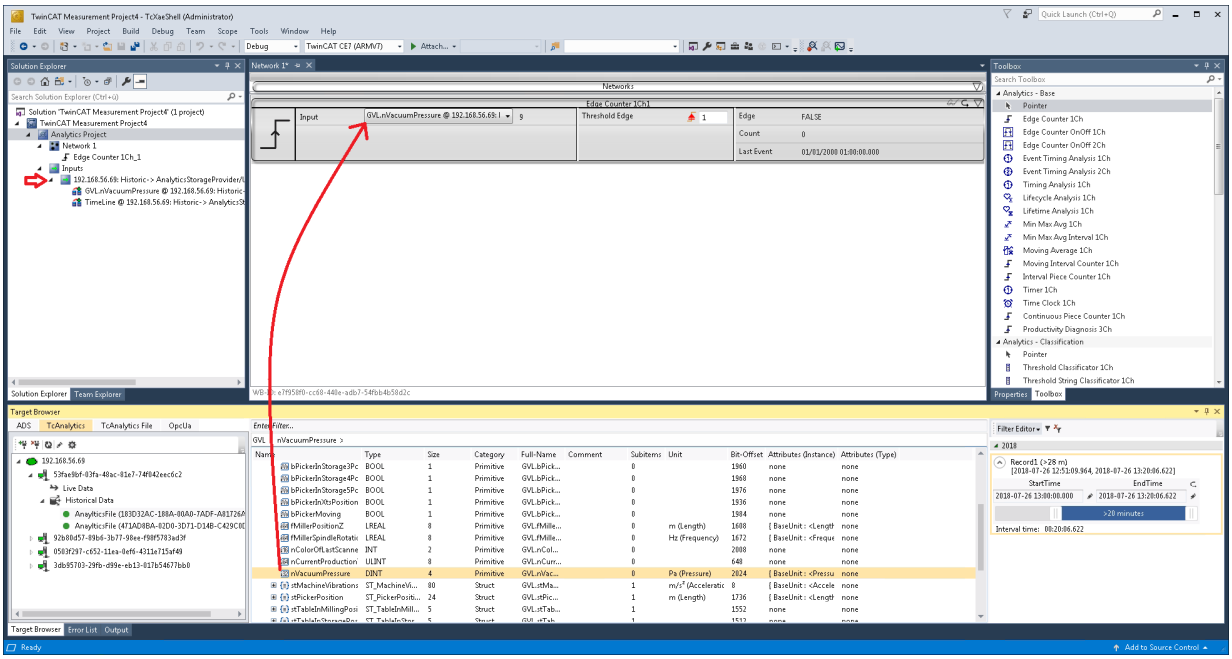
2. Gehen Sie zu dem historischen Stream, den Sie erstellt haben, und wählen Sie die Aufzeichnung aus, die analysiert werden soll. Alle Ihre Aufzeichnungen sind rechts im Fenster **Record** aufgelistet. Standardmäßig wird die letzte Aufzeichnung ausgewählt.



3. Wenn Sie live aufzeichnen, wird der Zeitbereich der Aufzeichnung alle paar Sekunden aktualisiert. Standardmäßig wird der gesamte Zeitbereich einer Aufzeichnung verwendet. Außerdem können Sie die Start- und Endzeit bearbeiten, um Ihren gewünschten Datenbereich zu analysieren. Dies kann mit einem Schieberegler, Textfeldern oder in einer grafischen Kalenderansicht erfolgen. Wenn Sie auf das Symbol rechts neben den Textfeldern klicken, wird die Kalenderansicht angezeigt.



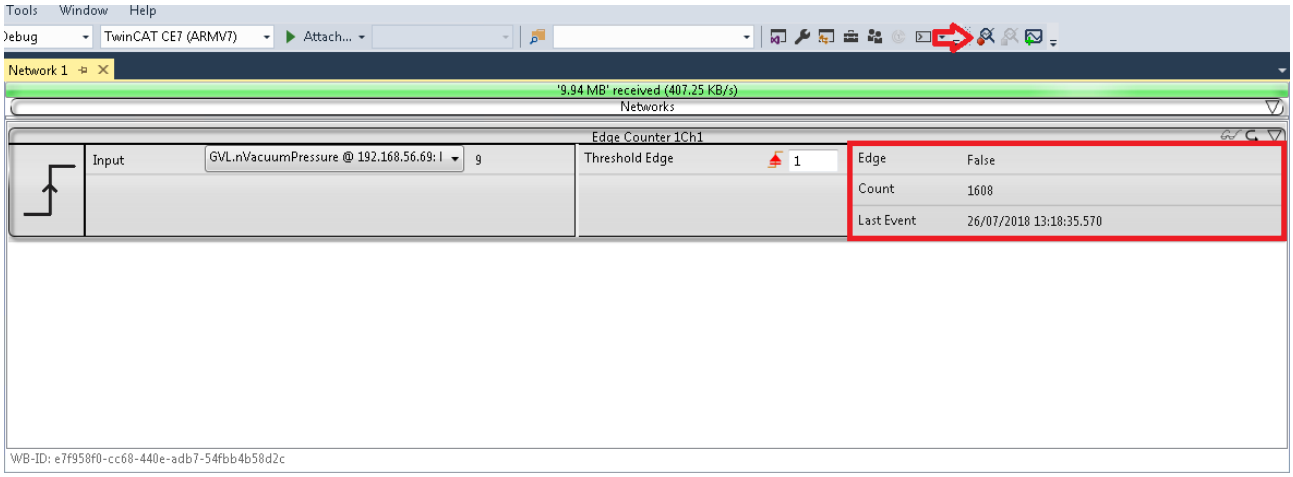
4. Nach diesen Schritten können Sie ein Symbol ebenso wie bei den Symbolen der Live-Daten per Drag-and-Drop zu einem Eingang eines Algorithmus ziehen.



⇨ Anschließend wird eine neue Eingangsquelle für Ihren historischen Stream generiert und kann im Projektmappen-Explorer Ihres Visual Studios® angezeigt werden. Als Erstes werden das gezogene Symbol und ein Zeitstempel der aktuellen Gerätezeit unter diesem Stream aufgelistet. Auch neue gezogene Symbole dieses Streams werden dort aufgelistet.

Analyse Ihrer historischen Daten im Analytics-Konfigurator

Um Ihre historischen Daten zu analysieren, drücken Sie auf die Schaltfläche **Start Analytics**. Anders als bei der Analyse von Live-Daten erscheint ein grüner Fortschrittsbalken. Die Geschwindigkeit Ihrer Analyse hängt von der Länge Ihrer Aufzeichnung, der Anzahl und Größe Ihrer Symbole sowie von Ihrer Breitbandgeschwindigkeit zum Broker ab. Die Analyse stoppt automatisch, wenn der Fortschrittsbalken am Ende angelangt ist. Die Ergebnisse bleiben sichtbar.



6.7 Runtime-Deployment

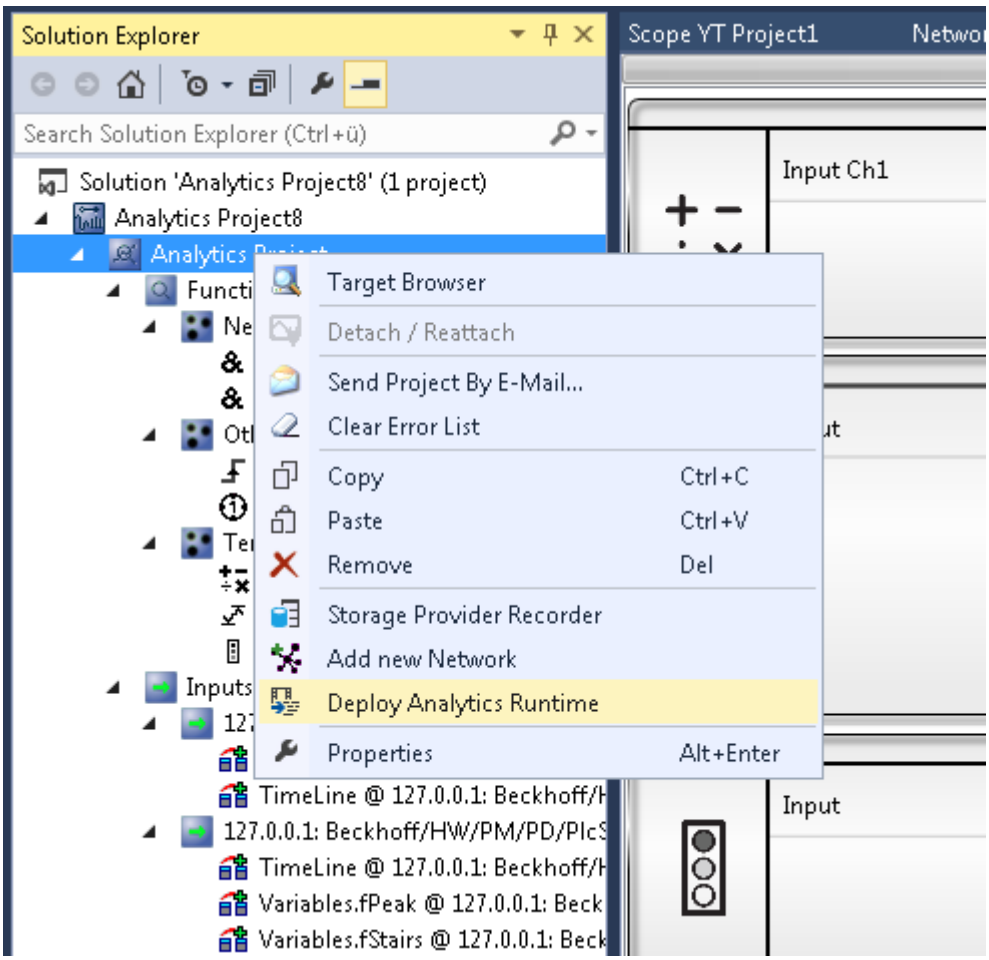
SPS-Code kann mit allen Modulen und Parametern generiert werden, die im TwinCAT Analytics Workbench Configurator konfiguriert sind. Dieser Code kann in eine TwinCAT Analytics Runtime heruntergeladen werden, um eine 24/7-Datenanalyse zu realisieren.

HINWEIS

Kompatibilität von automatisch generiertem SPS-Code

Der automatisch generierte SPS-Code basiert auf der TwinCAT Analytics Library. Schnittstellen der Basismodelle der Bibliothek sind mit früheren Versionen nach Freigabe der Bibliothek kompatibel. Der automatisch generierte Code selbst ist nur ein Beispielcode! Diese Codegenerierung kann sich ggf. von Version zu Version ändern. Soweit möglich, wird dies durch die Codegenerierungsversion gelöst.

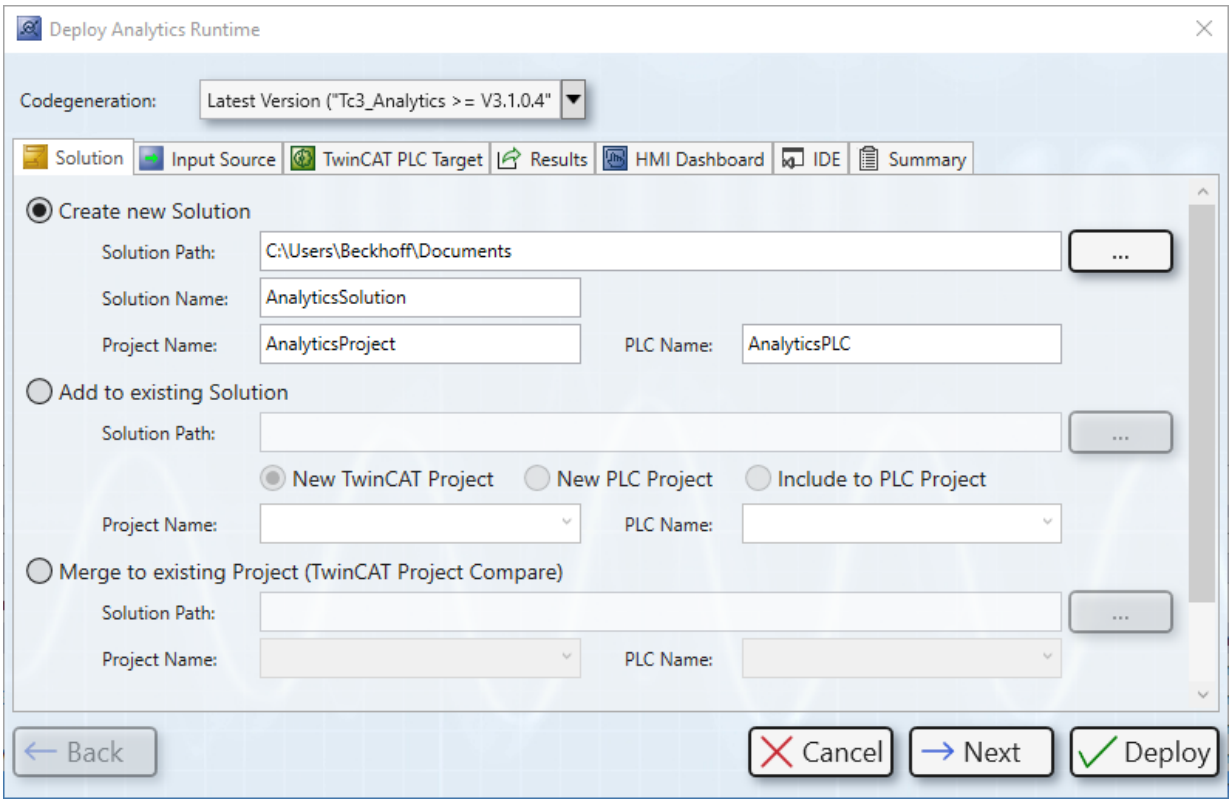
✓ Klicken Sie nach der Konfiguration auf den Befehl **Deploy Analytics Runtime** im Kontextmenü.



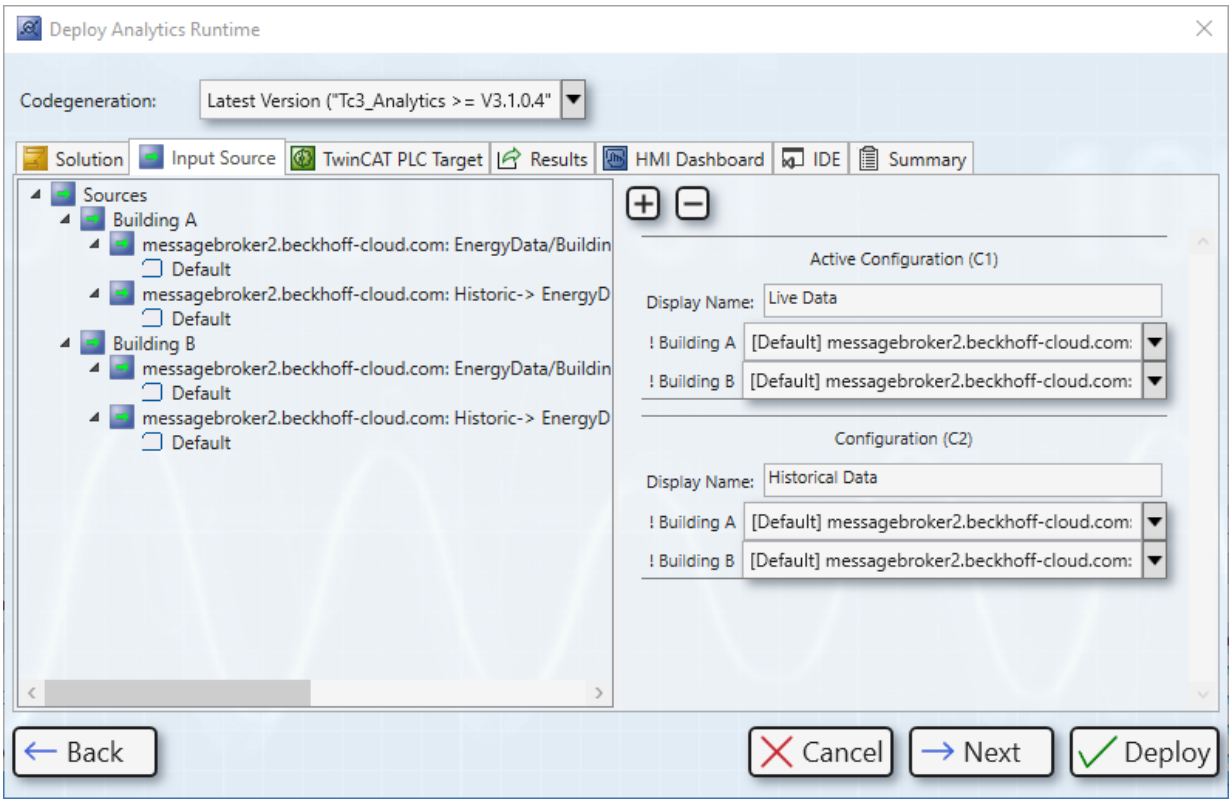
✓ Der Deploy-Assistent startet und es ist möglich, die gesamte benötigte Konfiguration für den Einsatz schrittweise einzurichten.

1. Wählen Sie auf dem ersten Karteireiter **Solution**, ob eine neue Projektmappe erzeugt werden soll (**Create new Solution**), ob in eine bereits bestehende Projektmappe der Analyse SPS Code als neues Projekt, neue PLC oder in eine bestehende PLC integriert bzw. hinzugefügt werden soll (**Add to existing**

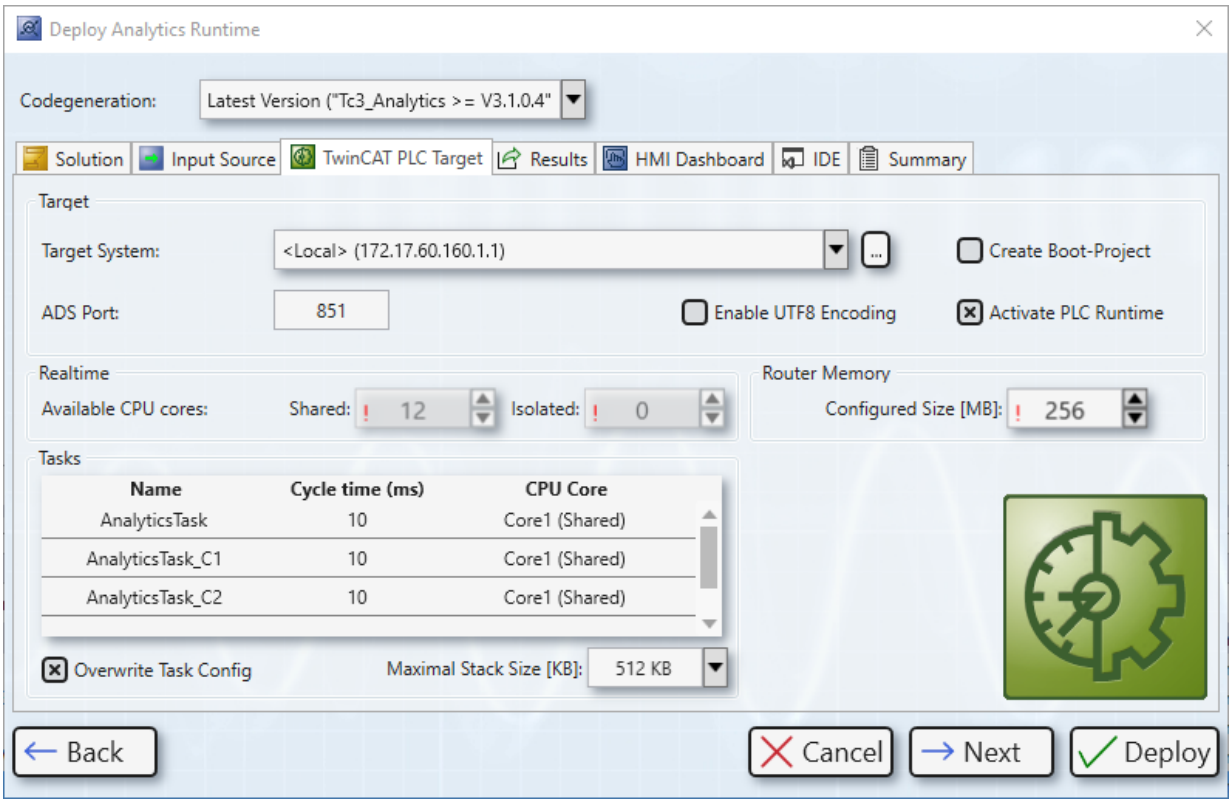
Solution), oder ob das neue Projekt in eine vorhandene Projektmappe mit Hilfe des TwinCAT Project Compare Tool eingefügt werden soll (**Merge to existing Project**).



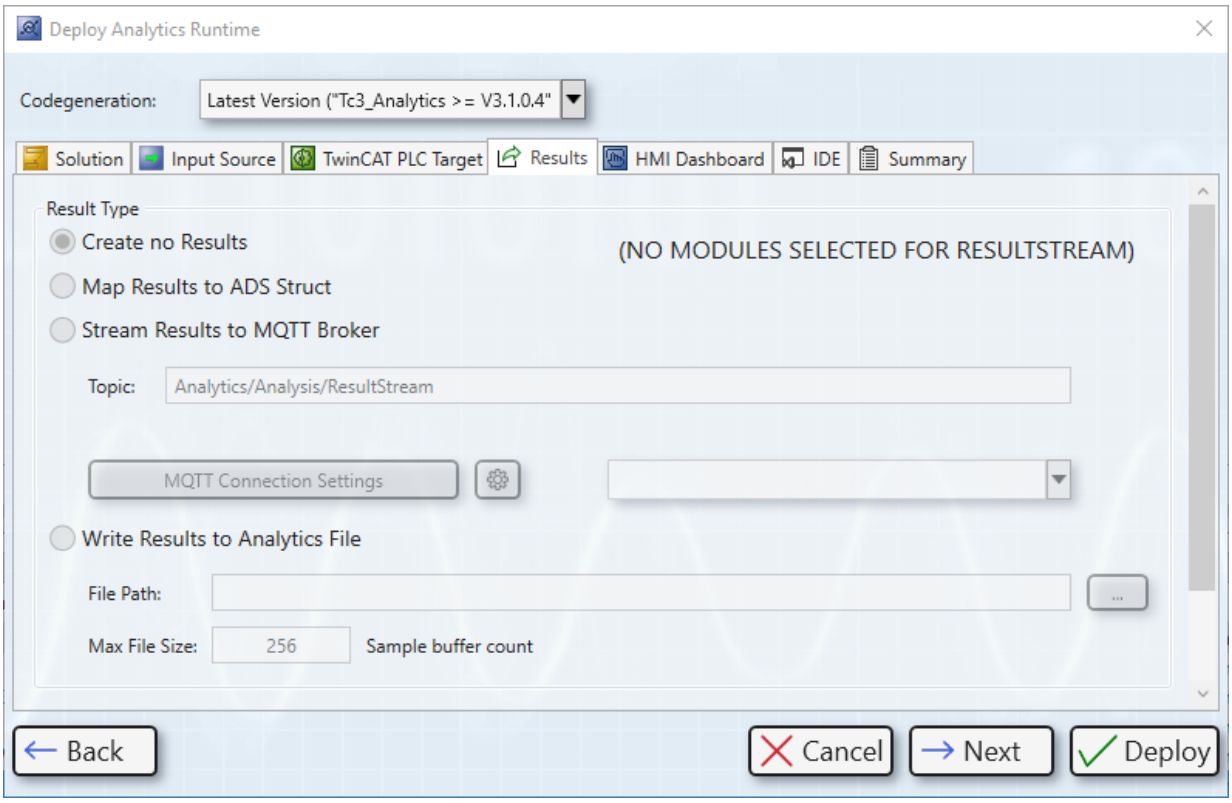
2. Wählen Sie auf dem zweiten Karteireiter **Input Source** mehrere Konfigurationen von Datenquellen aus. Dies ist z. B. erforderlich, wenn sowohl live-Daten als auch historische-Daten analysiert werden sollen.



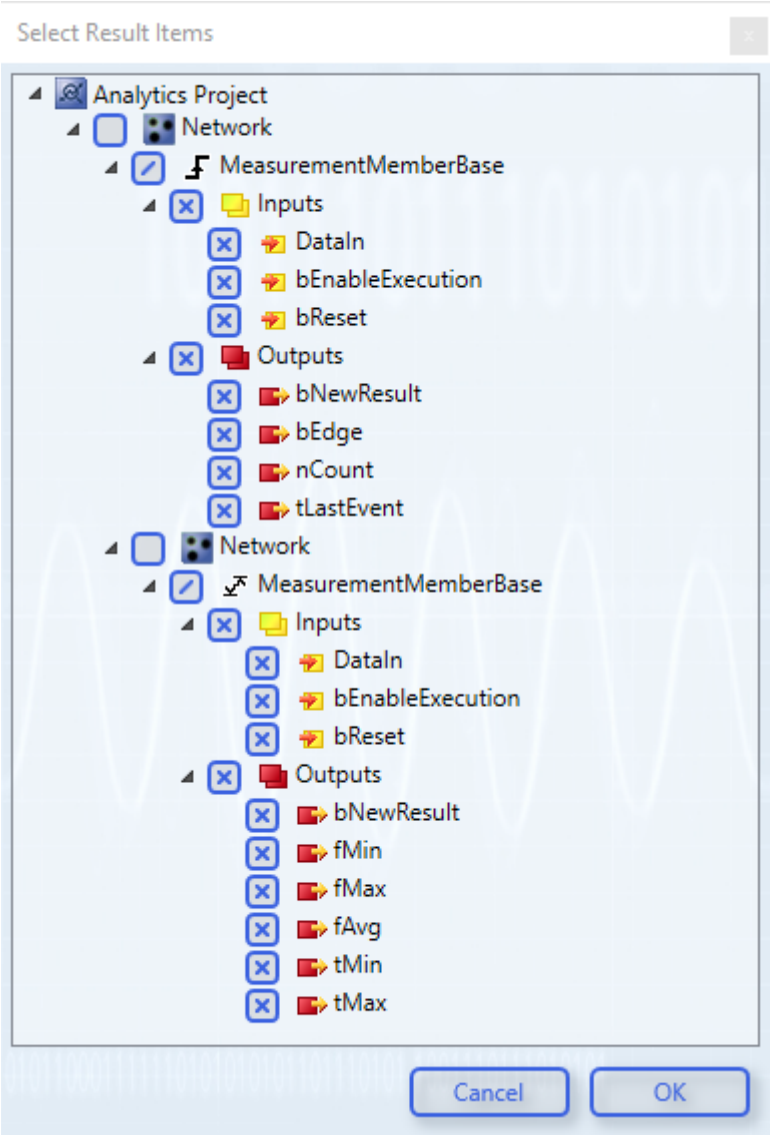
3. Legen Sie auf dem dritten Karteireiter **TwinCAT PLC Target** SPS-spezifische Parameter wie Target System, Task-Zykluszeit, oder auch die Taskzuordnung zu entsprechenden CPU-Kernen, fest.



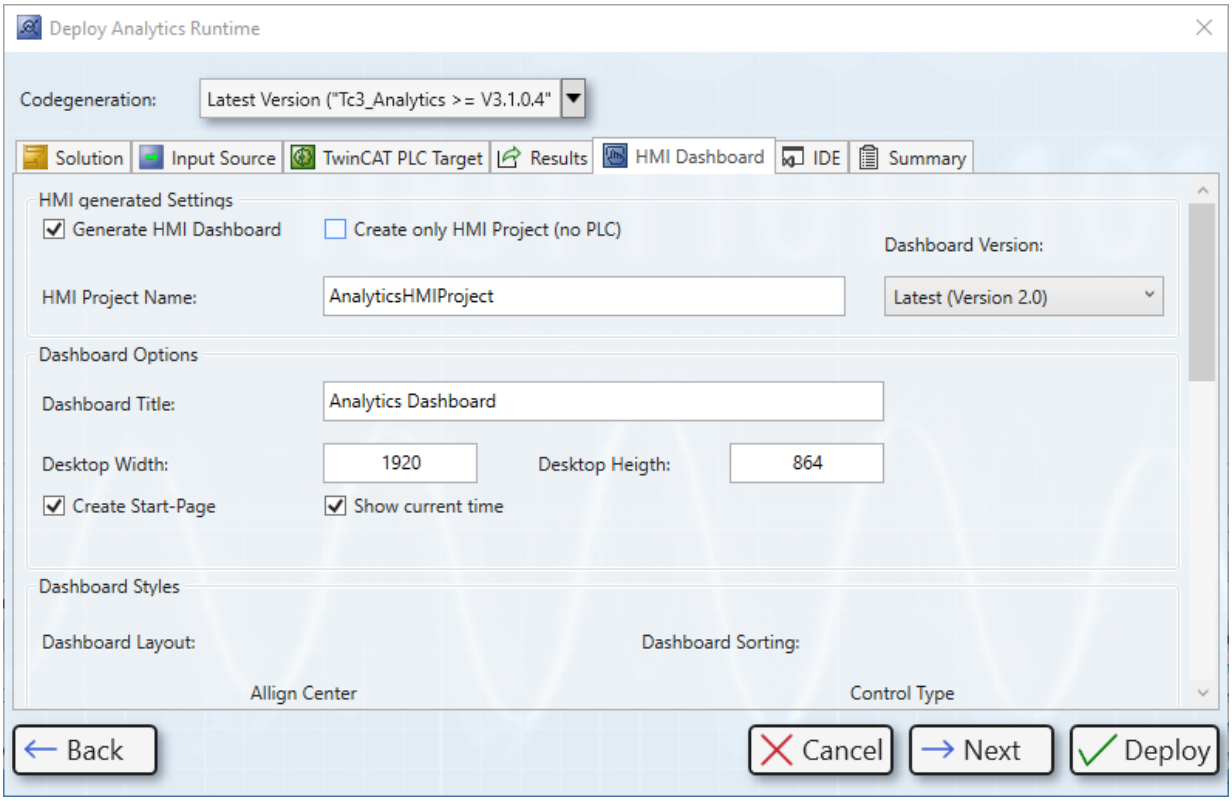
4. Wenn Sie die Eigenschaft **PLC Result** einiger Funktionen im Konfigurator gesetzt haben, ist der Karteireiter **Result** freigeschaltet.
   Legen Sie dort fest, wo die Ergebnisse gestreamt oder gespeichert werden.



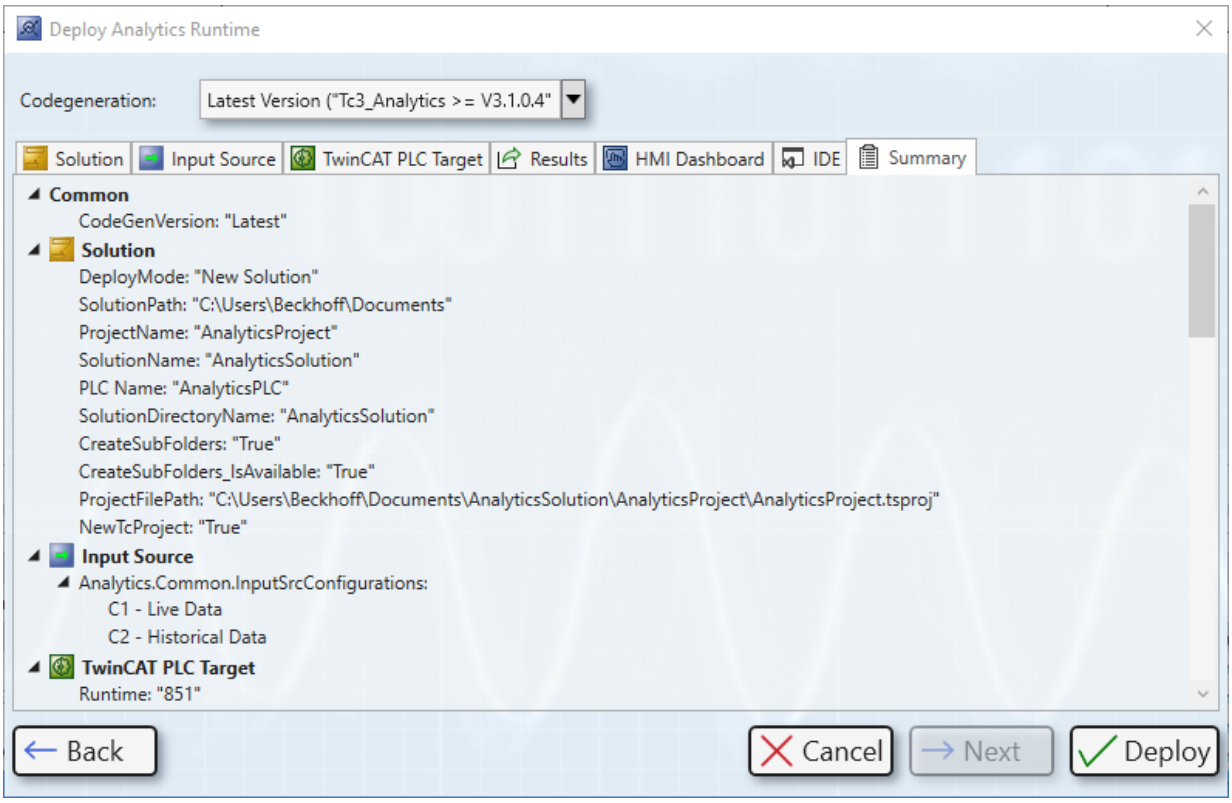
5. Durch Anklicken von **Select Result Items** ist es möglich, nur die gewünschten Werte auszuwählen.



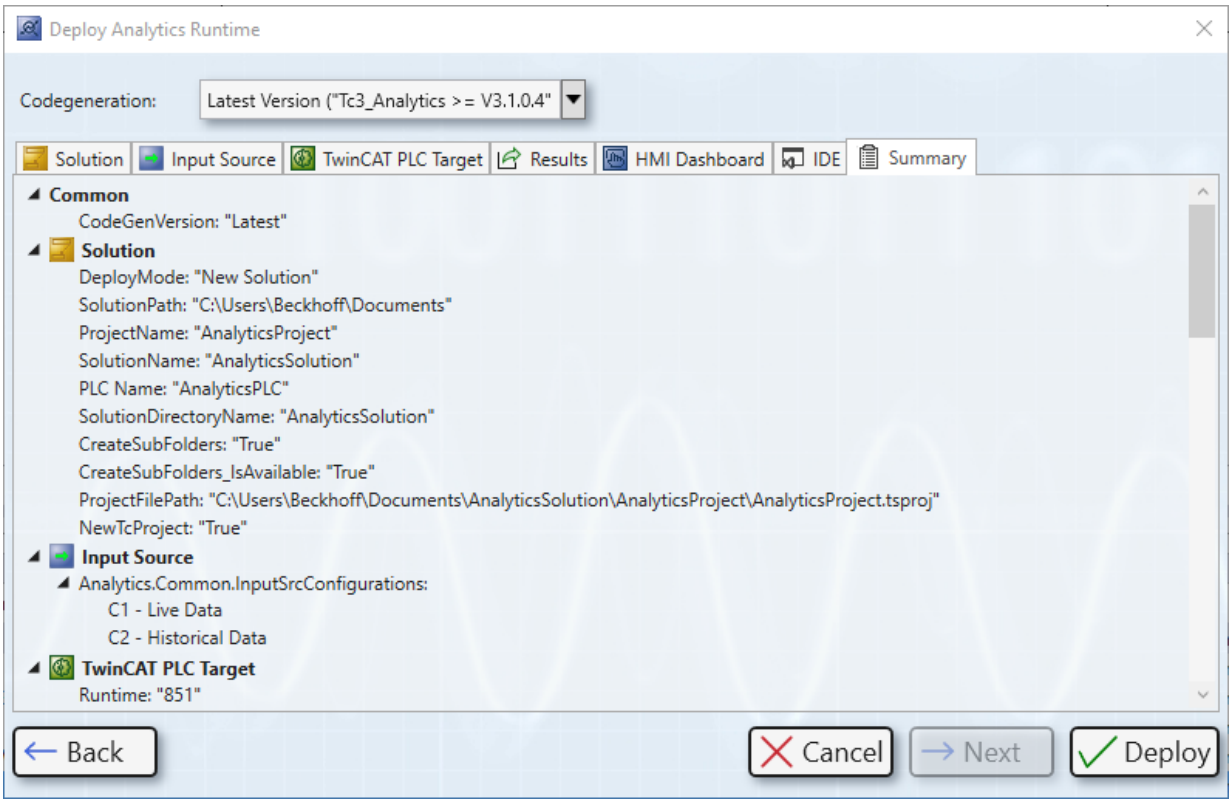
6. Wenn Sie ein oder mehreren Funktionen HMI Controls zugewiesen haben, ist der Karteireiter **HMI Dashboard** freigeschaltet. Hier können Sie verschiedene Einstellungen vornehmen, um ein maßgeschneidertes Dashboard zu generieren, das zu Ihren Bedürfnissen passt.



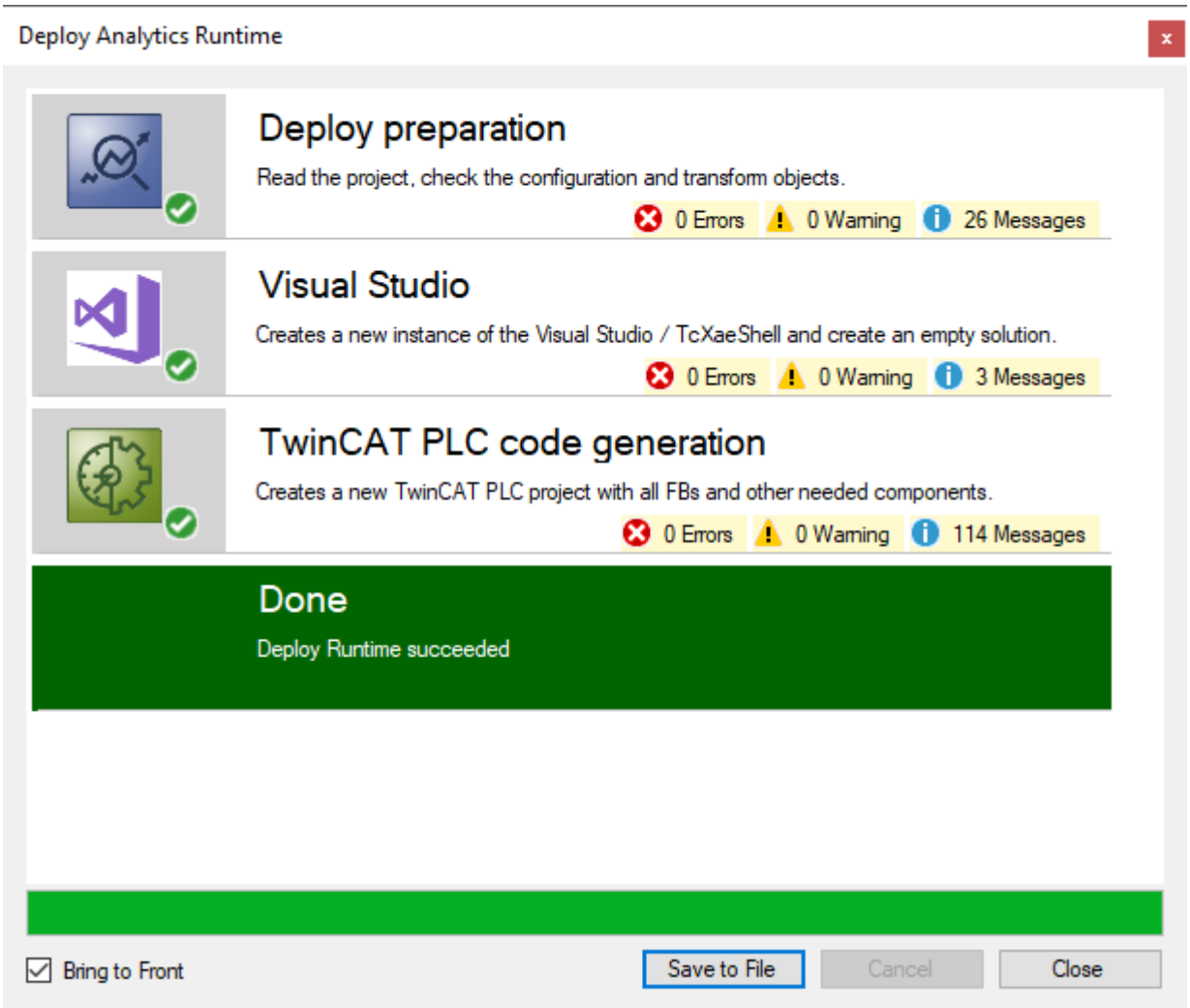
7. Wählen Sie auf dem Karteireiter **IDE** aus, welche Visual Studio®-Version oder TwinCAT XAE Shell für die Generierung verwendet werden soll, falls mehrere installiert sind.



8. Der Karteireiter **Summary** zeigt Ihnen alle Einstellungen, die Sie für die Generierung vorgenommen haben. Mit einem Klick auf **Deploy** starten Sie den Generierungsprozess.

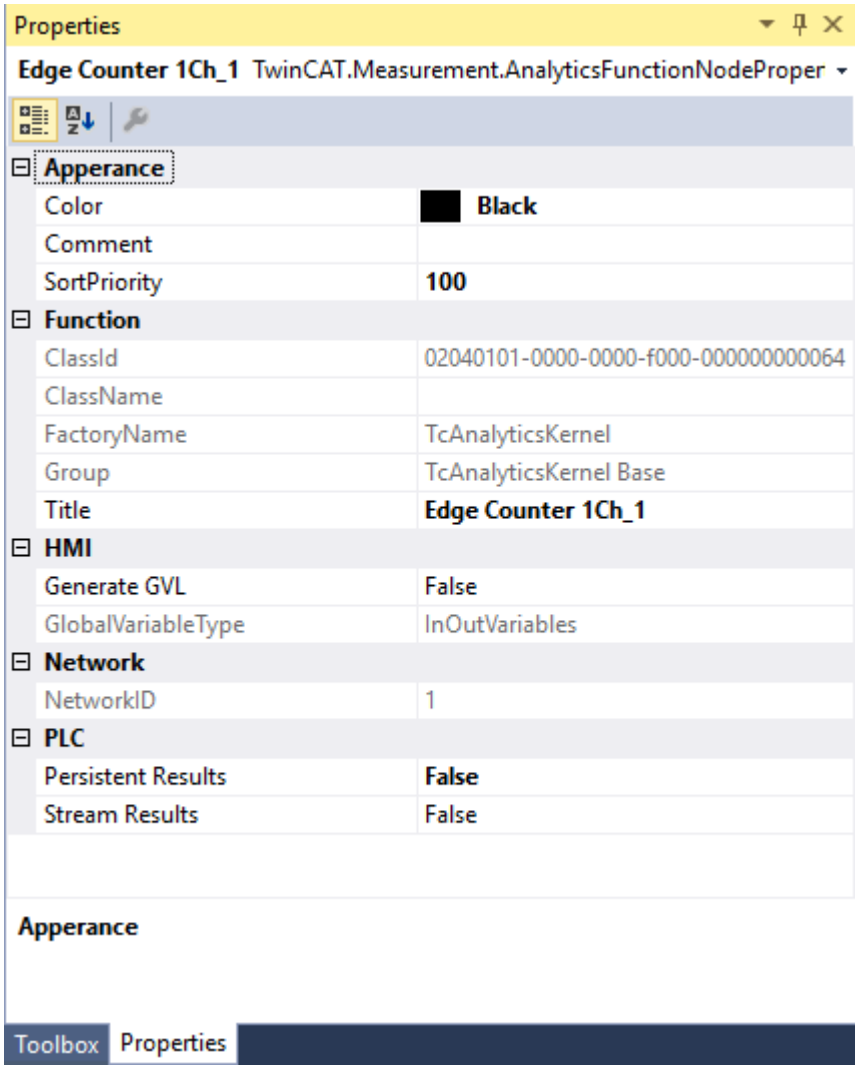


⇨ Das Übersichtsfenster zeigt jeden Schritt während des Generierungsprozesses, übersichtlich und in Kategorien unterteilt, an.



6.7.1 Algorithmuseigenschaften

Jeder Algorithmus des Analytics-Configurators stellt einige Eigenschaften bereit. Die Abschnitte HMI und PLC sind für die automatische Codegenerierung notwendig.



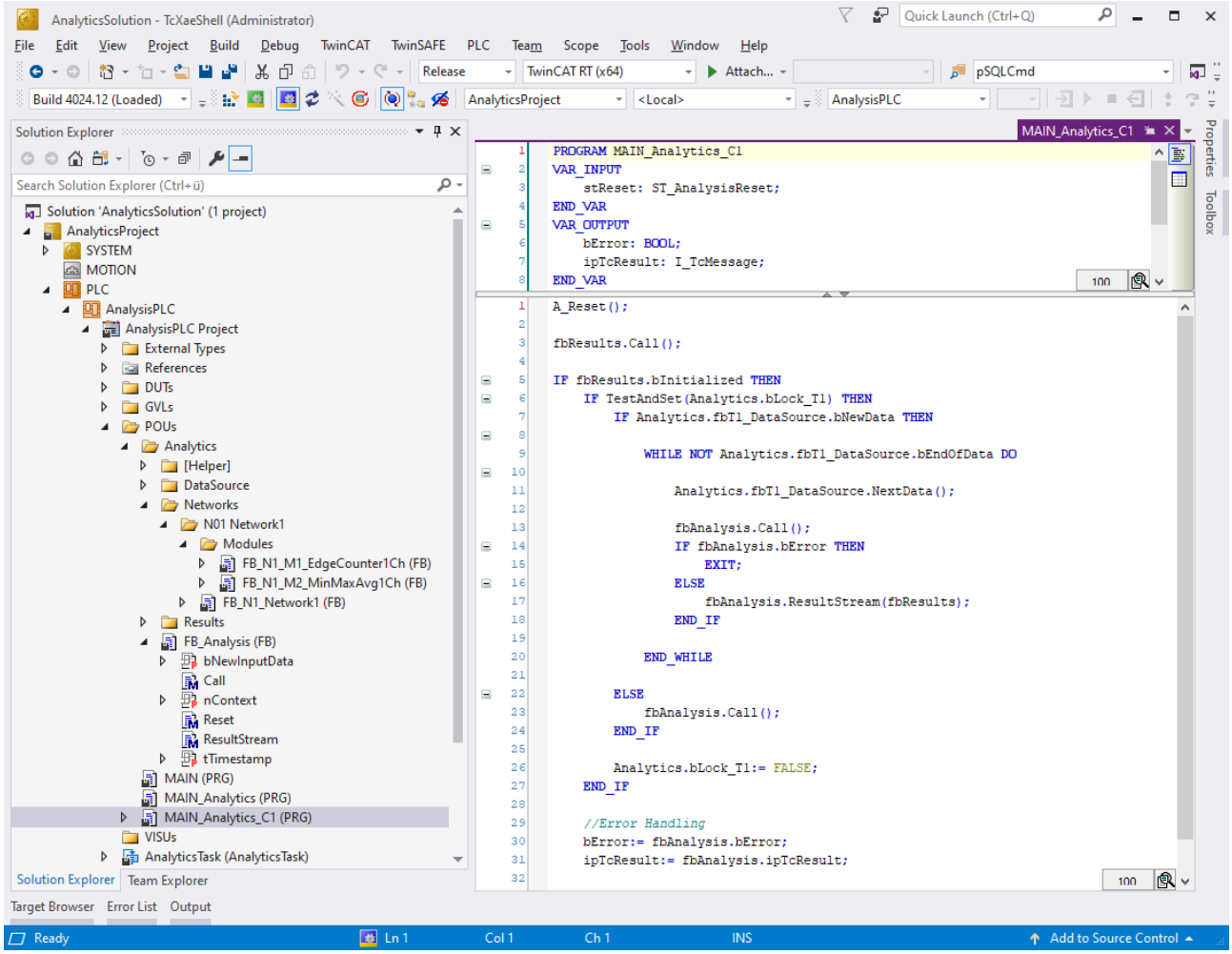
HMI

- **Generate GVL:** Ermöglichen Sie die Erzeugung einer globalen Variablenliste mit einer Sammlung von Variablen und entsprechendem Datentyp-Mapping für TwinCAT HMI.
- **GlobalVariableType:** Wählen Sie den Typ mit InOutVariables nur für die Eingänge und Ausgänge des Algorithmus oder KeyValuePairs für eine allgemeine Zuordnung zum STRING für Tabellen.

PLC

- **Persistent Results:** Aktivieren Sie dieses Flag, um die Ergebnisse des Algorithmus dauerhaft im Zielsystem der Analytics Runtime zu speichern.
- **Stream Results:** Aktivieren Sie dieses Flag, um die Ein- und Ausgänge des Algorithmus zu einem Ergebnis-Stream hinzuzufügen, der durch die Codegenerierung erzeugt wird.

6.7.2 SPS-Code



HINWEIS

Kompatibilität von automatisch generiertem SPS-Code

Der automatisch generierte SPS-Code basiert auf der TwinCAT Analytics Library. Schnittstellen der Basismodelle der Bibliothek sind mit früheren Versionen nach Freigabe der Bibliothek kompatibel. Der automatisch generierte Code selbst ist nur ein Beispielcode! Diese Codegenerierung kann sich ggf. von Version zu Version ändern.

Codegenerierung Versions Kompatibilität

	Version 2.1 (veraltet)	Version 3.0 (veraltet)	Version 4.0	Version 4.1	Version 5.0
Analytics Algorithmen	(X)	(X)	X	X	X
Filter Algorithmen	(X)	(X)	X	X	X
HMI Support	(X)	(X)	X	X	X
HMI mit erweitertem Inputstream Handling	-	(X)	X	X	X
Unterstützung von Array Eingänge	(X)	(X)	X	X	X
Unterstützung von Oversampling Eingängen	-	-	X	X	X
Netzwerktemplates	(X)	(X)	X	X	X
„Closed“ Netzwerktemplates	-	-	X	X	X
Netzwerk mit Eingängen, Ausgängen und Parametern	-	-	X	X	X
„Condition Monitoring“ Netzwerktemplates	-	-	X	X	X
Lambda Algorithmen	-	-	X	X	X
Parameter schreiben per HMI	-	-	-	X	X
„Power Monitoring“ Algorithmen	-	-	X	X	X
Scope Support	-	-	-	-	X

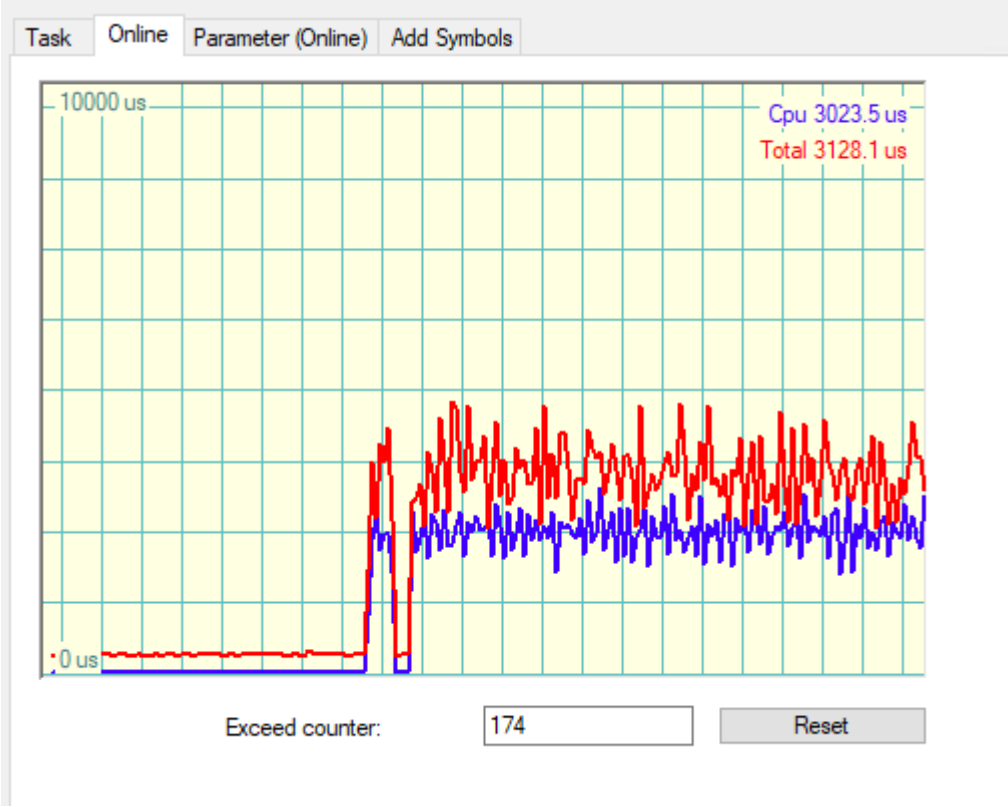
HINWEIS

Mögliche manuelle SPS-Code Optimierungen

Jedes automatisch generierte SPS-Projekt hat spezifische Eigenschaften und Anforderungen für eine reibungslose und fehlerfreie Ausführung. Manche dieser Eigenschaften können zum Zeitpunkt des Generierens nicht vorhergesehen werden. Um die Ausführung zu optimieren, sind unter Manuelle Optimierungen [▶ 376] verschiedene Stellen aufgeführt, wo manuell der generierte Code angepasst werden kann.

6.7.2.1 Manuelle Optimierungen

Zyklusüberschreitungen in der Abarbeitung der historischen Daten

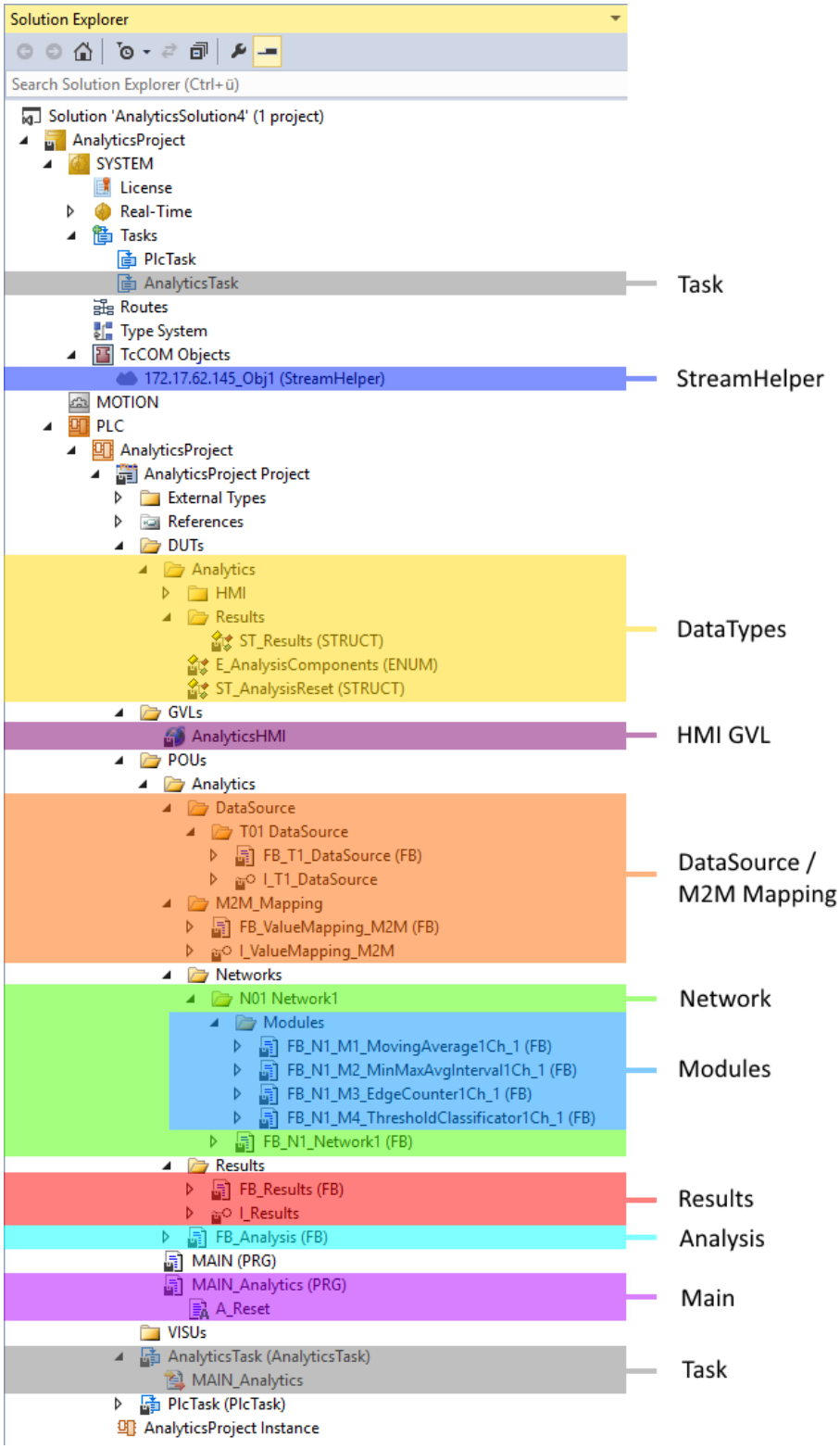


Dies kann durch folgende Parameter in der Analytics.GVL optimiert werden

```
{attribute 'qualified_only'}
VAR_GLOBAL
    //DataSources
    //Live:
    fbT1_InputSource: FB_T1_InputSource;
    //Historical:
    fbT2_C2_InputSourceHist: FB_T2_C2_InputSourceHistorical;
    //Empty:

    fbT1_IecCriticalSection: FB_IecCriticalSection;

    //Historical DataSource parameter
    nHistMaxInputBuffer: UDINT := 100;
    nHistMaxSampleCount: UDINT := 400;

    stReset: ST_AnalysisReset;
END_VAR
```

- Der Parameter nHistMaxInputBuffer gibt an, wie viele Samples maximal pro Zyklus prozessiert werden sollen. Je höher dieser Wert, umso größer ist die Gefahr von Zyklusüberschreitungen. Ist dieser Wert zu klein, kann es passieren, dass der Puffer vom StreamHelper überläuft und Datenverlust entsteht.
- Der Parameter nHistMaxSampleCount gibt an, wie viele Samples eine Message vom ASP enthalten darf. Je höher der Wert, umso schneller werden die Daten einer historischen Aufnahme übermittelt. Allerdings kann es so auch schneller zum Überlauf des StreamHelper Puffers kommen

6.7.2.2 Codeversion 2.1



Task:

Für die Analytics-Analyse wird ein separater Task erzeugt.

StreamHelper:

Wenn eine oder mehrere Datenquellen vom Typ MQTT-Binärstrom sind, erstellt die Codegenerierung eine Instanz eines StreamHelper-Objekts, um die eingehenden Binärstrommuster zu verarbeiten.

DataTypes:

Die Datentypen werden für die Analyse erstellt. Sie enthalten STRUCTs für die Reset-Funktion oder Ergebnisverarbeitung und ENUMs, um die verschiedenen Komponenten auszuwählen.

HMI GVL:

Um bequem Ein- und Ausgänge der Module mit dem HMI Dashboard zu mappen, werden ausgewählte Variablen als globale Variable generiert.

DataSource/M2M Mapping:

Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge. Der FB ValueMapping_M2M verwaltet das Werte-Mapping zwischen den Modulen (M2M – Module to Module) vom Modul INPUTs zum Modul OUTPUTs.

Network:

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

Modules:

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

Results:

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

Analyse:

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

MAIN:

Im Programm MAIN_Analytics wird der FB Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A_Reset definiert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics Version >= 3.1.0.0

6.7.2.2.1 FB_DataSource

Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

Syntax

Definition:

```
FUNCTION_BLOCK FB_T[n]_DataSource IMPLEMENTS I_T[n]_DataSource
VAR
END_VAR
```

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 379]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
GetData [▶ 379]	Local	Methode zum Abrufen der Daten des spezifizierten Elements.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.2.1.1 GetData

Syntax

```
METHOD GetData : BOOL
VAR_INPUT
    nElement : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Beispiel zu erhalten

Rückgabewert

Name	Typ	Beschreibung
GetData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

6.7.2.2.1.2 Call

Syntax

```
METHOD Call : BOOL
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.2.2 FB_Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_[Network1]
VAR_INPUT
    [module FBs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
Modul-FBs		Funktionsbausteine der konfigurierten Module.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 384]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 385]	Local	Zurücksetzen des Netzwerks mit allen Submodulen.
ValueMapping [▶ 380]	Local	Zuordnung der Eingangswerte zu den verschiedenen Moduleingängen.
SetHMIValues [▶ 381]	Local	Optional: Mapping der Ein-/Ausgänge der Module zu der globalen HMI-Variable.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.2.2.1 ValueMapping

Syntax

```
METHOD ValueMapping : BOOL
VAR_INPUT
    ipDataSource : I_T[n]_DataSource;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipDataSource	I_T[n]_DataSource	Daten für die Analyse von der spezifischen Datenquelle

Rückgabewert

Name	Typ	Beschreibung
ValueMapping	BOOL	

6.7.2.2.2.2 SetHMIValues

Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

Rückgabewert

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.2.2.3 Reset

Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.2.2.4 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipDataSource: I_T[n]_DataSource;
    [ipValueMapping_M2M: I_ValueMapping_M2M;]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipDataSource	I_T[n]_DataSource	Daten für die Analyse.
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.2.3 FB_Module

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_M[n]_[Module]
VAR_INPUT
    [module inputs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    [module outputs]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
Module inputs		Eingänge des ausgewählten Moduls.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.
Module outputs		Ausgänge des ausgewählten Moduls.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 383]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 383]	Local	Zurücksetzen des Moduls.
SetHMI [▶ 383]	Local	Optional: Setzt die Ein-/Ausgänge auf die globalen HMI-Strukturen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.2.3.1 Call

Syntax

```
METHOD Call : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.2.3.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.2.3.3 SetHMI

Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

Rückgabewert

Name	Typ	Beschreibung
SetHMI	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.2.4 FB_Analysis

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

Syntax

Definition:

```
FUNCTION_BLOCK FB_Analysis
VAR_INPUT
    [network FBs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
```

```
    ipTcResult: I_TcMessage;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
Network FBs		Funktionsbausteine der konfigurierten Netzwerke.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 384]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 385]	Local	Zurücksetzen der gesamten Analyse.
ResultStream [▶ 385]	Local	Optional: Wenn ein Ergebnis-Stream erstellt werden muss.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.2.4.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipDataSource: I_T[n]_DataSource;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipDataSource	I_T[n]_DataSource	Daten für die Analyse.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.2.4.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR_IN_OUT
    stReset: ST_AnalysisReset;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stReset	ST_AnalysisReset	Struktur, um zu definieren, welches Modul oder Netzwerk zurückgesetzt werden soll.

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.2.4.3 ResultStream

Syntax

```
METHOD ResultStream : BOOL
VAR_INPUT
    ipResults: I_Results;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipResults	I_Results	Schnittstellenzeiger auf den FB Results

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.2.5 FB_Results

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

Syntax

Definition:

```
FUNCTION_BLOCK FB_Results
VAR_OUTPUT
    stResults: ST_Results;
END_VAR
```

Ausgänge

Name	Typ	Beschreibung
stResults	ST_Results	Ergebnisstruktur, die alle Elemente des Ergebnis-Streams enthält.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 386]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
AddResult [▶ 386]	Local	Fügt dem Ergebnis-Stream ein Muster hinzu.
SendResults [▶ 387]	Local	Sendet alle gepufferten Muster des Ergebnis-Streams.
Release [▶ 387]	Local	Schließt den Stream oder die Datei des Ergebnis-Streams.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.2.5.1 Call

Syntax

```
METHOD Call : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.2.5.2 AddResult

Syntax

```
METHOD AddResult : BOOL
VAR_INPUT
    tTimestamp: ULINT;
    stSample: ST_Results;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
tTimestamp	ULINT	Zeitstempel des Samples
stSample	ST_Results	Sample-Struktur

Rückgabewert

Name	Typ	Beschreibung
AddResult	BOOL	

6.7.2.2.5.3 SendResults

Syntax

```
METHOD SendResults : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
SendResults	BOOL	

6.7.2.2.5.4 Release

Syntax

```
METHOD Release : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Release	BOOL	

6.7.2.2.6 MAIN_Analytics

Im Programm MAIN_Analytics wird der FB Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A_Reset definiert.

Syntax

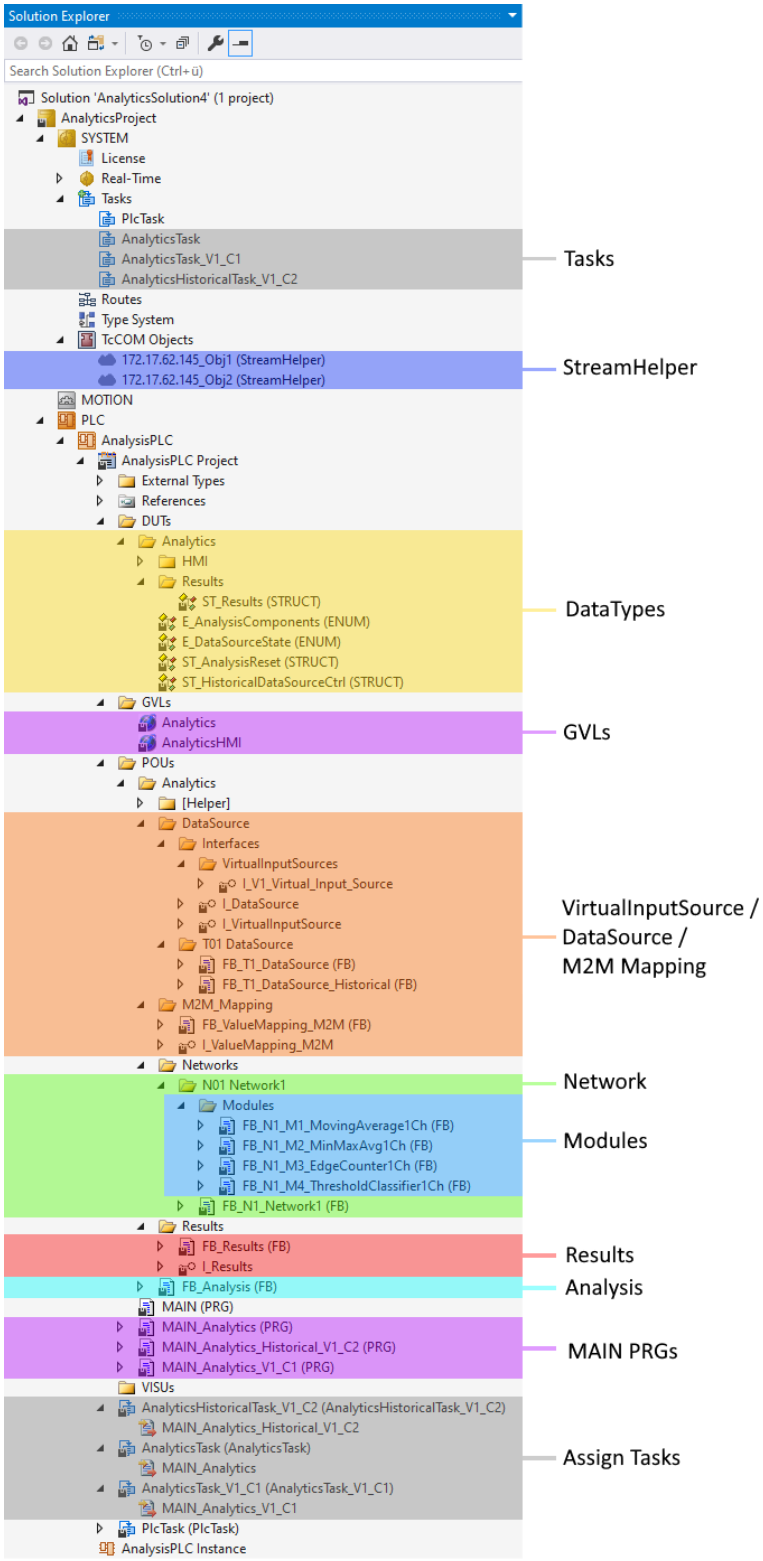
Definition:

```
PROGRAM MAIN_Analytics
VAR_INPUT
    stReset: ST_AnalysisReset;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.3 Codeversion 3.0



Tasks:

Für die Analytics-Analyse und für jede Konfiguration einer Virtual Input Source wird ein separater Task erzeugt.

StreamHelper:

Für jede Datenquellen vom Typ MQTT-Binärstrom, erstellt die Codegenerierung eine Instanz eines Stream Helper-Objekts, um die eingehenden Binärstrommuster zu verarbeiten.

DataTypes:

Die Datentypen werden für die Analyse erstellt. Sie enthalten STRUCTs für die Reset-Funktion oder Ergebnisverarbeitung und ENUMs, um die verschiedenen Komponenten auszuwählen.

GVLs:

Um bequem Ein- und Ausgänge der Module mit dem HMI Dashboard zu mappen, werden ausgewählte Variablen als globale Variable generiert. Zusätzlich werden die Data Source Baustein Instanzen und verschiedene Parameter als Globale Variable generiert.

VirtualInputSource / DataSource / M2M Mapping:

Die Virtual Input Source Interfaces abstrahieren die Data Source Symbole von der Analyse. Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge. Der FB ValueMapping_M2M verwaltet das Werte-Mapping zwischen den Modulen (M2M – Module to Module) vom Modul INPUTs zum Modul OUTPUTs.

Network:

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

Modules:

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

Results:

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker im Binär oder Json Format, oder speichert die Daten lokal in eine Analytics-Binärdatei.

Analysis:

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

MAIN PRGs:

In dem Programm MAIN_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion verwaltet und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

In den Programmen MAIN_Analytics_Vx_Cx wird der FB Analysis aufgerufen. Die Programme sind jeweils einem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A_Reset definiert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics Version >= 3.1.0.0

6.7.2.3.1 FB_DataSource

Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

Syntax

Definition:

```
FUNCTION_BLOCK FB_T[n]_DataSource IMPLEMENTS I_DataSource, I_V[n]_Virtual_Input_Source
VAR_OUTPUT
eDataState: E_DataSourceState;
END_VAR
```

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 390]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
GetData [▶ 390]	Local	Methode zum Abrufen der Daten des spezifizierten Elements.
GetDataOversampling [▶ 391]	Local	Methode zum Abrufen der Oversampling Daten des spezifizierten Elements.
NewDataAvailable [▶ 391]	Local	Methode zum Prüfen ob neue Daten verfügbar sind.
HistoricalCtrl [▶ 391]	Local	Methode zum Abrufen von historischen Daten.
UpdateRecordList [▶ 392]	Local	Methode zum Aktualisieren der Historischen Aufnahmeliste.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.3.1.1 GetData

Syntax

```
METHOD GetData : BOOL
VAR_INPUT
    nElement : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Beispiel zu erhalten.

Rückgabewert

Name	Typ	Beschreibung
GetData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist.

6.7.2.3.1.2 Call

Syntax

```
METHOD Call : BOOL
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.3.1.3 GetDataOversampling

Syntax

```
METHOD GetDataOversampling : BOOL
VAR_INPUT
nElement : UDINT;
nSample: : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Sample-Element zu erhalten.
nSample	UDINT	Sample-ID, um das spezifische Sample zu erhalten.

Rückgabewert

Name	Typ	Beschreibung
GetDataOversampling	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist.

6.7.2.3.1.4 NewDataAvailable

Syntax

```
METHOD NewDataAvailable : BOOL
VAR_INPUT
    nLastDataHandle : ULINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLastDataHandle	ULINT	Handle vom zuletzt abgeholten Datenpaket.

Rückgabewert

Name	Typ	Beschreibung
NewDataAvailable	BOOL	Ist TRUE, wenn neue Daten verfügbar sind.

6.7.2.3.1.5 HistoricalCtrl

Syntax

```
METHOD HistoricalCtrl : BOOL
VAR_INPUT
    stCtrl : REFERENCE TO ST_HMI_DataSourceCtrl;
    stHistStreamInfo : REFERENCE TO ST_HMI_DataSourceHist;
    stRecordInfo : REFERENCE TO ST_HMI_DataSourceHistRecordInfo;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stCtrl	ST_HMI_DataSourceCtrl	
stHistStreamInfo	ST_HMI_DataSourceHist	
stRecordInfo	ST_HMI_DataSourceHistRecordInfo	

Rückgabewert

Name	Typ	Beschreibung
HistoricalCtrl	BOOL	

6.7.2.3.1.6 UpdateRecordList

Syntax

```
METHOD UpdateRecordList : BOOL
VAR_INPUT
    stCtrl : REFERENCE TO ST_HMI_DataSourceCtrl;
    stHistStreamInfo : REFERENCE TO ST_HMI_DataSourceHist;
    sStreamSystemID : GUID;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stCtrl	ST_HMI_DataSourceCtrl	
stHistStreamInfo	ST_HMI_DataSourceHist	
sStreamSystemID	GUID	

Rückgabewert

Name	Typ	Beschreibung
UpdateRecordList	BOOL	

6.7.2.3.2 FB_Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_[Network1]
VAR_INPUT
    [module FBs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
Modul-FBs		Funktionsbausteine der konfigurierten Module.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die weitere Informationen über den Rückgabewert bietet.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 397]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 397]	Local	Zurücksetzen des Netzwerks mit allen Submodulen.
ValueMapping [▶ 393]	Local	Zuordnung der Eingangswerte zu den verschiedenen Moduleingängen.
SetHMIValues [▶ 393]	Local	Optional: Mapping der Ein-/Ausgänge der Module zu der globalen HMI-Variable.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.3.2.1 ValueMapping

Syntax

```
METHOD ValueMapping : BOOL
VAR_INPUT
    ipVirtual_Input_Source : I_V[n]_Virtual_Input_Source;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse von der spezifischen Datenquelle.

Rückgabewert

Name	Typ	Beschreibung
ValueMapping	BOOL	

6.7.2.3.2.2 SetHMIValues

Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur.

Rückgabewert

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.3.2.3 Reset

Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.3.2.4 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source: I_V[n]_Virtual_Input_Source;
    [ipValueMapping_M2M: I_ValueMapping_M2M;]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_T[n]_Virtual_Input_Source	Daten für die Analyse.
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.3.3 FB_Module

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_M[n]_[Module]
VAR_INPUT
    [module inputs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    [module outputs]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
Module inputs		Eingänge des ausgewählten Moduls.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die weitere Informationen über den Rückgabewert bietet.
Module outputs		Ausgänge des ausgewählten Moduls.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 395]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 396]	Local	Zurücksetzen des Moduls.
SetHMI [▶ 396]	Local	Optional: Setzt die Ein-/Ausgänge auf die globalen HMI-Strukturen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.3.3.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source : I_V[n]_Virtual_Input_Source;
    [ipValueMapping_M2M : I_ValueMapping_M2M;]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse.
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.3.3.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.3.3.3 SetHMI

Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur.

Rückgabewert

Name	Typ	Beschreibung
SetHMI	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.3.4 FB_Analysis

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

Syntax

Definition:

```
FUNCTION_BLOCK FB_Analysis
VAR_INPUT
    [network FBs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
Network FBs		Funktionsbausteine der konfigurierten Netzwerke.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die weitere Informationen über den Rückgabewert bietet.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 397]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 397]	Local	Zurücksetzen der gesamten Analyse.
ResultStream [▶ 398]	Local	Optional: Wenn ein Ergebnis-Stream erstellt werden muss.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.3.4.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source: I_V[n]_Virtual_Input_Source;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.3.4.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR_IN_OUT
    stReset: ST_AnalysisReset;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stReset	ST_AnalysisReset	Struktur, um zu definieren, welches Modul oder Netzwerk zurückgesetzt werden soll.

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.3.4.3 ResultStream

Syntax

```
METHOD ResultStream : BOOL
VAR_INPUT
    ipResults: I_Results;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipResults	I_Results	Schnittstellenzeiger auf den FB Results.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.3.5 FB_Results

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

Syntax

Definition:

```
FUNCTION_BLOCK FB_Results
VAR_OUTPUT
    stResults: ST_Results;
END_VAR
```

Ausgänge

Name	Typ	Beschreibung
stResults	ST_Results	Ergebnisstruktur, die alle Elemente des Ergebnis-Streams enthält.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 399]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
AddResult [▶ 399]	Local	Fügt dem Ergebnis-Stream ein Muster hinzu.
SendResults [▶ 399]	Local	Sendet alle gepufferten Muster des Ergebnis-Streams.
Release [▶ 400]	Local	Schließt den Stream oder die Datei des Ergebnis-Streams.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.3.5.1 Call

Syntax

```
METHOD Call : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.3.5.2 AddResult

Syntax

```
METHOD AddResult : BOOL
VAR_INPUT
    tTimestamp: ULINT;
    stSample: ST_Results;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
tTimestamp	ULINT	Zeitstempel des Samples
stSample	ST_Results	Sample-Struktur

Rückgabewert

Name	Typ	Beschreibung
AddResult	BOOL	

6.7.2.3.5.3 SendResults

Syntax

```
METHOD SendResults : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
SendResults	BOOL	

6.7.2.3.5.4 Release

Syntax

```
METHOD Release : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Release	BOOL	

6.7.2.3.6 MAIN_Analytics

Im Programm MAIN_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion gemanaged und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

Syntax

Definition:

```
PROGRAM MAIN_Analytics
VAR_INPUT
    sCurrentStreamSystemID: GUID;
    stHistStreamInfo: REFERENCE TO ST_HMI_DataSourceHist;
    stHistRecordInfo: REFERENCE TO ST_HMI_DataSourceHistRecordInfo;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.3.7 MAIN_Analytics_V[n]_C[n]

Im Programm MAIN_Analytics_V[n]_C[n] wird der FB Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A_Reset definiert.

Das Mapping der HMI Werte wird in der Aktion A_MapToHMI vorgenommen.

Syntax

Definition:

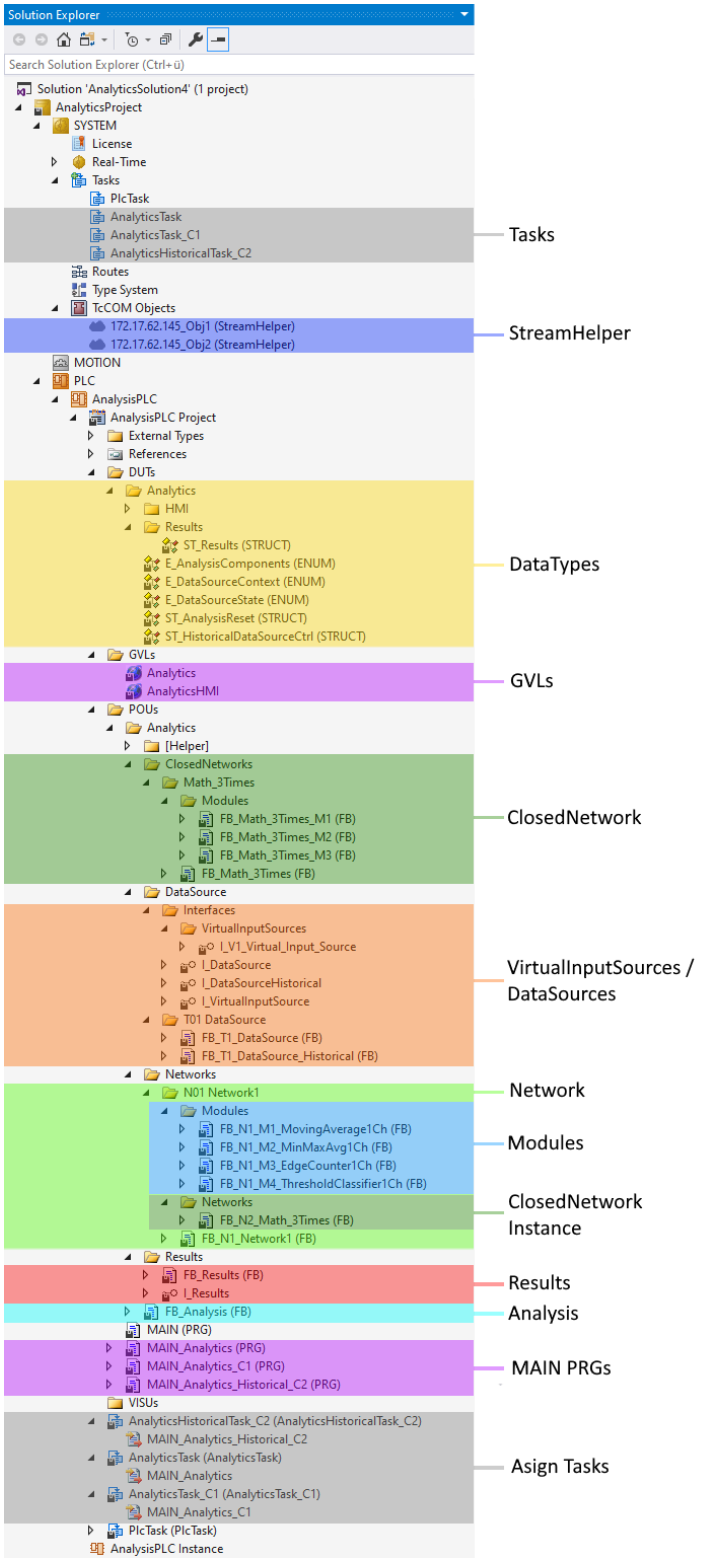
```
PROGRAM MAIN_Analytics_V[n]_C[n]
VAR_INPUT
    stReset: ST_AnalysisReset;
```

```
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    nLastDataHandle: ULINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.4 Codeversion 4.x



Tasks:

Für die Analytics-Analyse und für jede Konfiguration einer Virtual Input Source wird ein separater Task erzeugt.

StreamHelper:

Für jede Datenquellen vom Typ MQTT-Binärstrom, erstellt die Codegenerierung eine Instanz eines Stream Helper-Objekts, um die eingehenden Binärstrommuster zu verarbeiten.

DataTypes:

Die Datentypen werden für die Analyse erstellt. Sie enthalten STRUCTs für die Reset-Funktion oder Ergebnisverarbeitung und ENUMs, um die verschiedenen Komponenten auszuwählen.

GVLs:

Um bequem Ein- und Ausgänge der Module mit dem HMI Dashboard zu mappen, werden ausgewählte Variablen als globale Variable generiert. Zusätzlich werden die Data Source Baustein Instanzen und verschiedene Parameter als Globale Variable generiert.

ClosedNetwork:

Die ClosedNetwork FBs werden einmal mit allen Subnetzwerken und Modulen generiert. Sie können in der Analyse mehrmalig instanziiert werden. Auf diese Weise kann der generierte Code reduziert und vereinfacht werden.

VirtualInputSource / DataSource:

Die VirtualInputSource Interfaces abstrahieren die DataSource Symbole von der Analyse. Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

Network:

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

Modules:

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

ClosedNetwork Instance:

In diesem FB wird das entsprechende ClosedNetwork für die Analyse instanziiert. Intern verwendete Module werden in diesem Fall nicht weiter als Modul-FBs generiert.

Results:

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker im Binär oder Json Format, oder speichert die Daten lokal in eine Analytics-Binärdatei.

Analysis:

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

MAIN PRGs:

In dem Programm MAIN_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion verwaltet und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

In den Programmen MAIN_Analytics_Vx_Cx wird der FB Analysis aufgerufen. Die Programme sind jeweils einem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A_Reset definiert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics Version >= 3.1.0.0

6.7.2.4.1 FB_DataSource

Der FB DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

Syntax

Definition:

```
FUNCTION_BLOCK FB_T[n]_DataSource IMPLEMENTS I_DataSource, I_V[n]_Virtual_Input_Source
VAR_OUTPUT
eDataState: E_DataSourceState;
END_VAR
```

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 405]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
GetData [▶ 404]	Local	Methode zum Abrufen der Daten des spezifizierten Elements.
GetDataOversampling [▶ 405]	Local	Methode zum Abrufen der Oversampling Daten des spezifizierten Elements
NewDataAvailable [▶ 405]	Local	Methode zum Prüfen ob neue Daten verfügbar sind.
HistoricalCtrl [▶ 406]	Local	Methode zum Abrufen von historischen Daten
UpdateRecordList [▶ 406]	Local	Methode zum Aktualisieren der Historischen Aufnahmeliste.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.4.1.1 GetData

Syntax

```
METHOD GetData : BOOL
VAR_INPUT
    nElement : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Beispiel zu erhalten

Rückgabewert

Name	Typ	Beschreibung
GetData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

6.7.2.4.1.2 Call

Syntax

```
METHOD Call : BOOL
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.4.1.3 GetDataOversampling

Syntax

```
METHOD GetDataOversampling : BOOL
VAR_INPUT
nElement : UDINT;
nSample: : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Sampleelement zu erhalten
nSample	UDINT	Sample-ID, um das spezifische Sample zu erhalten

Rückgabewert

Name	Typ	Beschreibung
GetDataOversampling	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

6.7.2.4.1.4 NewDataAvailable

Syntax

```
METHOD NewDataAvailable : BOOL
VAR_INPUT
    nLastDataHandle : ULINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLastDataHandle	ULINT	Handle vom zuletzt abgeholten Datenpacket

Rückgabewert

Name	Typ	Beschreibung
NewDataAvailable	BOOL	Ist TRUE, wenn neue Daten verfügbar sind

6.7.2.4.1.5 HistoricalCtrl

Syntax

```
METHOD HistoricalCtrl : BOOL
VAR_INPUT
    stCtrl : REFERENCE TO ST_HMI_DataSourceCtrl;
    stHistStreamInfo : REFERENCE TO ST_HMI_DataSourceHist;
    stRecordInfo : REFERENCE TO ST_HMI_DataSourceHistRecordInfo;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stCtrl	ST_HMI_DataSourceCtrl	
stHistStreamInfo	ST_HMI_DataSourceHist	
stRecordInfo	ST_HMI_DataSourceHistRecordInfo	

Rückgabewert

Name	Typ	Beschreibung
HistoricalCtrl	BOOL	

6.7.2.4.1.6 UpdateRecordList

Syntax

```
METHOD UpdateRecordList : BOOL
VAR_INPUT
    stCtrl : REFERENCE TO ST_HMI_DataSourceCtrl;
    stHistStreamInfo : REFERENCE TO ST_HMI_DataSourceHist;
    sStreamSystemID : GUID;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stCtrl	ST_HMI_DataSourceCtrl	
stHistStreamInfo	ST_HMI_DataSourceHist	
sStreamSystemID	GUID	

Rückgabewert

Name	Typ	Beschreibung
UpdateRecordList	BOOL	

6.7.2.4.1.7 NextData

Syntax

```
METHOD NextData : BOOL
VAR_INPUT
END_VAR
```

6.7.2.4.1.8 NextDataOversample

Syntax

```
METHOD GetDataOversampling : BOOL
VAR_INPUT
nMaxOversampling : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nMaxOversampling	UDINT	Gibt den maximalen Oversamplingfaktor an.

Rückgabewert

Name	Typ	Beschreibung
NextDataOversample	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

6.7.2.4.2 FB_Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_[Network1]
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
VAR
    [module FBs]
END_VAR
```

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 397]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 397]	Local	Zurücksetzen des Netzwerks mit allen Submodulen.
ValueMapping [▶ 408]	Local	Zuordnung der Eingangswerte zu den verschiedenen Moduleingängen.
SetHMIValues [▶ 408]	Local	Optional: Mapping der Ein-/Ausgänge der Module zu der globalen HMI-Variable.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.4.2.1 ValueMapping

Syntax

```
METHOD ValueMapping : BOOL
VAR_INPUT
    pAnalysis : POINTER TO FB_Analysis;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pAnalysis	FB_Analysis	Instanz des Analyse FBs

Rückgabewert

Name	Typ	Beschreibung
ValueMapping	BOOL	

6.7.2.4.2.2 SetHMIValues

Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

Rückgabewert

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.4.2.3 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    pAnalysis: POINTER TO FB_Analysis;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pAnalysis	FB_Analysis	Instanz des Analyse FBs.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.4.2.4 Reset

Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.4.3 FB_Module

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_M[n]_[Module]
VAR_INPUT
    [module inputs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    [module outputs]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
Module inputs		Eingänge des ausgewählten Moduls.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.
Module outputs		Ausgänge des ausgewählten Moduls.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 410]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 410]	Local	Zurücksetzen des Moduls.
SetHMI [▶ 411]	Local	Optional: Setzt die Ein-/Ausgänge auf die globalen HMI-Strukturen.
GetHMI [▶ 411]	Local	Optional: Setzt die Eingänge der globalen HMI-Strukturen auf die Eingänge des Moduls

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.4.3.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source : I_V[n]_Virtual_Input_Source;
    [ipValueMapping_M2M : I_ValueMapping_M2M;]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.4.3.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.4.3.3 SetHMI

Syntax

```
METHOD GetHMI : BOOL
VAR_INPUT
    nContent : DINT
    pContent : PVOID
    bHMIReinit : BOOL
END_VAR
```

```
VAR
    pHMI_C[n]_[Content] : POINTER TO ST_HMI_C[n]_[Content];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nContent	DINT	HMI Content Index
pContent	PVOID	Pointer zur HMI Content Struktur
bHMIReinit	BOOL	Initialisieren der HMI Content Struktur

Rückgabewert

Name	Typ	Beschreibung
SetHMI	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.4.3.4 GetHMI (4.1)

Syntax

```
METHOD GetHMI : BOOL
VAR_INPUT
    nContent :  DINT
    pContent :  PVOID
END_VAR
```

```
VAR
    pHMI_C[n]_[Content] : POINTER TO ST_HMI_C[n]_[Content];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nContent	DINT	HMI Content Index
pContent	PVOID	Pointer zur HMI Content Struktur

Rückgabewert

Name	Typ	Beschreibung
GetHMI	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.4.4 FB_Analysis

Im FB Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

Syntax

Definition:

```
FUNCTION_BLOCK FB_Analysis
VAR_INPUT
    ipV[n]_VirtualInputs: I_V[n]_Virtual_Input_Source;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
VAR
    [network FBs]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipV[n]_VirtualInputs	I_V[n]_Virtual_Input_Source	Daten für die Analyse von der spezifischen Datenquelle

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 412]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 413]	Local	Zurücksetzen der gesamten Analyse.
ResultStream [▶ 413]	Local	Optional: Wenn ein Ergebnis-Stream erstellt werden muss.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.4.4.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source: I_V[n]_Virtual_Input_Source;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.4.4.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR_IN_OUT
    stReset: ST_AnalysisReset;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stReset	ST_AnalysisReset	Struktur, um zu definieren, welches Modul oder Netzwerk zurückgesetzt werden soll.

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.4.4.3 ResultStream

Syntax

```
METHOD ResultStream : BOOL
VAR_INPUT
    ipResults: I_Results;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipResults	I_Results	Schnittstellenzeiger auf den FB Results

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.4.5 FB_Results

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

Syntax

Definition:

```
FUNCTION_BLOCK FB_Results
VAR_OUTPUT
    stResults: ST_Results;
END_VAR
```

Ausgänge

Name	Typ	Beschreibung
stResults	ST_Results	Ergebnisstruktur, die alle Elemente des Ergebnis-Streams enthält.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 414]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
AddResult [▶ 414]	Local	Fügt dem Ergebnis-Stream ein Muster hinzu.
SendResults [▶ 415]	Local	Sendet alle gepufferten Muster des Ergebnis-Streams.
Release [▶ 415]	Local	Schließt den Stream oder die Datei des Ergebnis-Streams.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.4.5.1 Call

Syntax

```
METHOD Call : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.4.5.2 AddResult

Syntax

```
METHOD AddResult : BOOL
VAR_INPUT
    tTimestamp: ULINT;
    stSample: ST_Results;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
tTimestamp	ULINT	Zeitstempel des Samples
stSample	ST_Results	Sample-Struktur

Rückgabewert

Name	Typ	Beschreibung
AddResult	BOOL	

6.7.2.4.5.3 SendResults

Syntax

```
METHOD SendResults : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
SendResults	BOOL	

6.7.2.4.5.4 Release

Syntax

```
METHOD Release : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Release	BOOL	

6.7.2.4.6 MAIN_Analytics

Im Programm MAIN_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion gemanaged und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

Syntax

Definition:

```
PROGRAM MAIN_Analytics
VAR_INPUT
    sCurrentStreamSystemID: GUID;
    stHistStreamInfo: REFERENCE TO ST_HMI_DataSourceHist;
    stHistRecordInfo: REFERENCE TO ST_HMI_DataSourceHistRecordInfo;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.4.7 MAIN_Analytics_C[n]

Im Programm MAIN_Analytics_C[n] wird der FB Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A_Reset definiert.

Das Mapping der HMI Werte wird in der Aktion A_MapToHMI vorgenommen.

Syntax

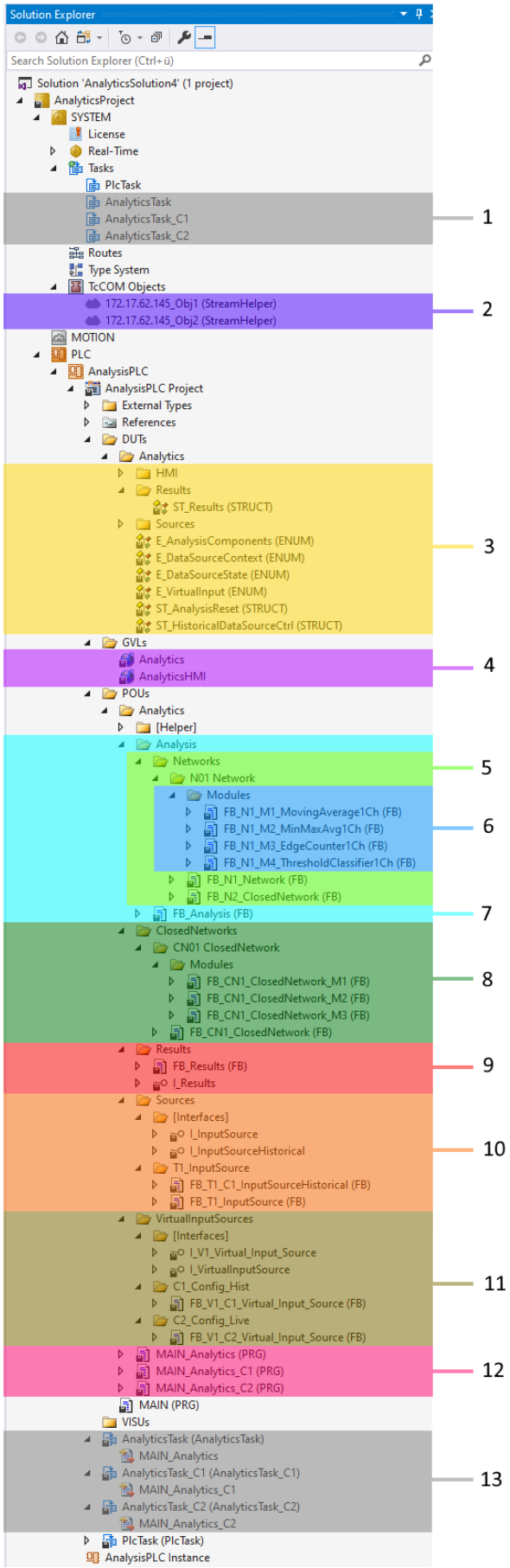
Definition:

```
PROGRAM MAIN_Analytics_C[n]
VAR_INPUT
    stReset: ST_AnalysisReset;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    nLastDataHandle: ULINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.5 Codeversion 5.0



1: Tasks

Für die Analytics-Analyse und für jede Konfiguration einer Virtual Input Source wird ein separater Task erzeugt.

2: StreamHelper

Für jede Datenquellen vom Typ MQTT-Binärstrom, erstellt die Codegenerierung eine Instanz eines Stream Helper-Objekts, um die eingehenden Binärstrommuster zu verarbeiten.

3: DataTypes

Die Datentypen werden für die Analyse erstellt. Sie enthalten STRUCTs für die Reset-Funktion oder Ergebnisverarbeitung und ENUMs, um die verschiedenen Komponenten auszuwählen.

4: GVLs

Um bequem Ein- und Ausgänge der Module mit dem HMI Dashboard zu mappen, werden ausgewählte Variablen als globale Variable generiert. Zusätzlich werden die Data Source Baustein Instanzen und verschiedene Parameter als Globale Variable generiert.

5: Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

6: Modules

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

7: Analysis

Im FB-Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

8: ClosedNetwork

Die ClosedNetwork-FBs werden einmal mit allen Subnetzwerken und Modulen generiert. Sie können in der Analyse mehrmalig instanziiert werden. Auf diese Weise kann der generierte Code reduziert und vereinfacht werden.

9: Results

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB-Results und streamt die ausgewählten Variablen zum Message-Broker im Binär oder Json Format, oder speichert die Daten lokal in eine Analytics-Binärdatei.

10: InputSource

Der FB-InputSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

11: VirtualInputSource

Die VirtualInputSource Interfaces abstrahieren die InputSource Symbole von der Analyse.

12: MAIN PRGs

In dem Programm MAIN_Analytics werden die DataSource FBs aufgerufen, die Reset Funktion verwaltet und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

In den Programmen MAIN_Analytics_Vx_Cx wird der FB-Analysis aufgerufen. Die Programme sind jeweils einem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A_Reset definiert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics Version >= 3.1.0.0

6.7.2.5.1 FB_Analysis

Im FB-Analysis ist die gesamte Analyseroutine definiert. Alle konfigurierten Netzwerke mit ihren Modulen und Fehlerbehandlung werden erstellt.

Syntax

Definition:

```
FUNCTION_BLOCK FB_Analysis
VAR_INPUT
    nConfigurationID: INT;
    ipV[n]_VirtualInputs: I_V[n]_Virtual_Input_Source;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
VAR
    [network FBs]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nConfigurationID	INT	Konfigurations Index
ipV[n]_VirtualInputs	I_V[n]_Virtual_Input_Source	Daten für die Analyse der spezifischen Datenquelle

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
nContext	DWORD	Get	Kontext Index

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 420]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 420]	Local	Zurücksetzen der gesamten Analyse.
ResultStream [▶ 421]	Local	Optional: Wenn ein Ergebnis-Stream erstellt werden muss.
SetHMIValues [▶ 421]	Local	Methode zum befüllen der HMI-Strukturen
GetHMIValues [▶ 421]	Local	Methode zum Setzen der Parameter aus der HMI in die Analyse

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.5.1.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source: I_V[n]_Virtual_Input_Source;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.5.1.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR_IN_OUT
    stReset: ST_AnalysisReset;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
stReset	ST_AnalysisReset	Struktur, um zu definieren, welches Modul oder Netzwerk zurückgesetzt werden soll.

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.5.1.3 ResultStream

Syntax

```
METHOD ResultStream : BOOL
VAR_INPUT
    ipResults: I_Results;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipResults	I_Results	Schnittstellenzeiger auf den FB-Results

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.5.1.4 SetHMIValues

Syntax

```
METHOD SetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

Rückgabewert

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.5.1.5 GetHMIValues

Syntax

```
METHOD GetHMIValues : BOOL
VAR_INPUT
    pHMI_N[n]_[Network1] : POINTER TO ST_HMI_N[n]_[Network1];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pHMI_N[n]_[Network1]	POINTER TO ST_HMI_N[n]_[Network1]	Zeiger zur globalen HMI-Struktur

Rückgabewert

Name	Typ	Beschreibung
SetHMIValues	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.5.2 FB_Network

Alle Module werden in einem spezifischen Netzwerk sortiert, um eine bessere Übersicht und Struktur der konfigurierten Analyse zu erreichen.

Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_[Network1]
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
END_VAR
VAR
    [module FBs]
END_VAR
```

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 397]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 397]	Local	Zurücksetzen des Netzwerks mit allen Submodulen.
ValueMapping [▶ 423]	Local	Zuordnung der Eingangswerte zu den verschiedenen Moduleingängen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.5.2.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    pAnalysis: POINTER TO FB_Analysis;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pAnalysis	FB_Analysis	Instanz des Analyse FBs.

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.5.2.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.5.2.3 ValueMapping

Syntax

```
METHOD ValueMapping : BOOL
VAR_INPUT
    pAnalysis : POINTER TO FB_Analysis;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pAnalysis	FB_Analysis	Instanz des Analyse FBs

Rückgabewert

Name	Typ	Beschreibung
ValueMapping	BOOL	

6.7.2.5.3 FB_Module

Die Modul-FBs enthalten alle Eingänge und Ausgänge der konfigurierten Module aus dem Workbench-Konfigurator. Es ist auch möglich, die Module während der Laufzeit neu zu konfigurieren. Dazu müssen Sie lediglich den Parameter ändern und dann den Neukonfigurierungsprozess mit einer steigenden Flanke am INPUT bReconfigure starten.

Syntax

Definition:

```
FUNCTION_BLOCK FB_N[n]_M[n]_[Module]
VAR_INPUT
    [module inputs]
END_VAR
VAR_INPUT PERSISTENT
    [module persistent parameter inputs]
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    [module outputs]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
Module inputs		Eingänge des ausgewählten Moduls.

Ausgänge

Name	Typ	Beschreibung
bError	BOOL	Wird TRUE, sobald eine Fehlersituation auftritt.
ipTcResult	I_TcMessage	Nachrichtenschnittstelle des TwinCAT 3 EventLoggers, die nähere Informationen über den Rückgabewert bietet.
Module outputs		Ausgänge des ausgewählten Moduls.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
nContext	DWORD	Get	Kontext Index

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 425]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
Reset [▶ 425]	Local	Zurücksetzen des Moduls.
SetHMI [▶ 425]	Local	Optional: Setzt die Ein-/Ausgänge auf die globalen HMI-Strukturen.
GetHMI [▶ 426]	Local	Optional: Setzt die Eingänge der globalen HMI-Strukturen auf die Eingänge des Moduls

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.5.3.1 Call

Syntax

```
METHOD Call : BOOL
VAR_INPUT
    ipVirtual_Input_Source : I_V[n]_Virtual_Input_Source;
    [ipValueMapping_M2M : I_ValueMapping_M2M;]
END_VAR
```

Eingänge

Name	Typ	Beschreibung
ipVirtual_Input_Source	I_V[n]_Virtual_Input_Source	Daten für die Analyse
ipValueMapping_M2M	I_ValueMapping_M2M	Optional: Notwendig für das Mapping von Werten zwischen Modulen

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.5.3.2 Reset

Syntax

```
METHOD Reset : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Reset	BOOL	Ist TRUE, wenn abgeschlossen.

6.7.2.5.3.3 SetHMI

Syntax

```
METHOD SetHMI : BOOL
VAR_INPUT
    nContent : DINT
    pContent : PVOID
    bHMIReinit : BOOL
END_VAR
```

```
VAR
    pHMI_C[n]_[Content] : POINTER TO ST_HMI_C[n]_[Content];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nContent	DINT	HMI Content Index
pContent	PVOID	Pointer zur HMI Content Struktur
bHMIReinit	BOOL	Initialisieren der HMI Content Struktur

Rückgabewert

Name	Typ	Beschreibung
SetHMI	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.5.3.4 GetHMI

Syntax

```
METHOD GetHMI : BOOL
VAR_INPUT
    nContent :  DINT
    pContent :  PVOID
END_VAR
```

```
VAR
    pHMI_C[n]_[Content] : POINTER TO ST_HMI_C[n]_[Content];
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nContent	DINT	HMI Content Index
pContent	PVOID	Pointer zur HMI Content Struktur

Rückgabewert

Name	Typ	Beschreibung
GetHMI	BOOL	Ist TRUE, wenn abgeschlossen

6.7.2.5.4 FB_InputSource

Der FB-DataSource verwaltet den Empfang der Eingangswerte der verschiedenen Quellen. In der OUTPUT-Deklaration finden Sie alle konfigurierten Eingänge.

Syntax

Definition:

```
FUNCTION_BLOCK FB_T[n]_InputSource IMPLEMENTS I_InputSource
VAR
END_VAR
```

Schnittstellen

Typ	Beschreibung
I_InputSource	Schnittstelle zur Kommunikation mit einer Datenquelle

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 427]	Local	Methode für Hintergrundkommunikation mit dem TwinCAT-Treiber. Die Methode muss zyklisch aufgerufen werden.
GetData [▶ 427]	Local	Methode zum Abrufen der Daten des spezifizierten Elements.
NewDataAvailable [▶ 428]	Local	Methode zum Prüfen ob neue Daten verfügbar sind.
AddClient [▶ 428]	Local	Methode zum Hinzufügen von Daten Clients
ClientDone [▶ 429]	Local	Methode zum Signalisieren das der Client alle Daten empfangen hat.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
bReadNewData	BOOL	Get	.
eDataState	E_DataSourceState	Get	
nDataHandle	ULINT	Get	
nElements	UDINT	Get	
nMaxOversamplingFactor	UDINT	Get	

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.5.4.1 Call

Syntax

```
METHOD Call : BOOL
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.5.4.2 GetData

Syntax

```
METHOD GetData : BOOL
VAR_INPUT
    nElement     : UDINT;
    pInputs      : PVOID;
    nInputsSize  : UDINT
END_VAR
VAR_OUTPUT
    nTimestamp   : ULINT;
    nContext     : DWORD;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nElement	UDINT	Element-ID, um das spezifische Beispiel zu erhalten
pInputs	PVOID	Pointer zur Datenstruktur
nInputsSize	UDINT	Größe der Datenstruktur

Ausgänge

Name	Typ	Beschreibung
nTimestamp	ULINT	Daten Zeitstempel
nContext	DWORD	Daten Kontext

Rückgabewert

Name	Typ	Beschreibung
GetData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

6.7.2.5.4.3 NewDataAvailable

Syntax

```
METHOD NewDataAvailable : BOOL
VAR_INPUT
    nLastDataHandle : ULINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nLastDataHandle	ULINT	Handle vom zuletzt abgeholten Datenpacket

Rückgabewert

Name	Typ	Beschreibung
NewDataAvailable	BOOL	Ist TRUE, wenn neue Daten verfügbar sind

6.7.2.5.4.4 AddClient

Syntax

```
METHOD AddClient : BOOL
VAR_OUTPUT
    nClientID    : DWORD;
END_VAR
```

Ausgänge

Name	Typ	Beschreibung
nClientID	LWORD	Client-ID

6.7.2.5.4.5 ClientDone

Syntax

```
METHOD ClientDone : BOOL
VAR_OUTPUT
    nClientID    : DWORD;
END_VAR
```

Ausgänge

Name	Typ	Beschreibung
nClientID	LWORD	Client-ID

6.7.2.5.5 FB_VirtualInputSource

Der FB VirtualInputSource abstrahiert die InputSourcen für die verschiedenen Analyse Konfigurationen. Die in der Analytics Workbench konfigurierten virtuellen Eingänge werden hier gesetzt.

Syntax

Definition:

```
FUNCTION_BLOCK FB_V[n]_C[n]_VirtualInputSource IMPLEMENTS I_VirtualInputSource,
I_V[n]_VirtualInputSource
VAR
END_VAR
```

Schnittstellen

Typ	Beschreibung
I_VirtualInputSource	Schnittstelle zur Kommunikation mit einer Datenquelle
I_V[n]_VirtualInputSource	Schnittstelle die alle definierten virtuellen Eingänge breitstellt

Methoden

Name	Definitionsort	Beschreibung
SourceSync [▶ 430]	Local	Methode zum Synchronisieren von mehreren FB-Instanzen
NextData [▶ 430]	Local	Methode zum entgegenzunehmen des nächsten Datensatzes
Done [▶ 430]	Local	Methode zum Signalisieren, dass das gesamte Datenpaket fertig bearbeitet wurde.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
bEndOfData	BOOL	Get	Signalisiert das Ende des Datenpakets
dtTimestamp	DCTIMESTRUCT	Get	Zeitstempel des aktuell entgegengenommenen Datensatz
nDataHandle	ULINT	Get	Daten-Handle
nContext	DWORD	Get	Daten-Kontext
tTimestamp	ULINT	Get	Zeitstempel des aktuell entgegengenommenen Datensatzes

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.5.5.1 SourceSync

Syntax

```
METHOD SourceSync : BOOL
```

6.7.2.5.5.2 NextData

Syntax

```
METHOD NextData : BOOL
VAR_INPUT
    nMaxOversampling    : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
nMaxOversampling	UDINT	Gibt den maximalen Oversampling-Wert an

Rückgabewert

Name	Typ	Beschreibung
NextData	BOOL	Ist TRUE, wenn ein neues Element ausgewählt ist

6.7.2.5.5.3 Done

Syntax

```
METHOD Done : BOOL
VAR_INPUT
END_VAR
```

6.7.2.5.6 FB_Results

Wenn Ergebnisse der Analyse gespeichert oder gestreamt werden müssen, verwaltet dies der FB-Results und streamt die ausgewählten Variablen zum Message-Broker oder speichert die Daten in die Analytics-Binärdatei.

Syntax

Definition:

```
FUNCTION_BLOCK FB_Results
VAR_OUTPUT
    nTimestamp: ULINT;
    stResults: ST_Results;
END_VAR
```

Ausgänge

Name	Typ	Beschreibung
nTimestamp	ULINT	Zugehöriger Zeitstempel der Ergebnisstrukturdaten
stResults	ST_Results	Ergebnisstruktur, die alle Elemente des Ergebnis-Streams enthält.

Eigenschaften

Name	Typ	Zugriff	Beschreibung
bInitialized	BOOL	Get	Gibt an ob der Bausteine ordnungsgemäß initialisiert wurde
nMaxSamples	INT	Get	Maximale Anzahl an gepufferten Ergebnissen
nResultCount	INT	Get	Aktuelle Anzahl an gepufferten Ergebnissen

Methoden

Name	Definitionsort	Beschreibung
Call [▶ 431]	Local	Methode für Hintergrundkommunikation. Die Methode muss zyklisch aufgerufen werden.
AddResult [▶ 431]	Local	Fügt dem Ergebnis-Stream ein Muster hinzu.
SendResults [▶ 432]	Local	Sendet alle gepufferten Muster des Ergebnis-Streams.
Release [▶ 432]	Local	Schließt den Stream oder die Datei des Ergebnis-Streams.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.5.6.1 Call

Syntax

```
METHOD Call : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Call	BOOL	

6.7.2.5.6.2 AddResult

Syntax

```
METHOD AddResult : BOOL
VAR_INPUT
    tTimestamp: ULINT;
    stSample: ST_Results;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
tTimestamp	ULINT	Zeitstempel des Samples
stSample	ST_Results	Sample-Struktur

Rückgabewert

Name	Typ	Beschreibung
AddResult	BOOL	

6.7.2.5.6.3 SendResults

Syntax

```
METHOD SendResults : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
SendResults	BOOL	

6.7.2.5.6.4 Release

Syntax

```
METHOD Release : BOOL
VAR
END_VAR
```

Rückgabewert

Name	Typ	Beschreibung
Release	BOOL	

6.7.2.5.7 MAIN_Analytics

Im Programm MAIN_Analytics werden die InputSource FBs aufgerufen, die Reset Funktion gemanaged und ggf. die Werte mit dem HMI Dashboard gemapped. Das Programm ist einem separaten Task zugewiesen.

Syntax

Definition:

```
PROGRAM MAIN_Analytics
VAR
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.7.2.5.8 MAIN_Analytics_C[n]

Im Programm MAIN_Analytics_C[n] wird der FB-Analysis aufgerufen. Das Programm ist dem separaten Task zugewiesen.

Es ist auch möglich, einzelne Module, ganze Netzwerke oder alle definierten Netzwerke mit nur einem steigendem Flag zurückzusetzen. Zunächst müssen Sie die Komponente auswählen, die zurückgesetzt werden soll. Dann startet eine steigende Flanke am INPUT bReset den Reset-Vorgang.

Alle Reset-Aufrufe sind in der Aktion A_Reset definiert.

Das Handling der „InputSourcen“ wird in der Aktion A_InputSources durchgeführt.

Das Mapping der HMI-Werte wird in der Aktion A_MapToHMI vorgenommen.

Syntax

Definition:

```
PROGRAM MAIN_Analytics_C[n]
VAR_INPUT
    stReset: ST_AnalysisReset;
END_VAR
VAR_OUTPUT
    bError: BOOL;
    ipTcResult: I_TcMessage;
    nAnalysisResultsTimestamp: ULINT;
    stAnalysisResults: ST_Results;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4024.0	PC oder CX (x64, x86)	Tc3_Analytics

6.8 HMI One-Click Dashboard

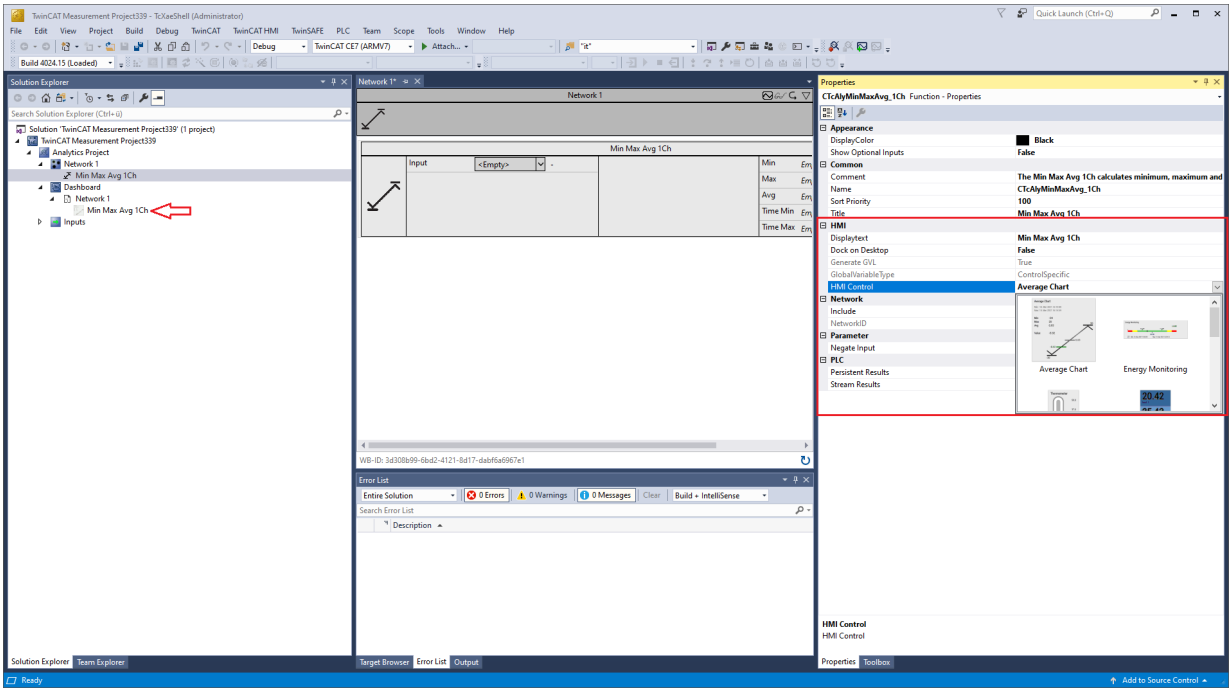
Es ist möglich, ein HMI Dashboard mit HMI Controls für alle Module und Parameter, die im TwinCAT Analytics Workbench Configurator konfiguriert sind, automatisch zu generieren. Das HMI Dashboard basiert auf der TwinCAT HMI und visualisiert die SPS-Daten aus dem Runtime-Deployment [▶ 365].

Das automatisch generierte One-Click Dashboard ist nur mit der neuen HMI Version 1.12 verfügbar. Um die Analytics HMI Controls zu verwenden, ist eine Analytics Runtime-Lizenz erforderlich.

✓ Nach der Konfiguration Ihres Analytics Workbench-Projekts kann für jeden Algorithmus ein HMI Control ausgewählt werden.

1. Öffnen Sie das Fenster **Properties** des Moduls und wählen Sie ein **HMI Control** aus der DropDown-Liste aus. Sie können den Anzeigetext für den Titel im HMI Dashboard ändern (Displaytext). Ebenfalls können Sie auswählen, ob das Control an die Startseite angedockt werden soll (Dock on Desktop). Im

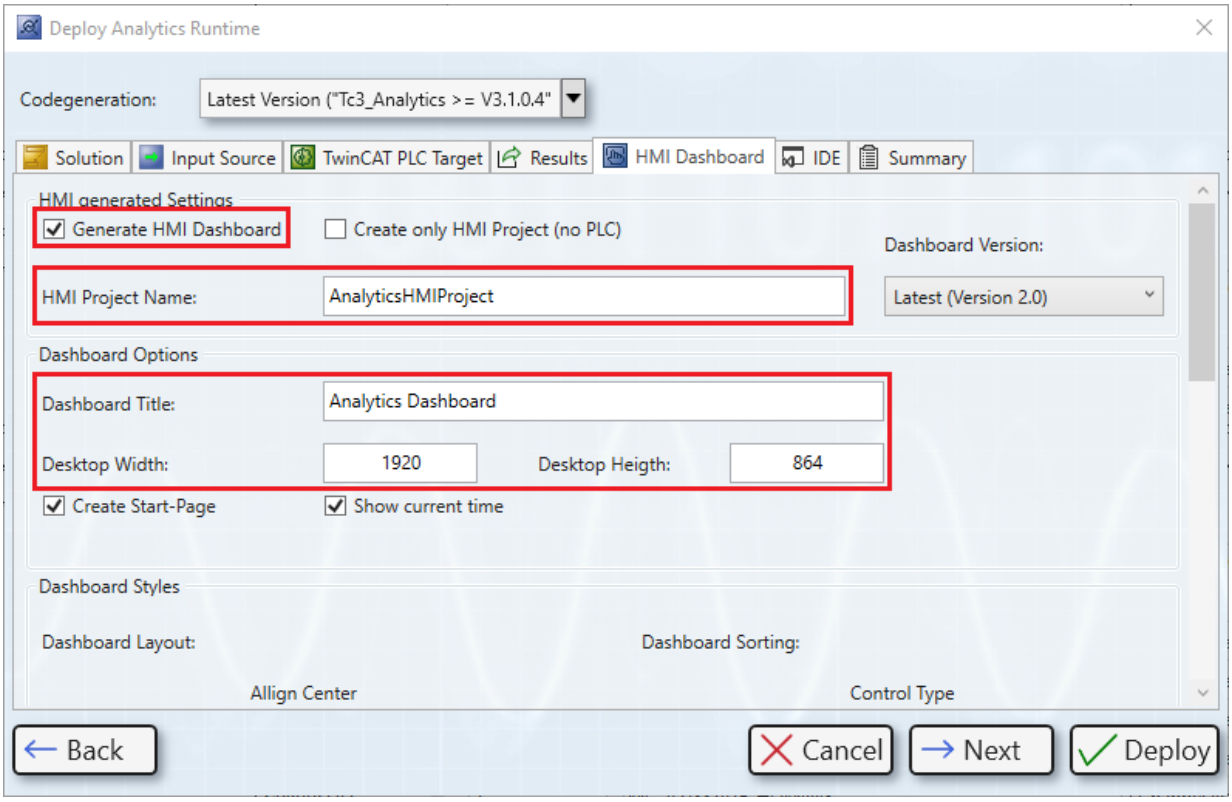
Solution Explorer werden alle zu generierenden Controls unter dem Dashboard Aufbau und Inhalte im Analytics Projekt verwalten [▶ 436] gespeichert



2. Nach Abschluss der Konfiguration klicken Sie auf den Befehl **Deploy Analytics Runtime** im Kontextmenü. Der Deploy-Assistent startet und es ist möglich, die gesamte benötigte Konfiguration für den Einsatz schrittweise einzurichten.

   ⇨ Auf dem Karteireiter **HMI Dashboard** können Sie Ihr HMI Dashboard konfigurieren.

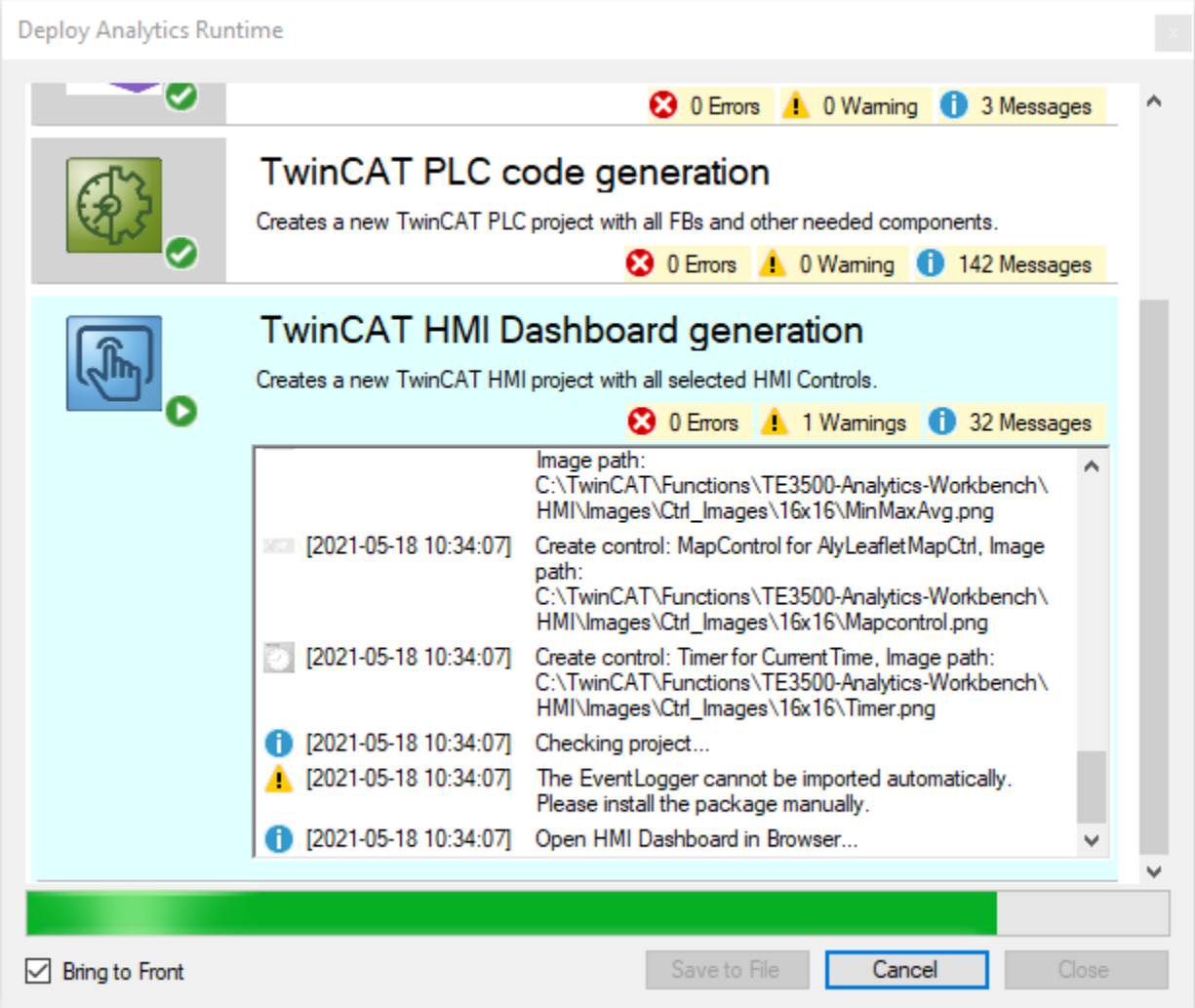
3. Aktivieren Sie die Checkbox **Generate HMI Dashboard**. Es ist auch möglich, nur ein HMI-Projekt ohne eine SPS zu erstellen. Des Weiteren können Sie einen **HMI Project Name** für das Dashboard vergeben und einen Dashboard Title sowie die Desktop Height und Desktop Width festlegen, um ein maßgeschneidertes Dashboard zu generieren, das zu Ihren Bedürfnissen passt. Die übrigen Konfigurationen werden unter Dashboard-Konfiguration [▶ 468] erläutert.



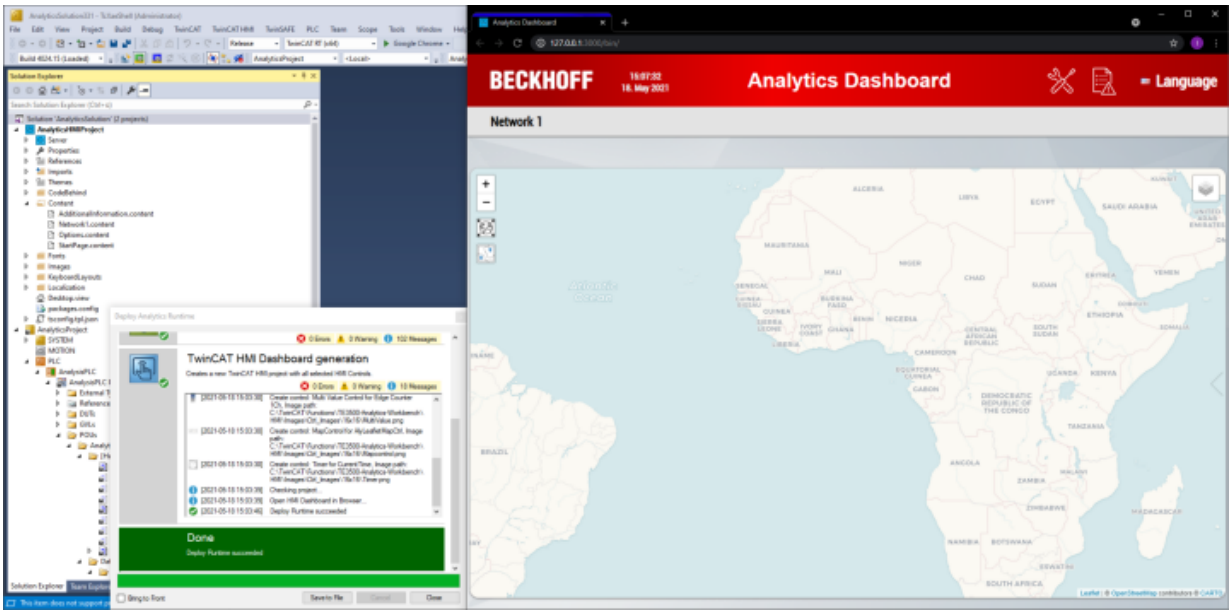
⇨ Wie üblich zeigt der letzte Karteireiter Ihnen alle Einstellungen, die Sie für die Generierung vorgenommen haben.

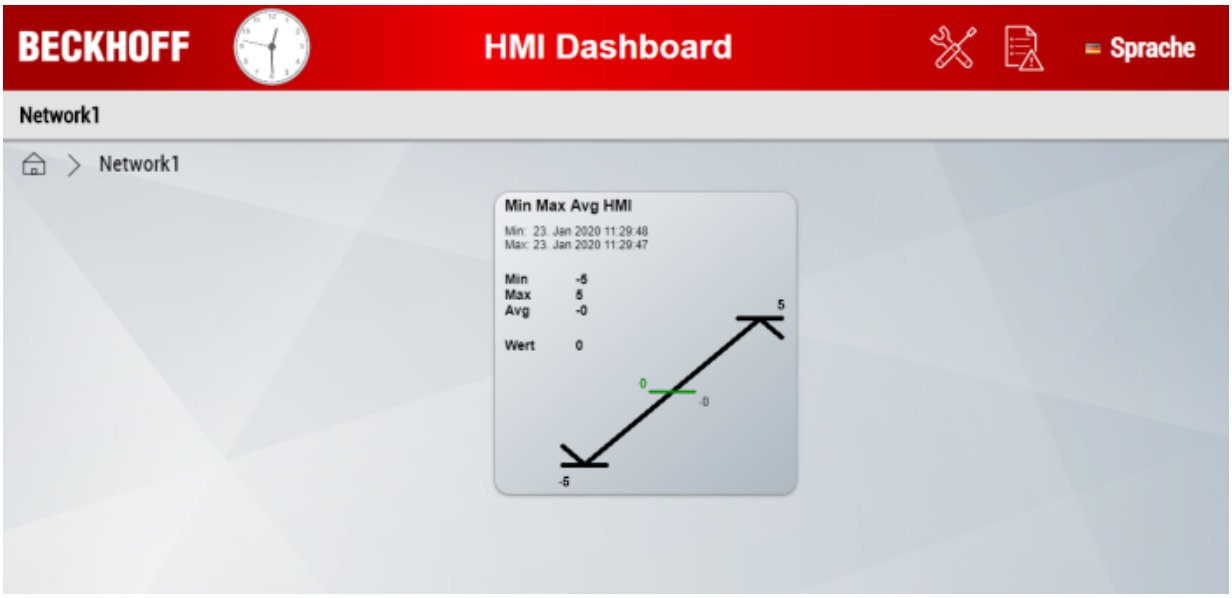
4. Nun können Sie den Generierungsprozess starten, indem Sie auf **Deploy** klicken.

⇨ Die HMI-Generierung beginnt direkt nach dem Runtime-Deployment (falls ausgewählt). Im Übersichtsfenster wird jeder Schritt während des Generierungsprozesses auch für die Generierung des HMI Dashboards angezeigt.



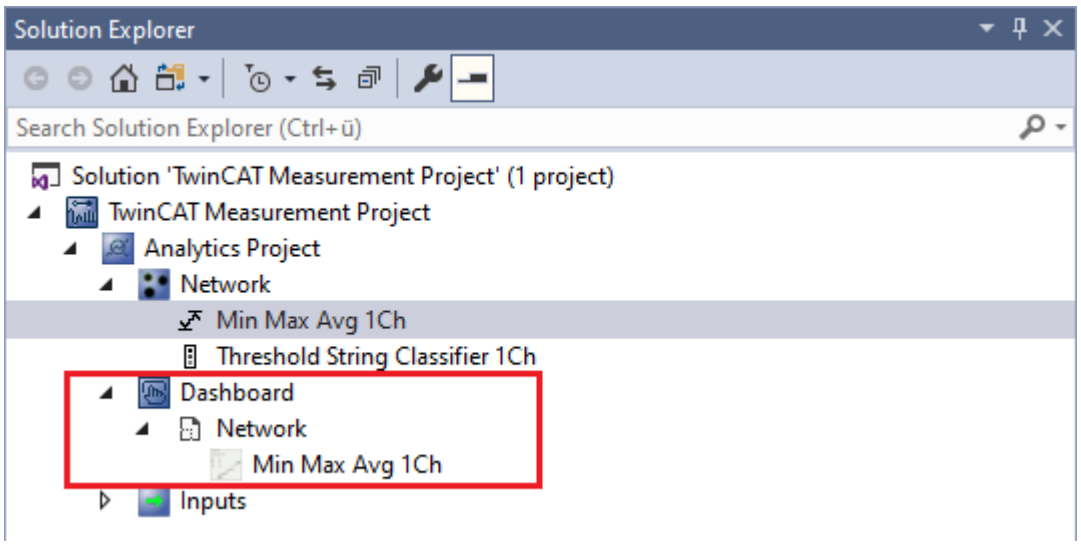
⇨ Das Dashboard wird automatisch in Ihrem Standard-Browser geöffnet.



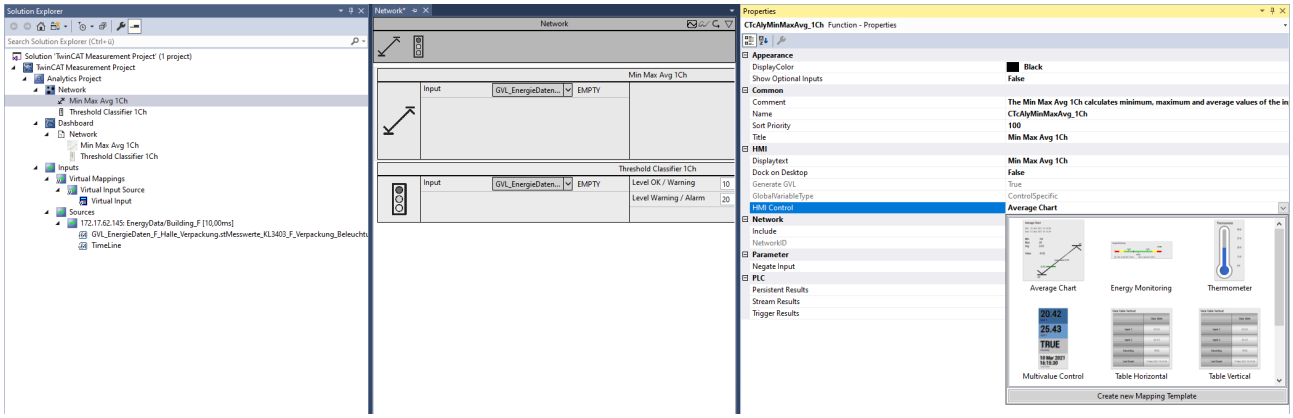


6.8.1 Dashboard Aufbau und Inhalte im Analytics Projekt verwalten

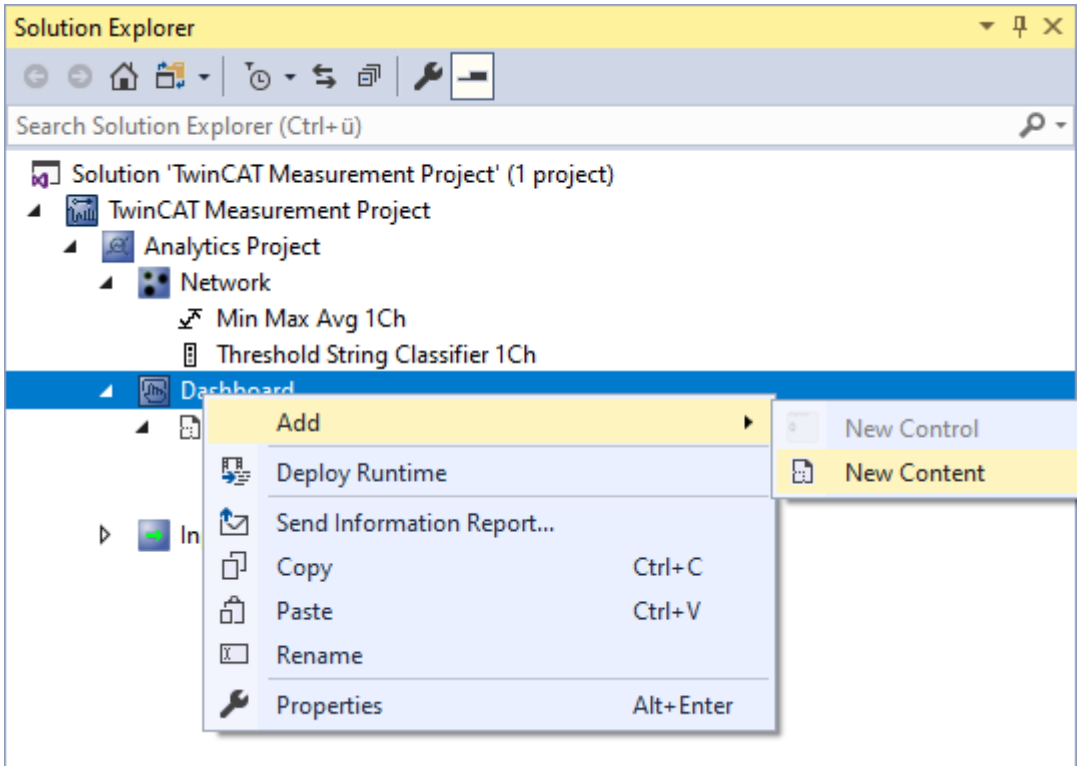
Zu einem Analytics Projekt liegt eine Konfiguration von HMI Contents (Seiten) und HMI Controls (Anzeigeelemente) vor, die bei der Dashboard Generierung erstellt werden. Diese Konfiguration ist über den **Dashboard-Knoten** im Solution Explorer ersichtlich und veränderbar. Sie können die Contents und Controls zu jeder Zeit umbenennen, in andere Contents via Drag & Drop verschieben, Kopieren (**Ctrl-C, Ctrl-V**) und löschen. Ein Control kann ebenfalls über einen Doppelklick editiert werden.



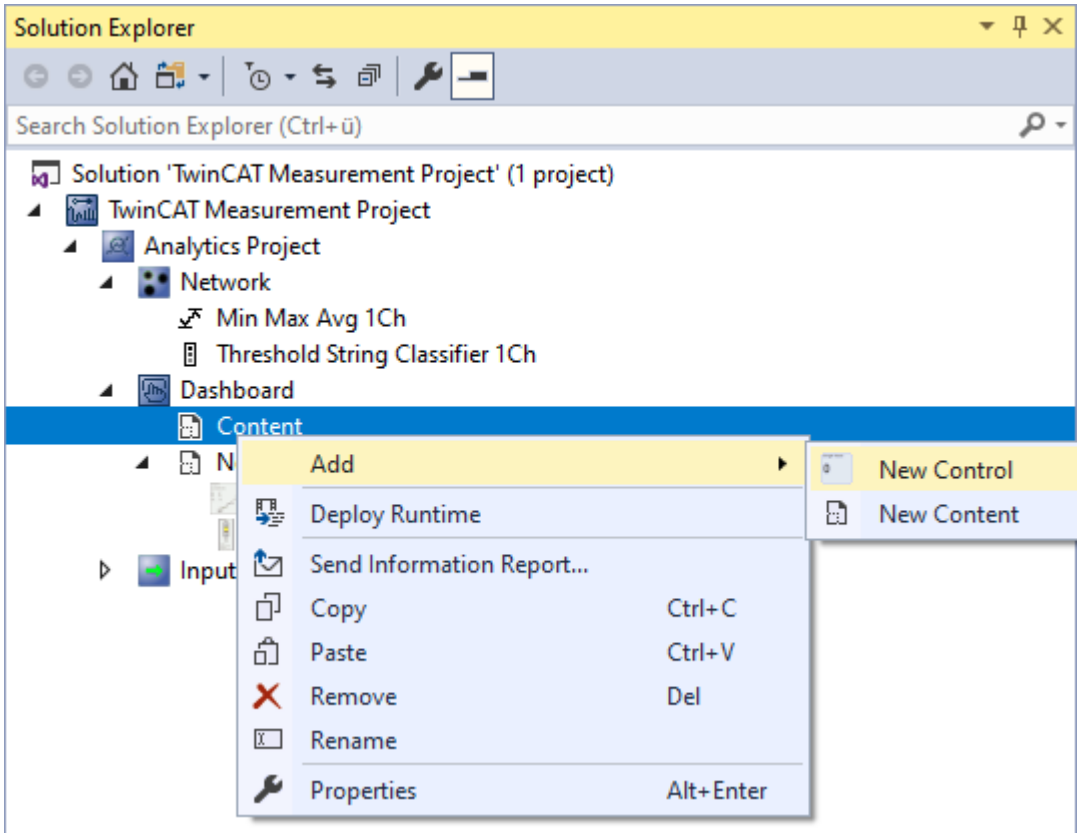
Jedes Analytics Modul besitzt bestehende Control-Mappings, welche über das Fenster **Properties** auswählbar sind (diese können Sie auch über „Create new Mapping Template“ bzw. über den Angepasste und eigene Controls verwenden [▶ 446] (Punkt 2) selbst erstellen/bearbeiten). Sobald ein Control für ein Modul ausgewählt wird, wird dieses unter dem Knoten **Dashboard** gelistet.



Auf dem Knoten **Dashboard** und dem Knoten **Content** können Sie via Rechtsklick neue Contents anlegen.



Ebenso können Sie **neue Controls** mit einen Rechtsklick auf einen Knoten **Content** hinzufügen.



Es öffnet sich der **Analytics Dashboard Wizard**. In diesem Wizard werden Sie Schritt für Schritt durch die Konfiguration zum Hinzufügen eines Controls geleitet.

Analytics Dashboard Wizard

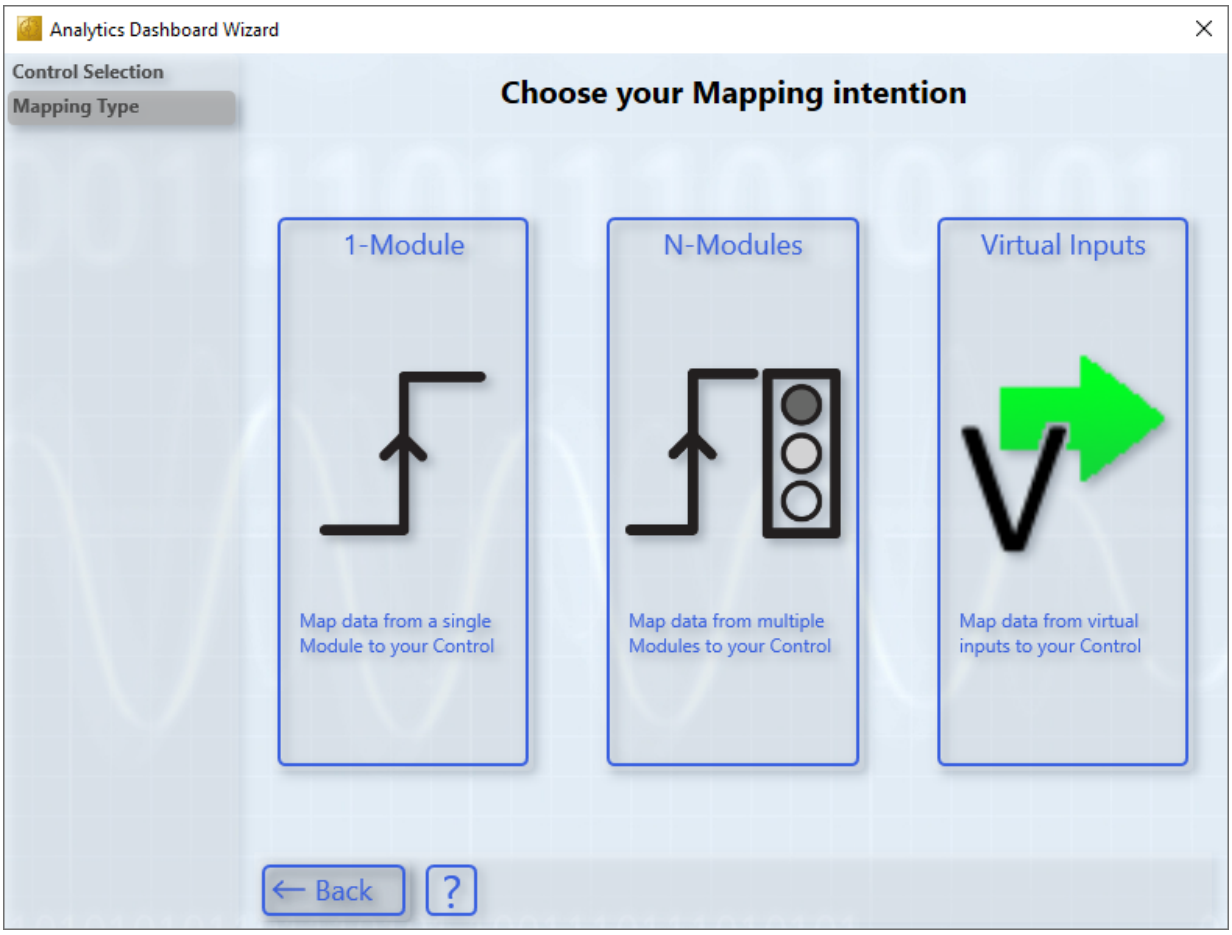
Machen Sie sich mit Wizard vertraut, indem Sie ein Control hinzufügen (Control über das **Properties** Fenster vom Modul auswählen). Anschließend können Sie die gesamte Konfiguration im Wizard nachsehen, indem Sie einen Doppelklick auf das Control machen. Dort können Sie die einzelnen Konfigurationsschritte nachvollziehen.

Analytics Dashboard Wizard

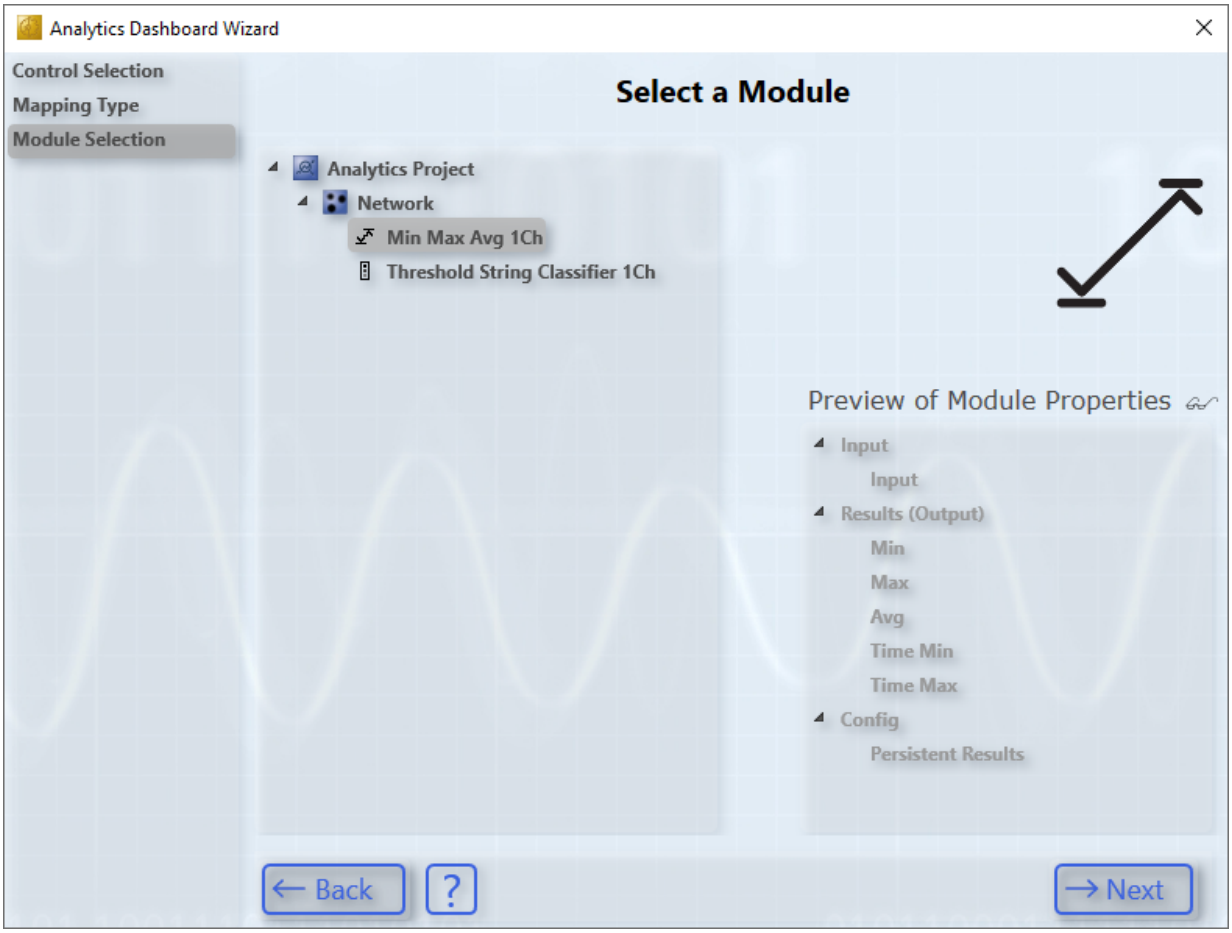
Wählen Sie ein Control aus. Auf der linken Seite sind alle verfügbaren Controls gelistet. Standardmäßig werden nur Controls gelistet, die Algorithmus unspezifisch sind. Über das Abhaken der Checkbox **Show only default Controls** sind alle Controls verfügbar.

1. Wählen Sie **Single Value**, um einen einzelnen Wert im Dashboard anzuzeigen. Drücken Sie auf den Button **Next,** um fortzufahren.

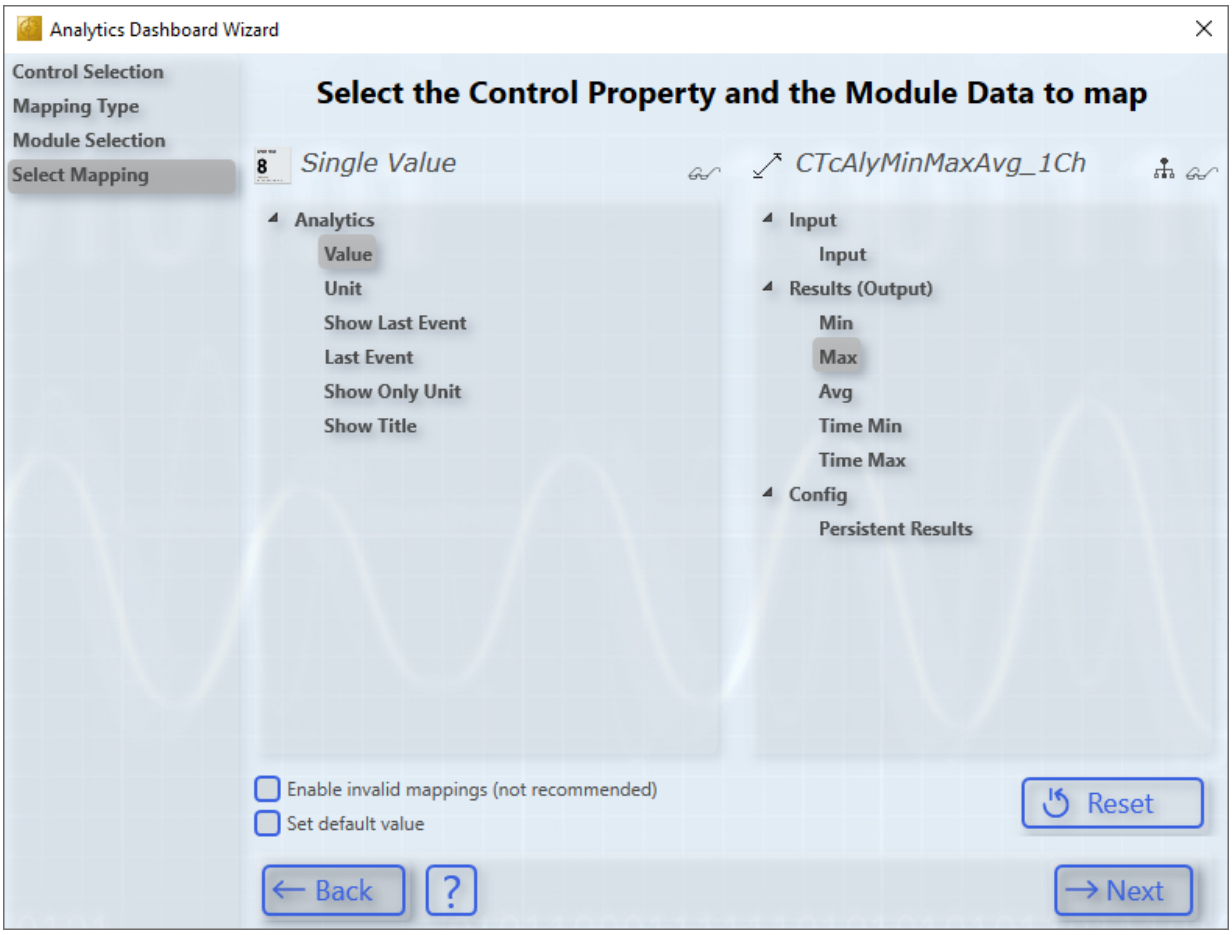
2. Sie können nun Daten von einem Modul (**1-Module**), mehreren Modulen (**N-Modules**) oder von Virtuellen Inputs (**Virtual Inputs**) verknüpfen. Wählen Sie für dieses Beispiel **1-Module** aus.



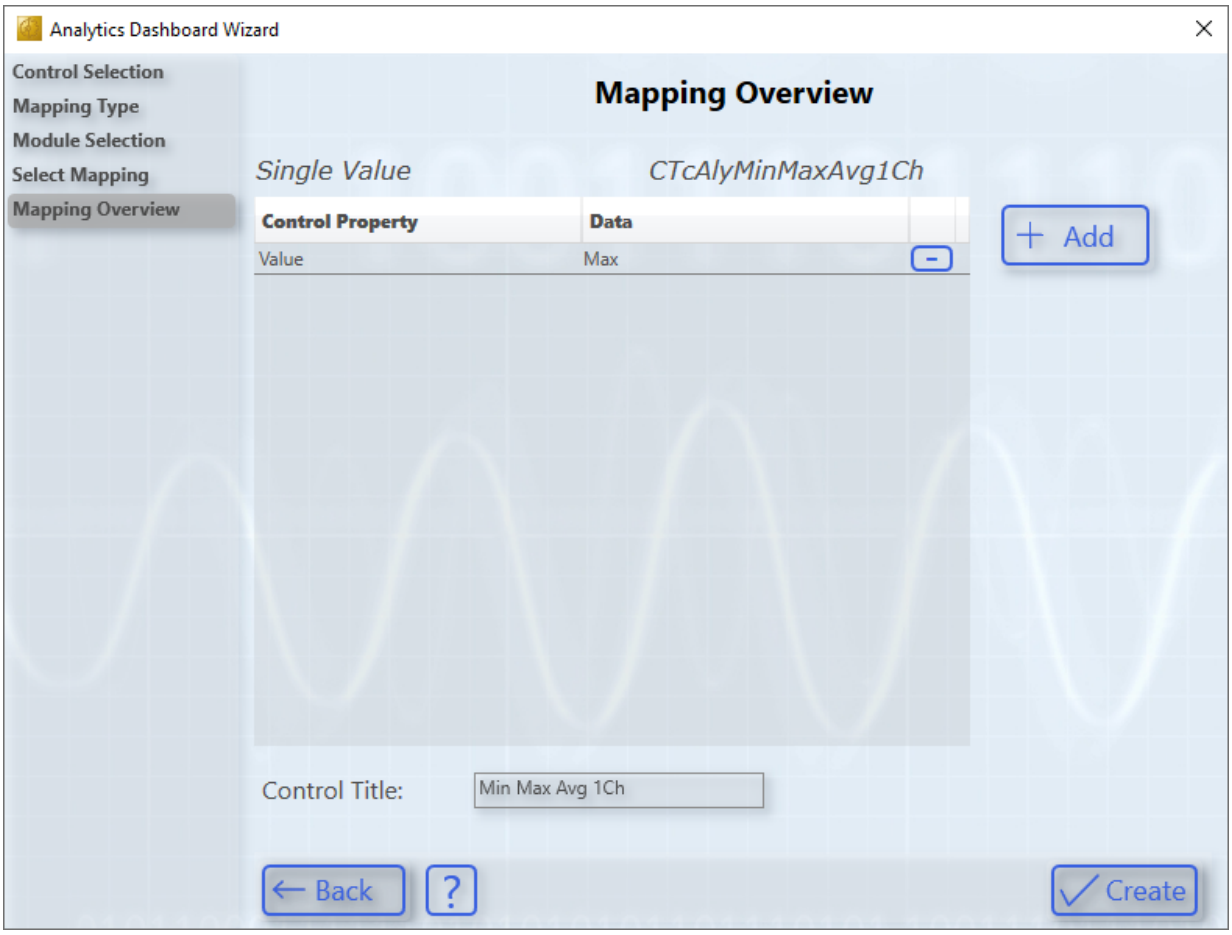
3. Es werden alle Module aus dem Analytics Projekt aufgelistet. Selektieren Sie das Modul, von dem Sie die Daten darstellen möchten. Auf der rechten Seite befindet sich eine Vorschau zu dem selektierten Modul.



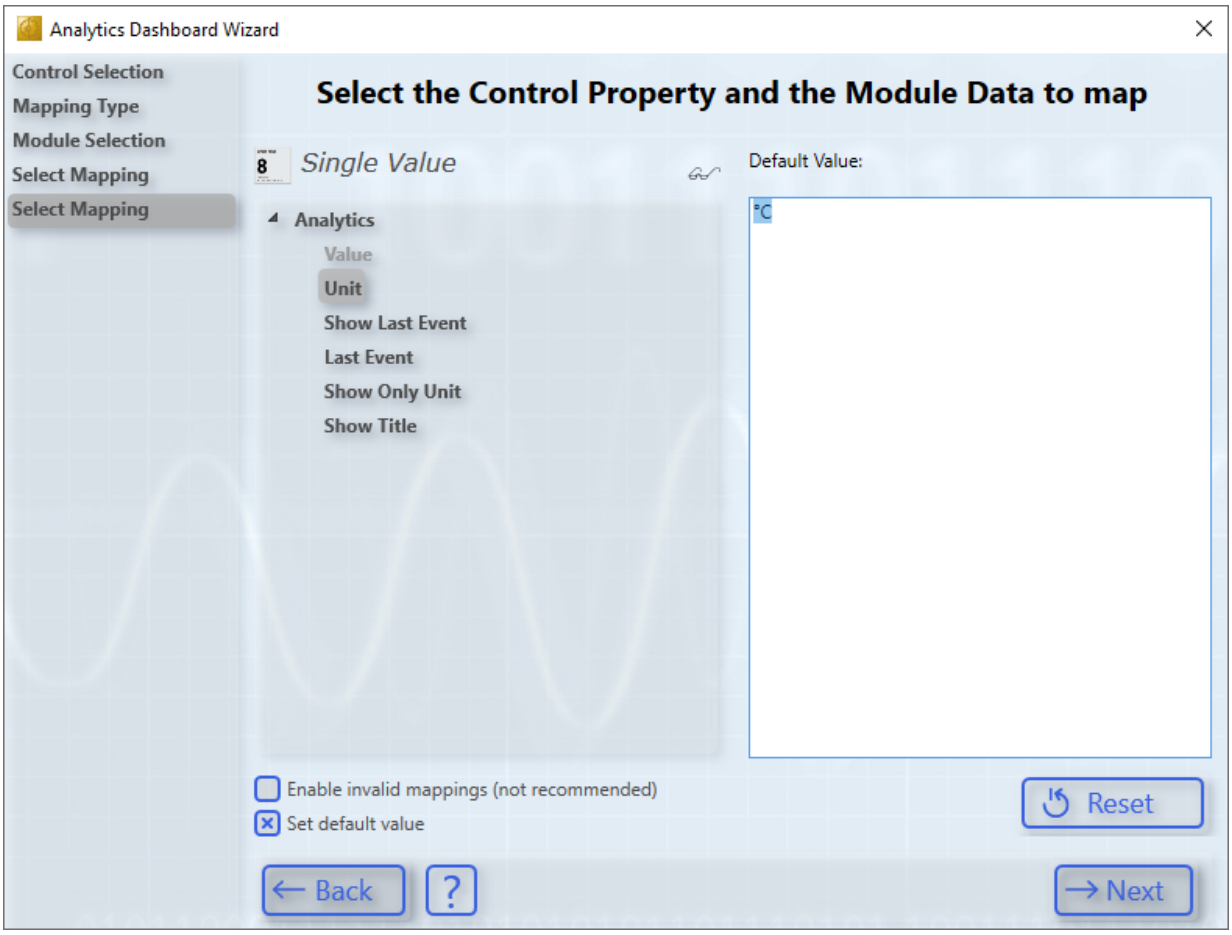
4. Hier erfolgt die Verknüpfung der ersten Control Eigenschaft mit einer Modul Variable. Wählen Sie dazu den anzuzeigenden Wert **Value** mit der Variable **Max**.



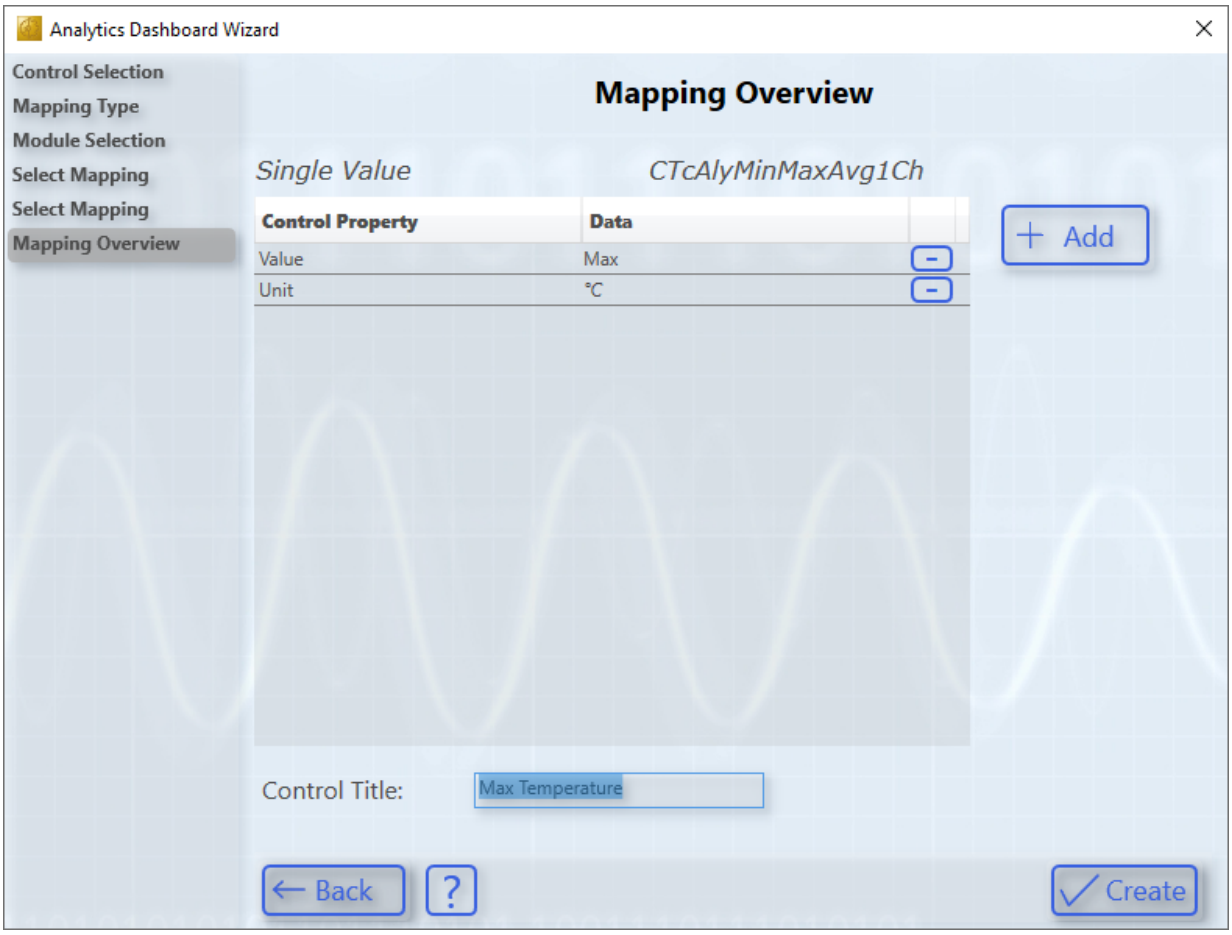
5. Auf dieser Übersichtsseite werden alle bestehenden Verknüpfungen aufgelistet. Fügen Sie mit **Add** eine weitere Verknüpfung hinzu.



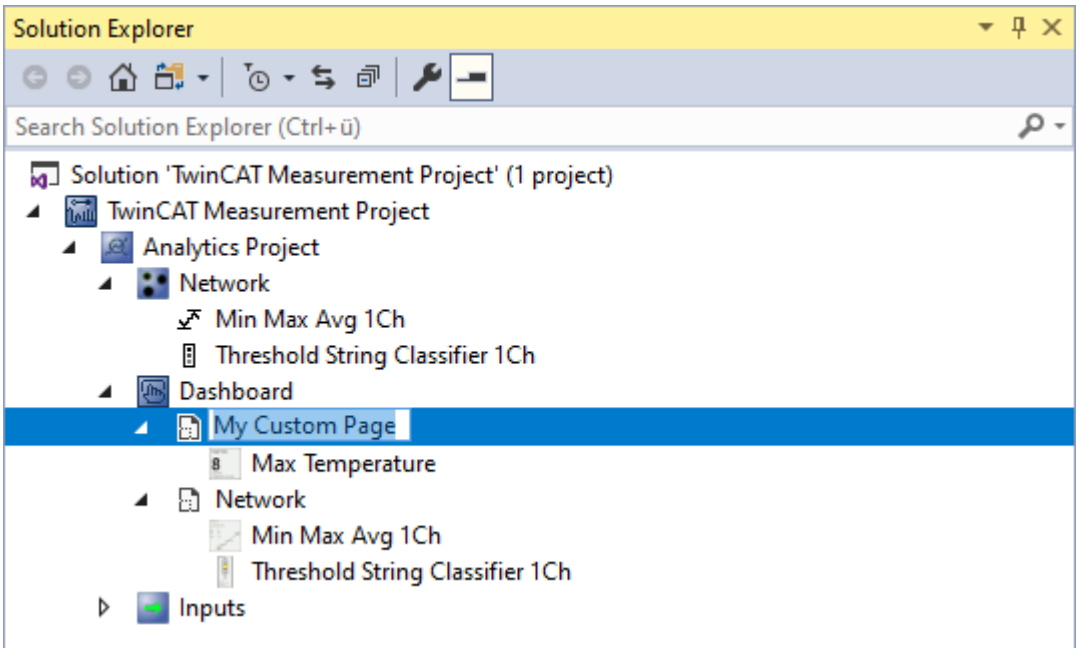
6. Wählen Sie **Unit** aus und setzen Sie einen Haken bei **Set default value**, um einen statischen Wert zu vergeben. Auf der rechten Seite öffnet sich ein Textfeld, tragen Sie dort **°C** ein.



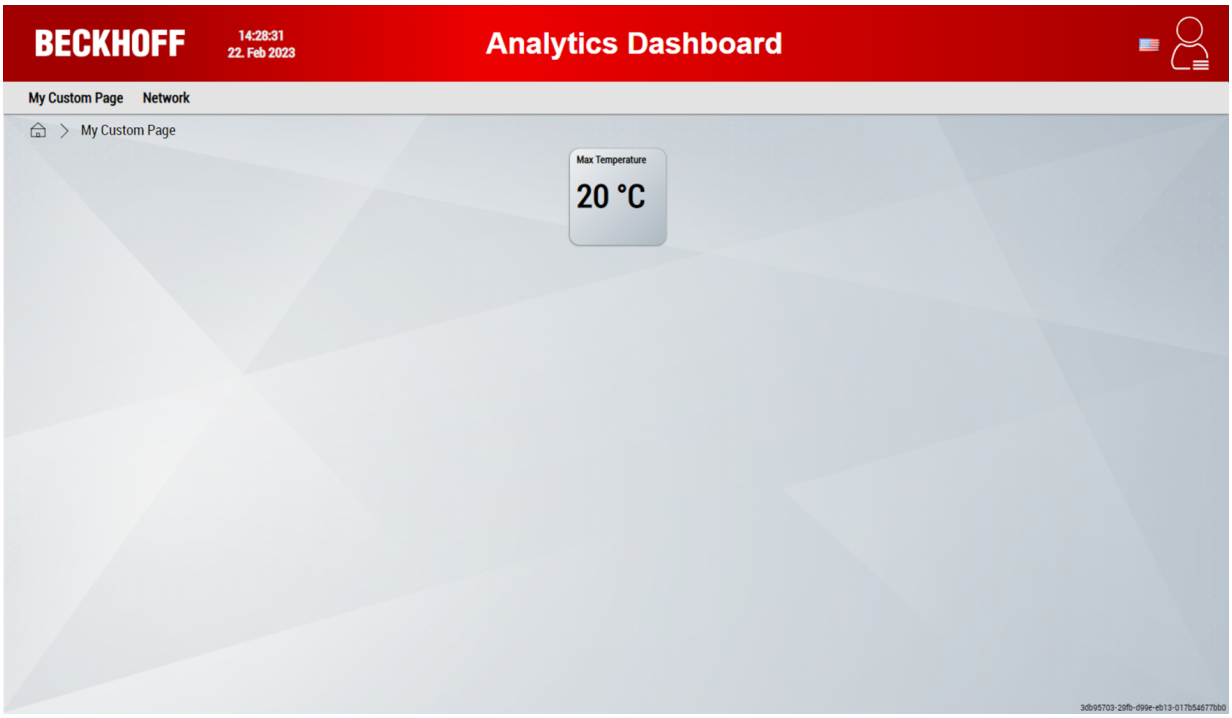
7. Ändern Sie Titel des Controls in **Max Temperature** und fügen Sie über **Create** das Control-Mapping hinzu.



8. Im Knoten **Dashboard** ist das erstelltes Control **Max Temperature** sichtbar. Mit einem **Rechtsklick** > **Rename** können Sie den Content umbenennen.



⇨ Nach einer erfolgreichen HMI und SPS Generierung können Sie zuvor erstelle Seite **My Custom Page** über die Navigation öffnen. Dort sehen Sie das manuell angelegte Control.



6.8.2 Angepasste und eigene Controls verwenden

HMI Control Mapping Wizzard

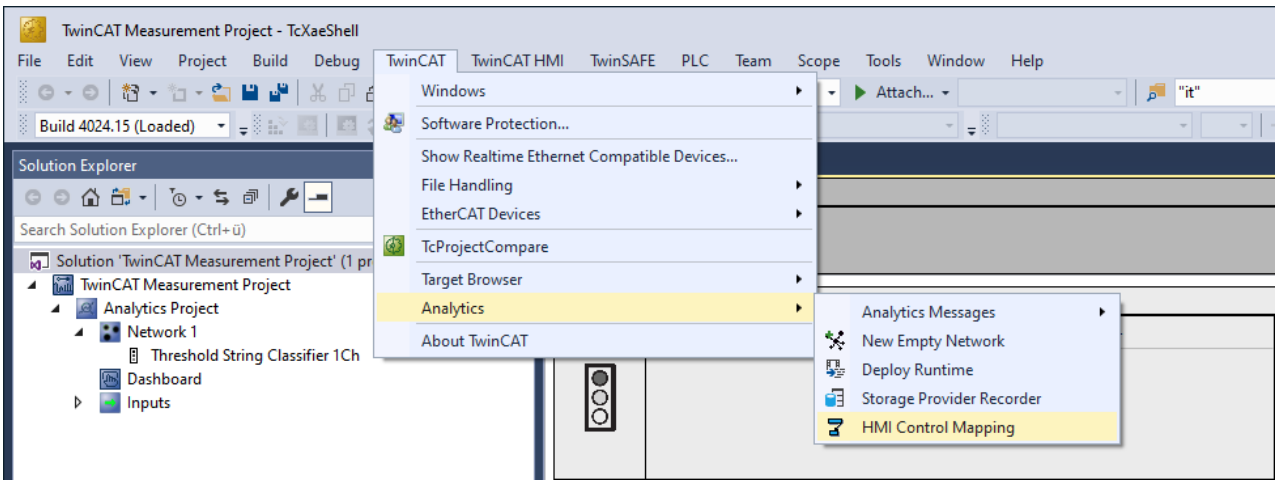
Der HMI Control Mapping Wizard ermöglicht Folgendes:

1. Das Anfügen eigener Controls.

2. Das Mappen von Controls zu Modul-Klassen oder das Ändern vorhandener Mappings.

3. Das Mappen von Controls zu Modul-Instanzen oder Virtuellen Inputs (Analytics Projekt muss geöffnet sein)

Öffnen Sie den Wizard über den Reiter **TwinCAT > Analytics > HMI Control Mapping**.

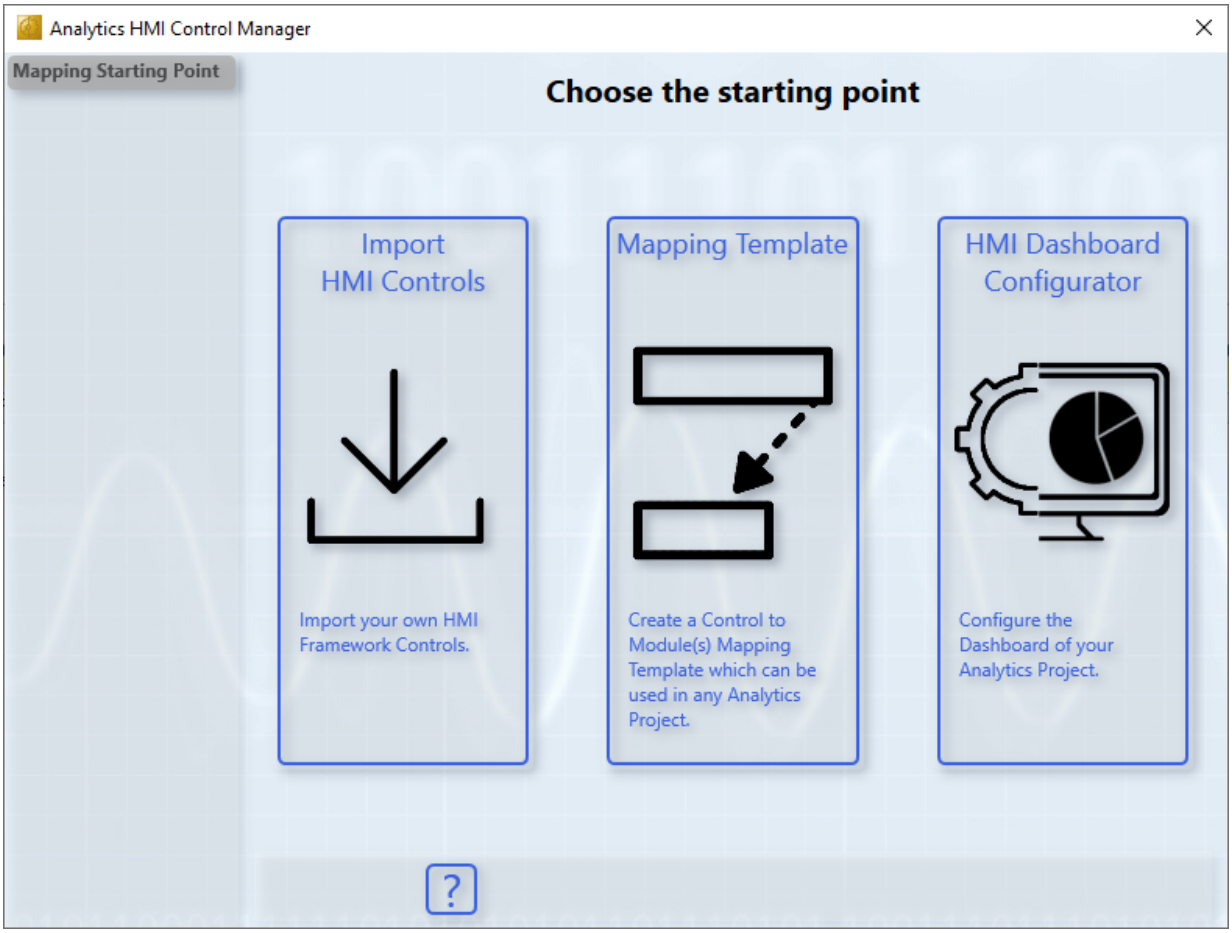


Über das Fragezeichen **?** im Wizard wird Ihnen zusätzliche Hilfe angeboten.

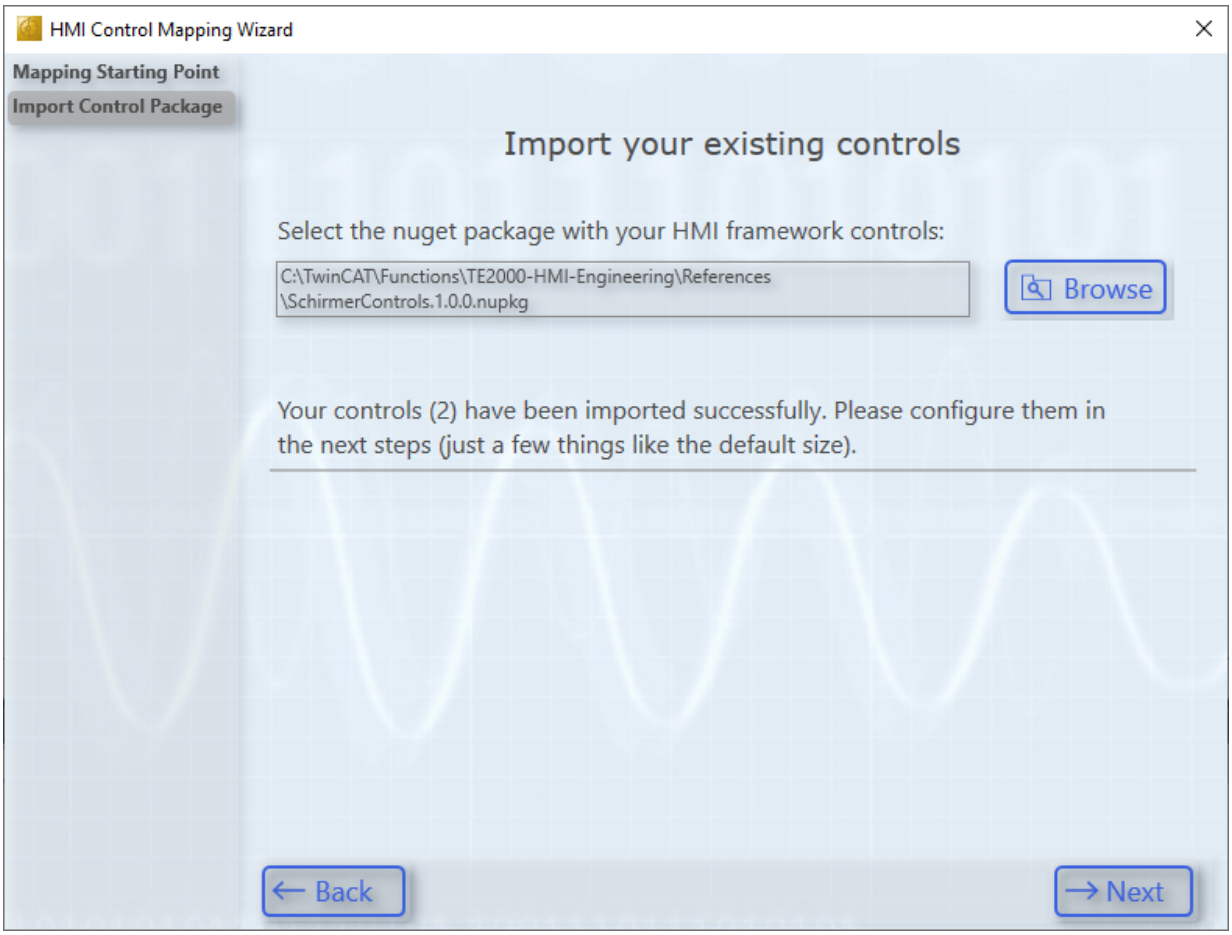
1. Anfügen eigener Controls

Die TwinCAT 3 HMI ermöglicht es, eigene HMI Framework Controls zu erstellen und diese als NuGet-Paket zu exportieren.

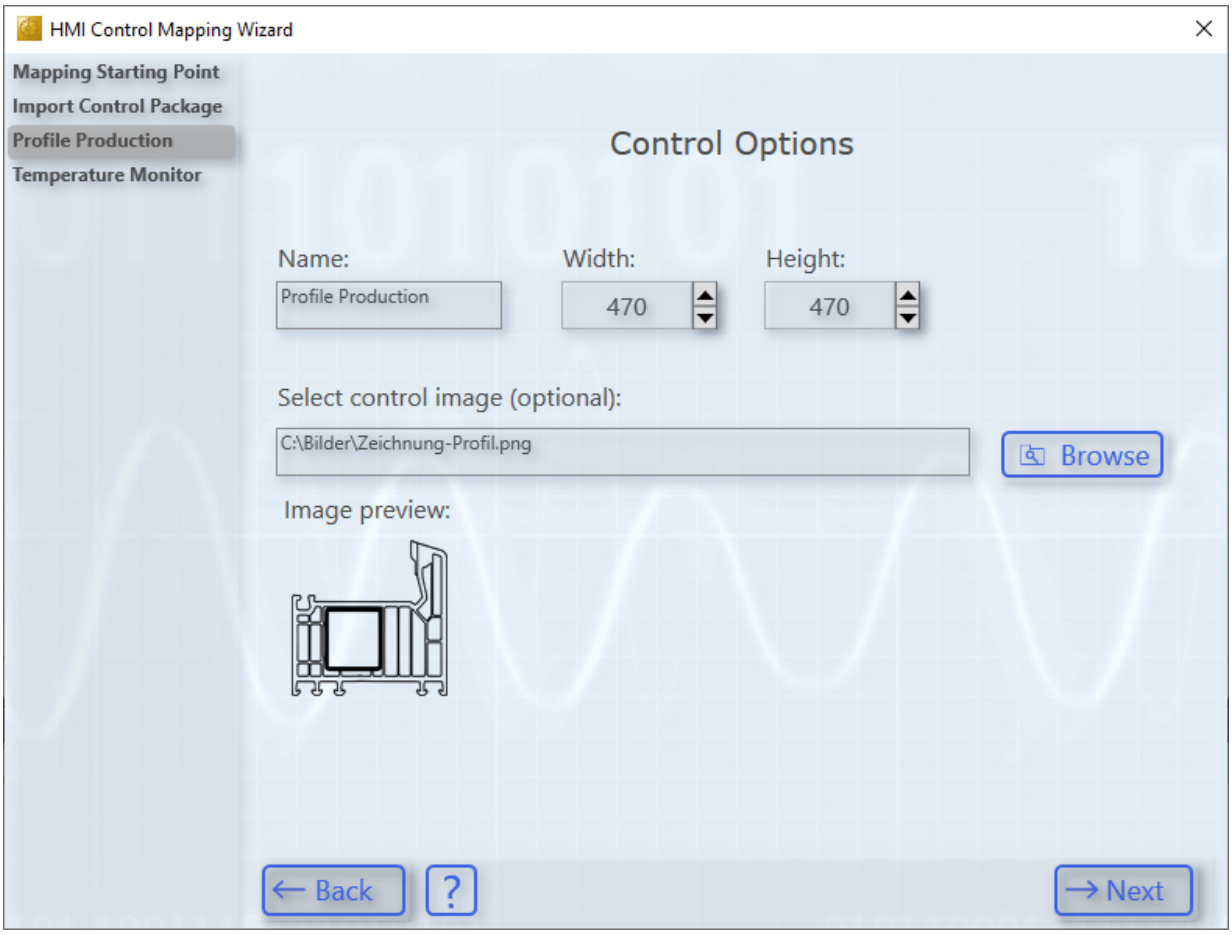
1. Um eigene Framework Controls den Analytics-Modulen zuzuordnen, klicken Sie auf **Import HMI Controls**



2. Wählen Sie das NuGet-Paket über **Browse** aus.



3. Wählen Sie für Ihre Controls als Nächstes einen Namen, eine Größe und ein Bild aus. Klicken Sie dann auf **Next** und bei dem letzten Control auf **Create**.

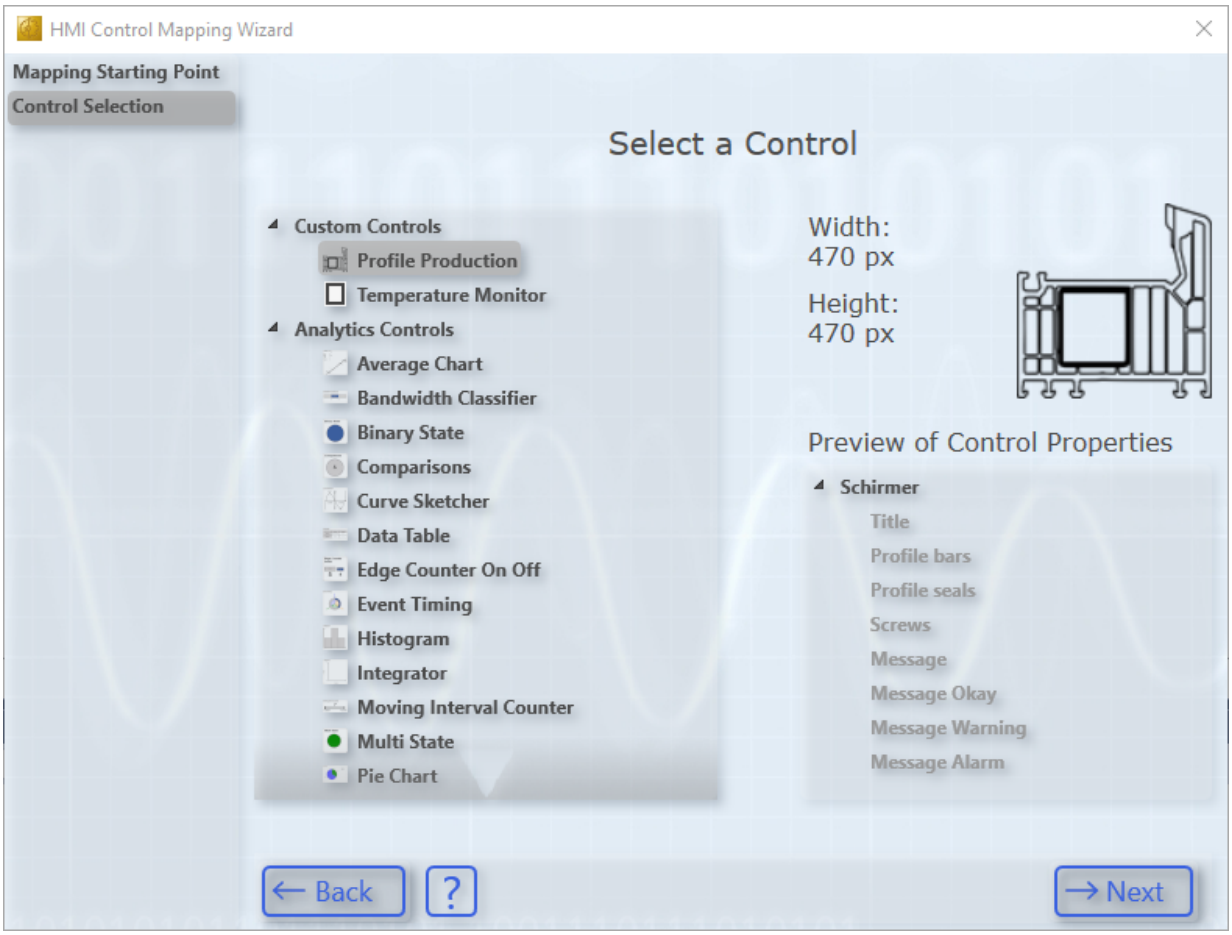


⇨ Sie werden automatisch weitergeleitet, sodass Sie ein Mapping zwischen Ihren Controls und Analytics Modulen erstellen können.

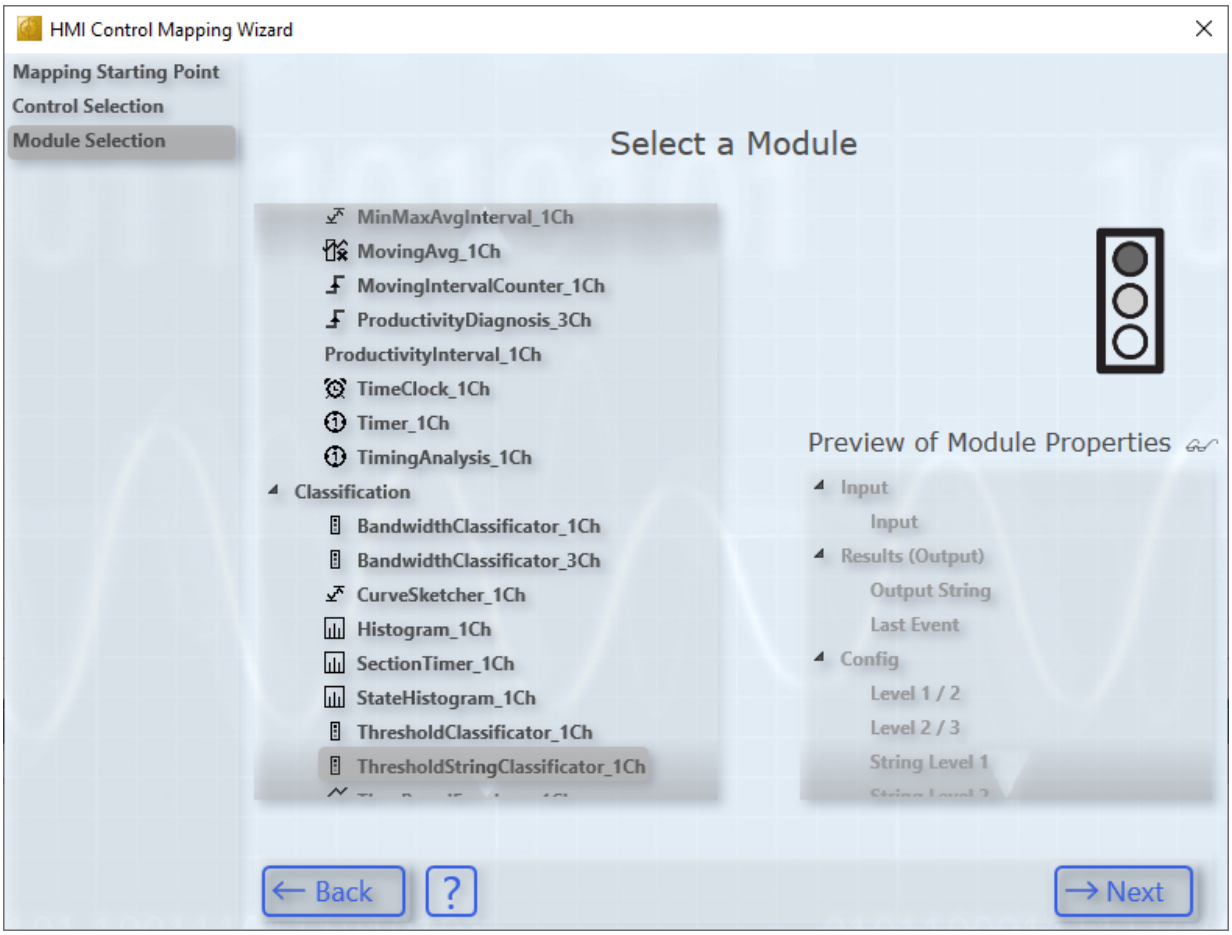
2. a) Mappen von Controls zu Modulen (weiterführend zu 1.)

Unter 2. b) wird dieser Schritt nochmal anhand des „Binary State“ Control detaillierter erläutert.

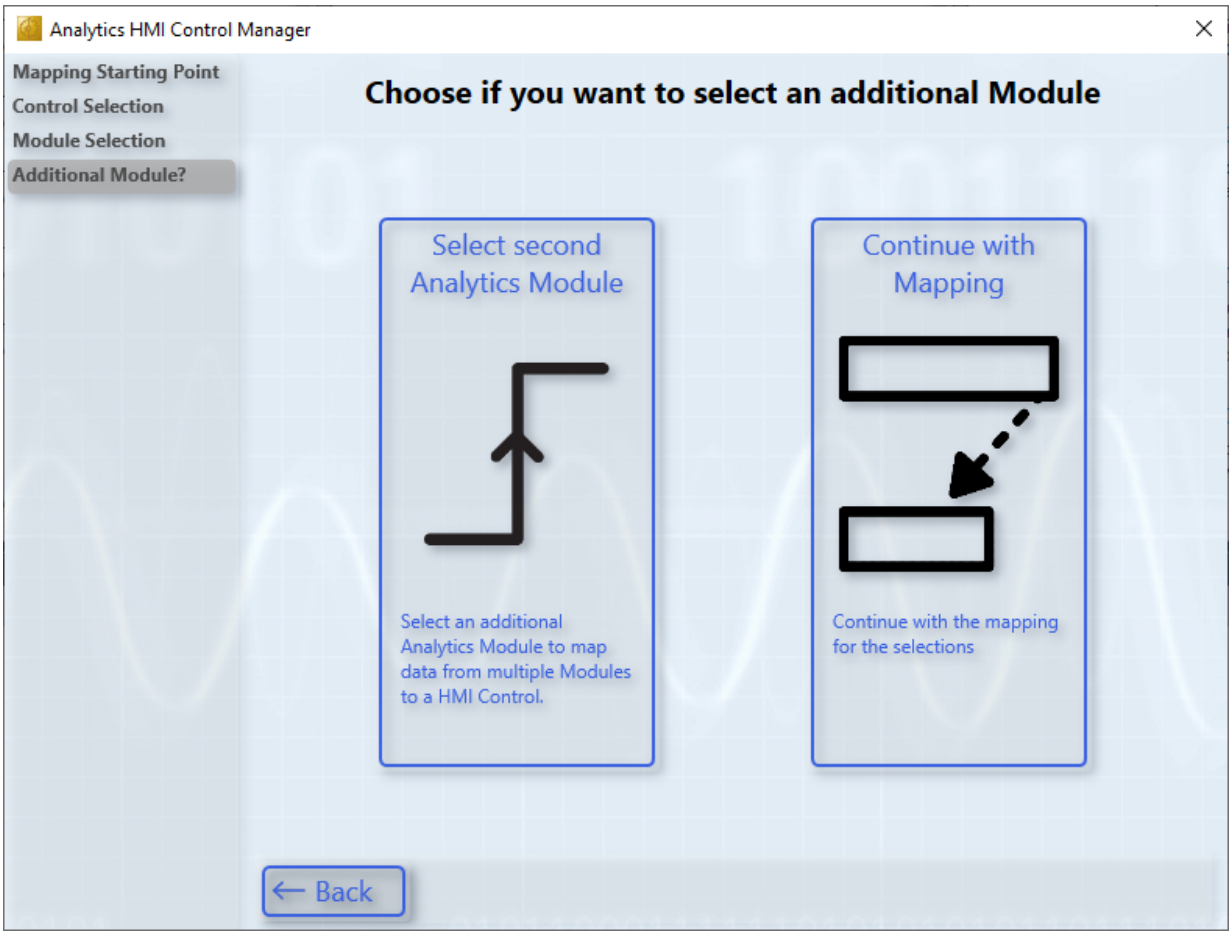
1. Wählen Sie nun Ihr Control aus.



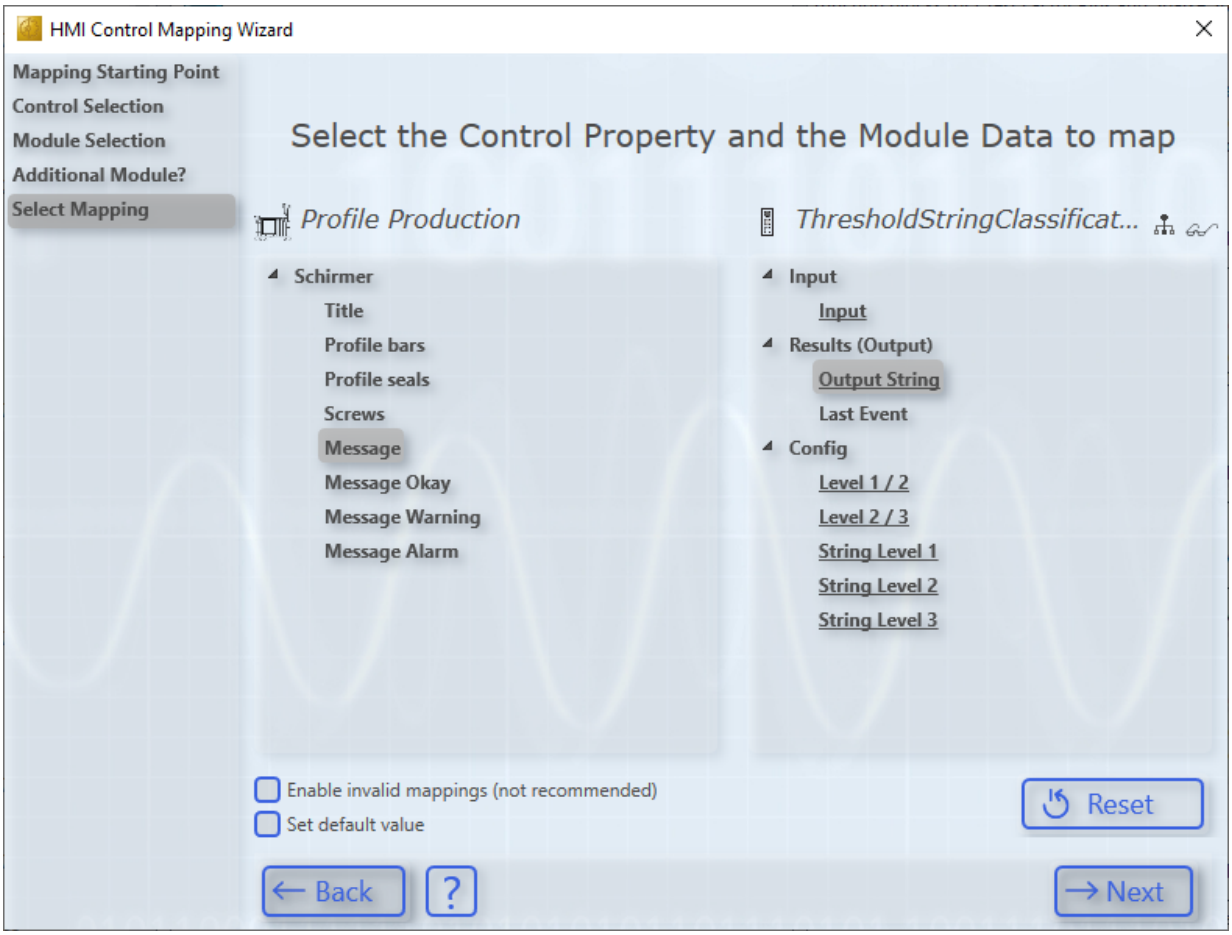
2. Wählen Sie nun das Modul aus, dem Sie das Control zuordnen wollen.



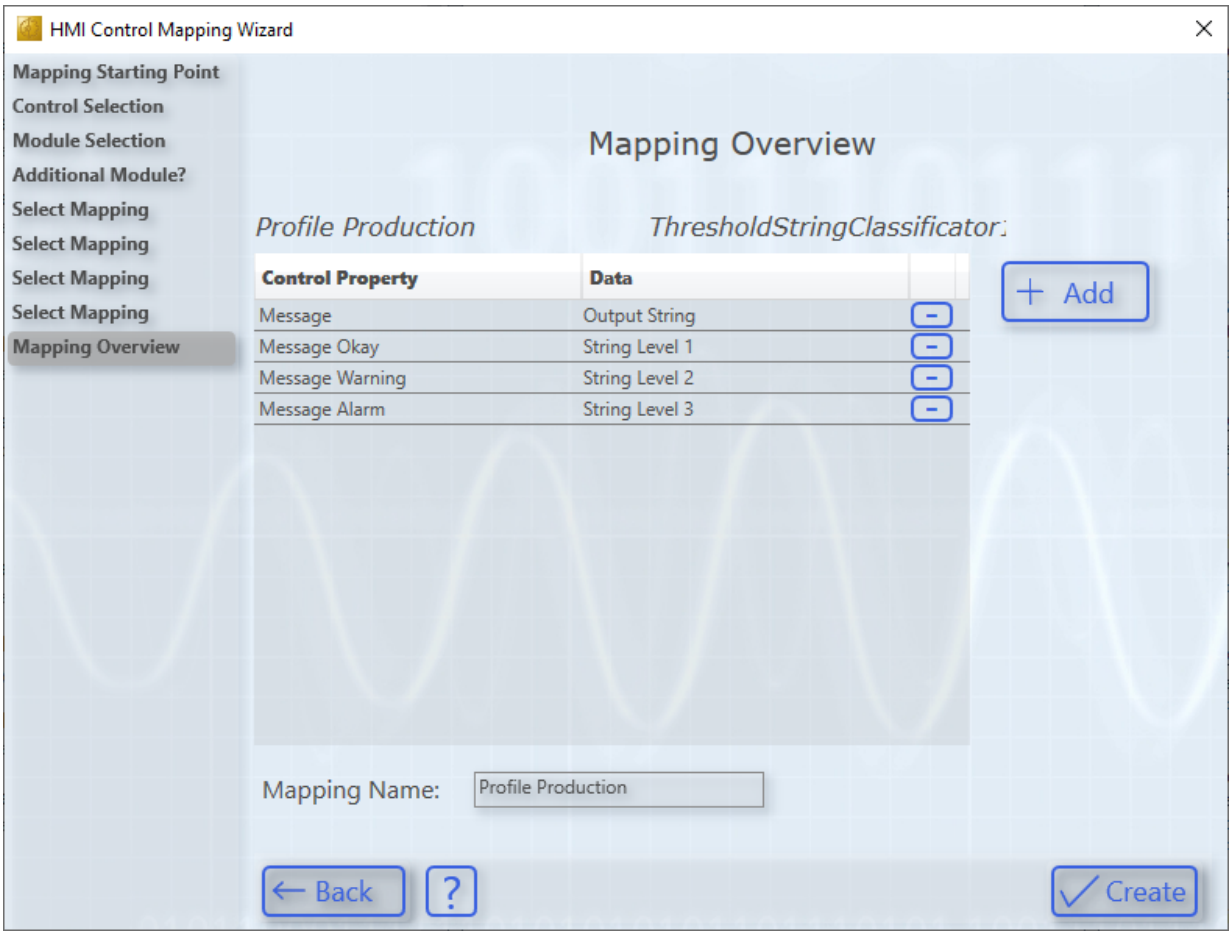
3. Wählen Sie **Continue with Mapping**.



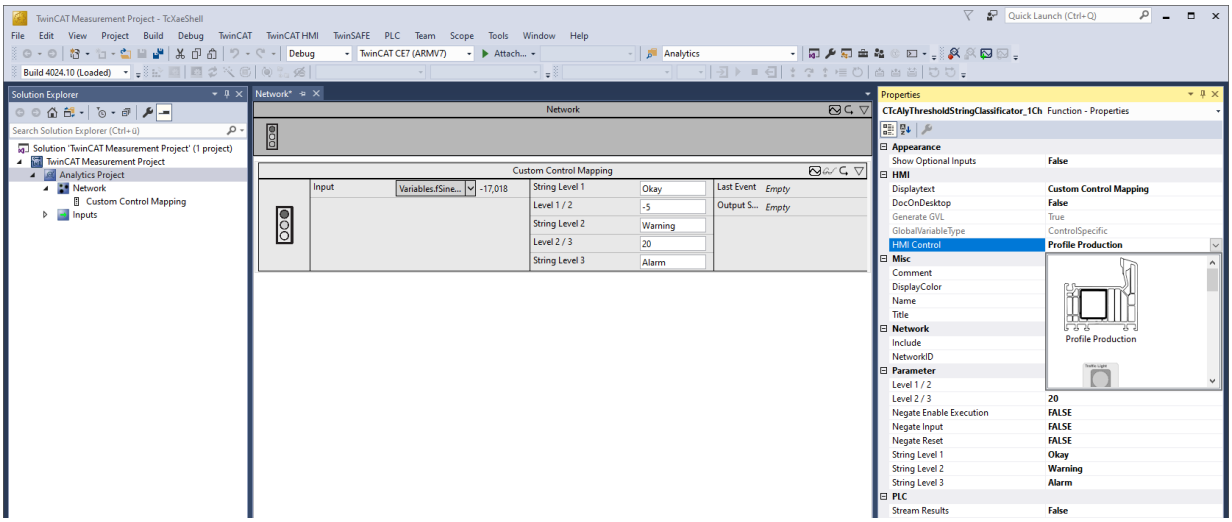
4. In den nächsten Schritten verbinden Sie die Control-Eingänge mit den Moduldaten.

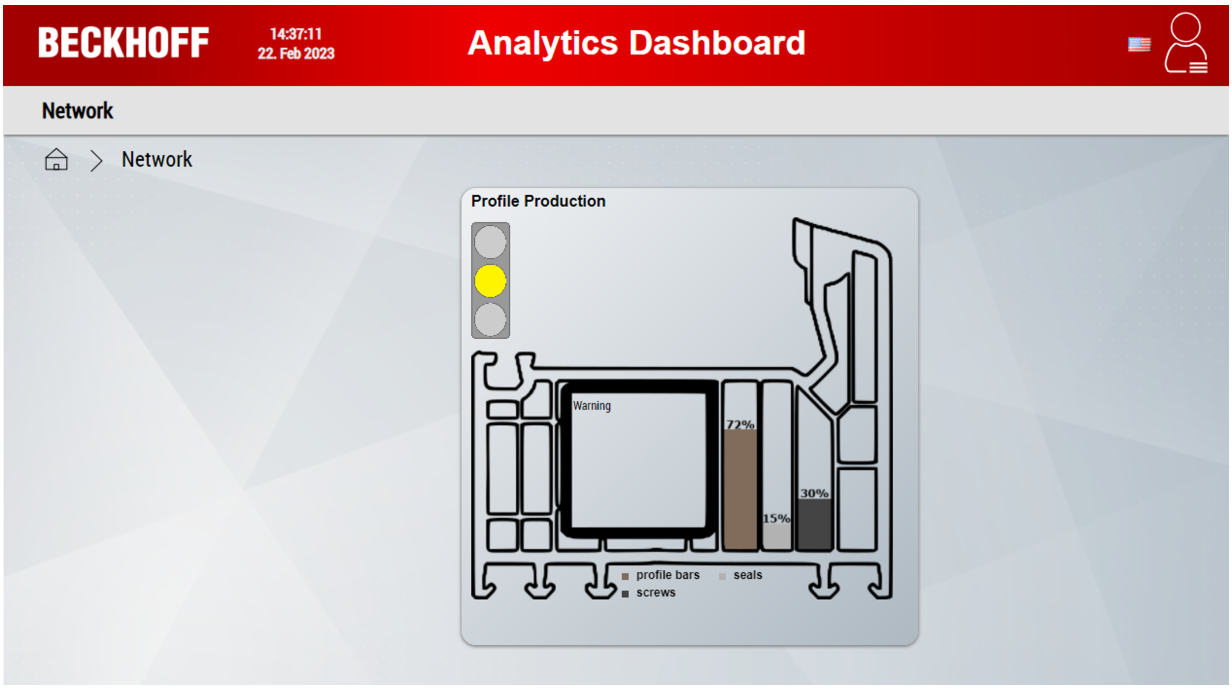


5. Als Letztes klicken Sie auf **Create**, um das Mapping der Analytics Workbench hinzuzufügen.



6. Sie können nun den Wizard schließen und das Mapping wird automatisch für das Modul verfügbar sein. Wählen Sie es aus und generieren Sie sich Ihr Dashboard.



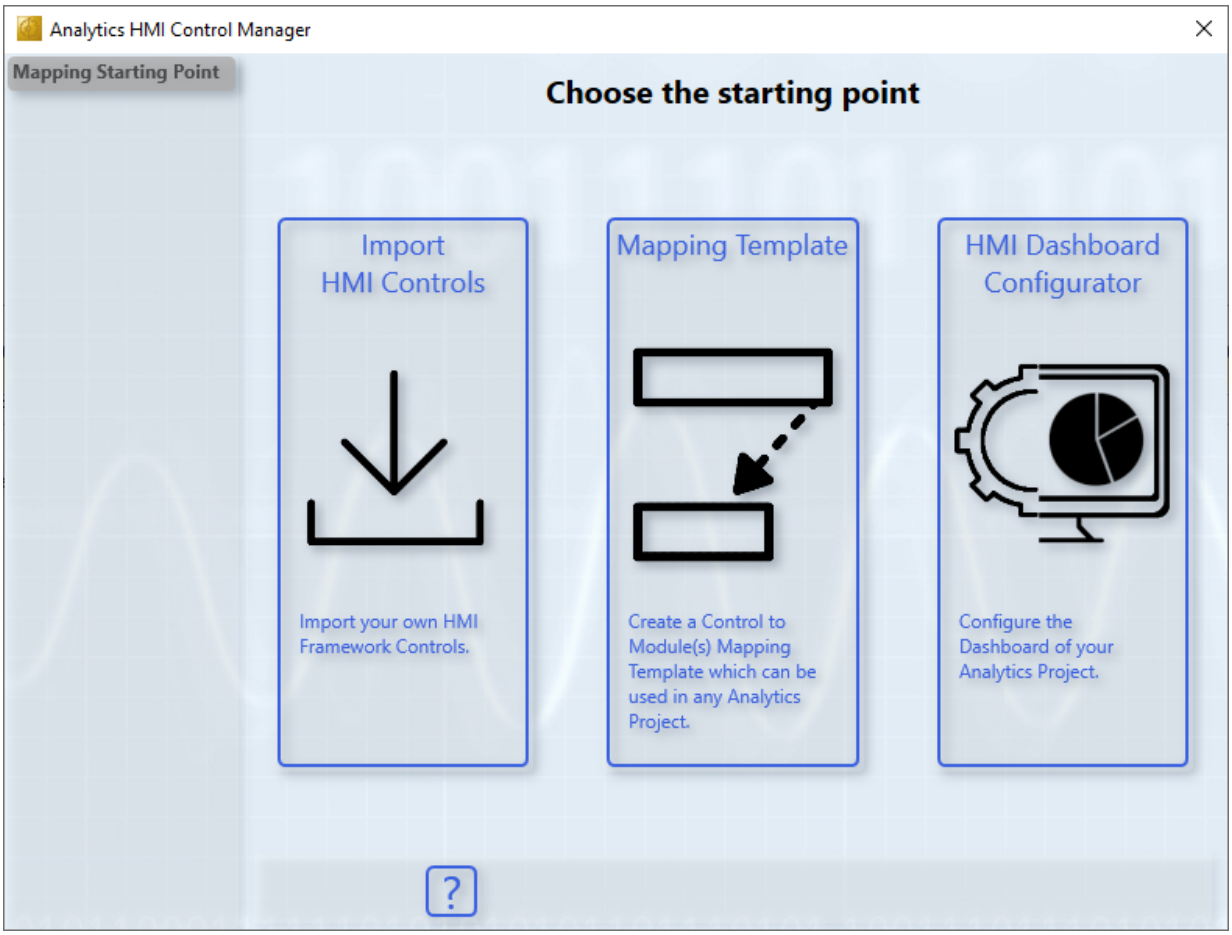


⇨ Das Dashboard ist fertig.

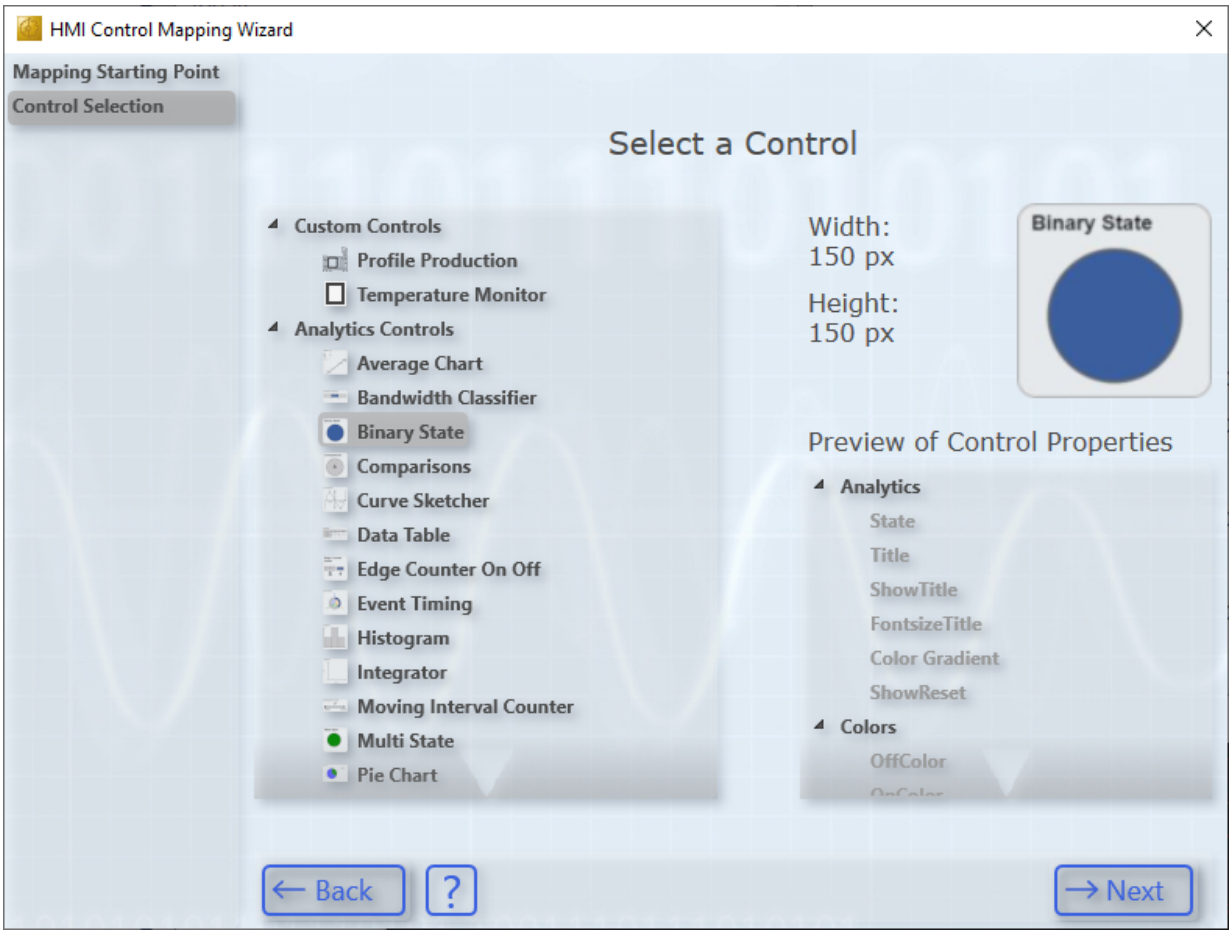
2.b) Mappen von Controls zu Modulen (ohne eigene Controls)

Analog zu 2.a) wird in diesem Abschnitt beschrieben, wie Sie ein Mapping zwischen einem Control und einem Modul durchführen. Als Beispiel wird ein bestehendes Analytics Control verwendet. Sie können dieses Beispiel also direkt ausprobieren.

1. Wählen Sie dafür den Punkt **Mapping Template**.

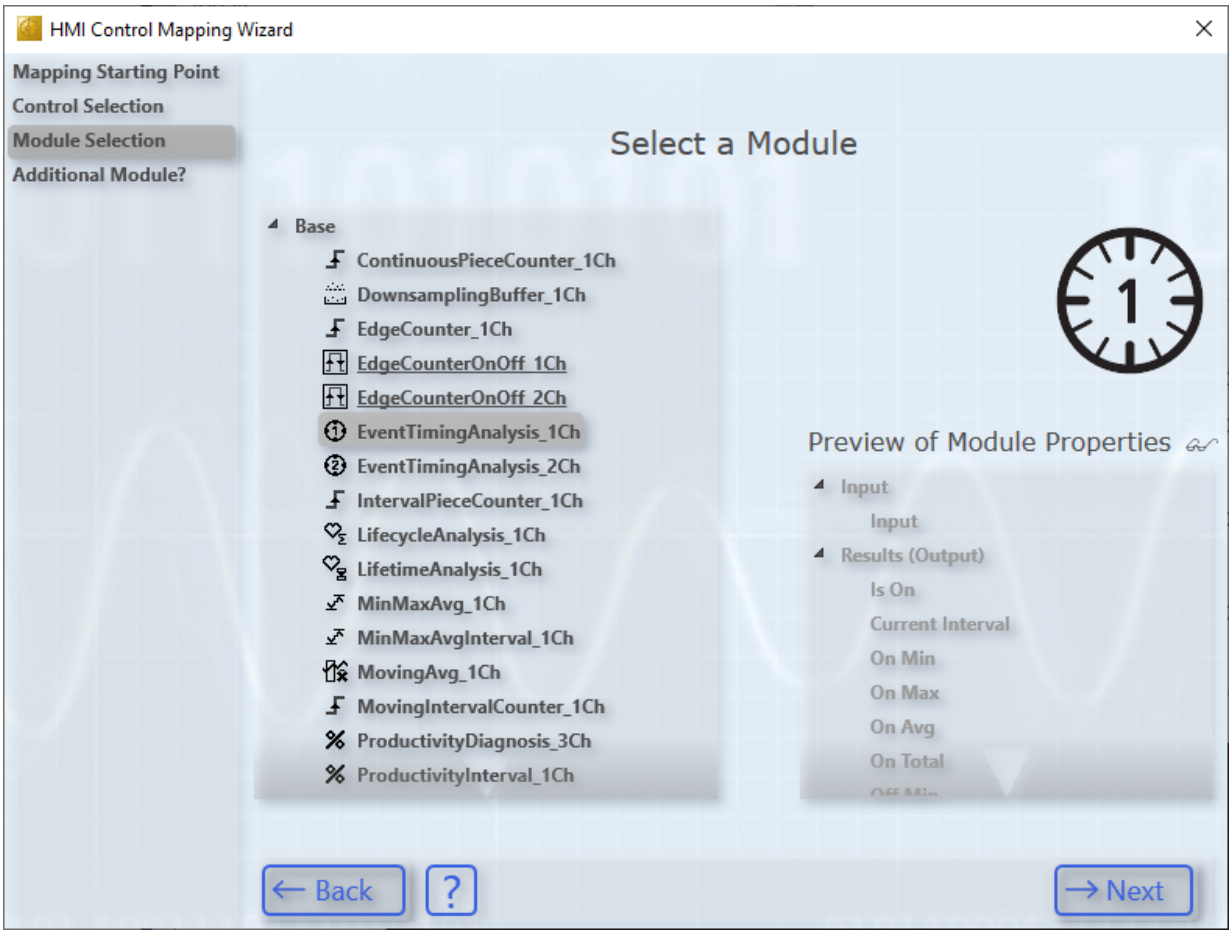


2. Wählen Sie ein Control aus. Auf der rechten Seite werden die Eigenschaften des Controls angezeigt. Ebenso wird angezeigt, wie groß das Control auf dem Dashboard sein wird.

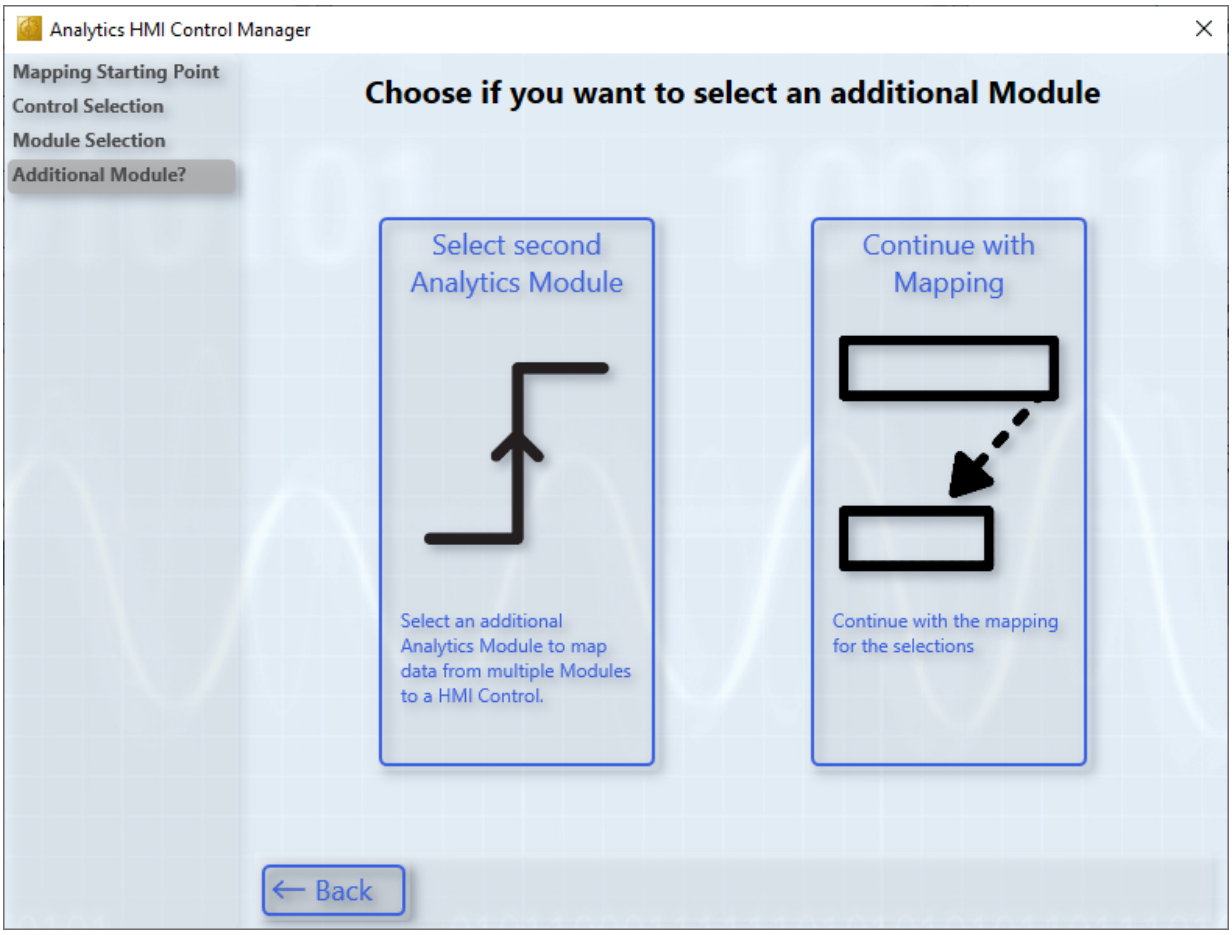


3. Wählen Sie das Modul aus, von dem Sie die Daten anzeigen lassen möchten. Es werden die Eingänge, Ausgänge und Parameter als Vorschau angezeigt, die anschließend mit den Eigenschaften des ausgewählten Controls verknüpft werden können.

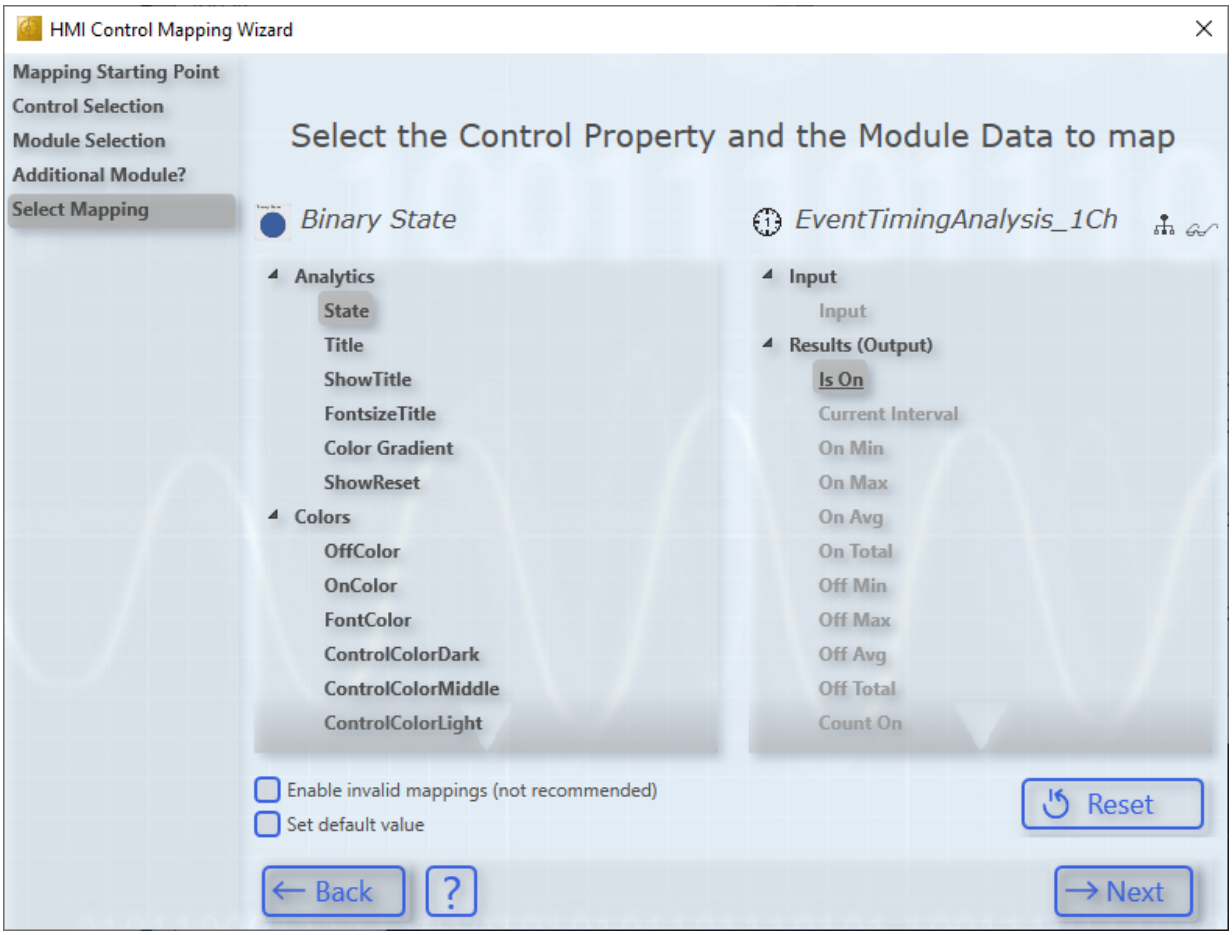
Bereits gemappte Module werden unterstrichen dargestellt. Diese Mappings können auch editiert werden.



4. Zukünftig kann ein Control auch von verschiedenen Modulen Daten darstellen. Da dies derzeit nicht möglich ist, wählen Sie **Continue with Mapping**.

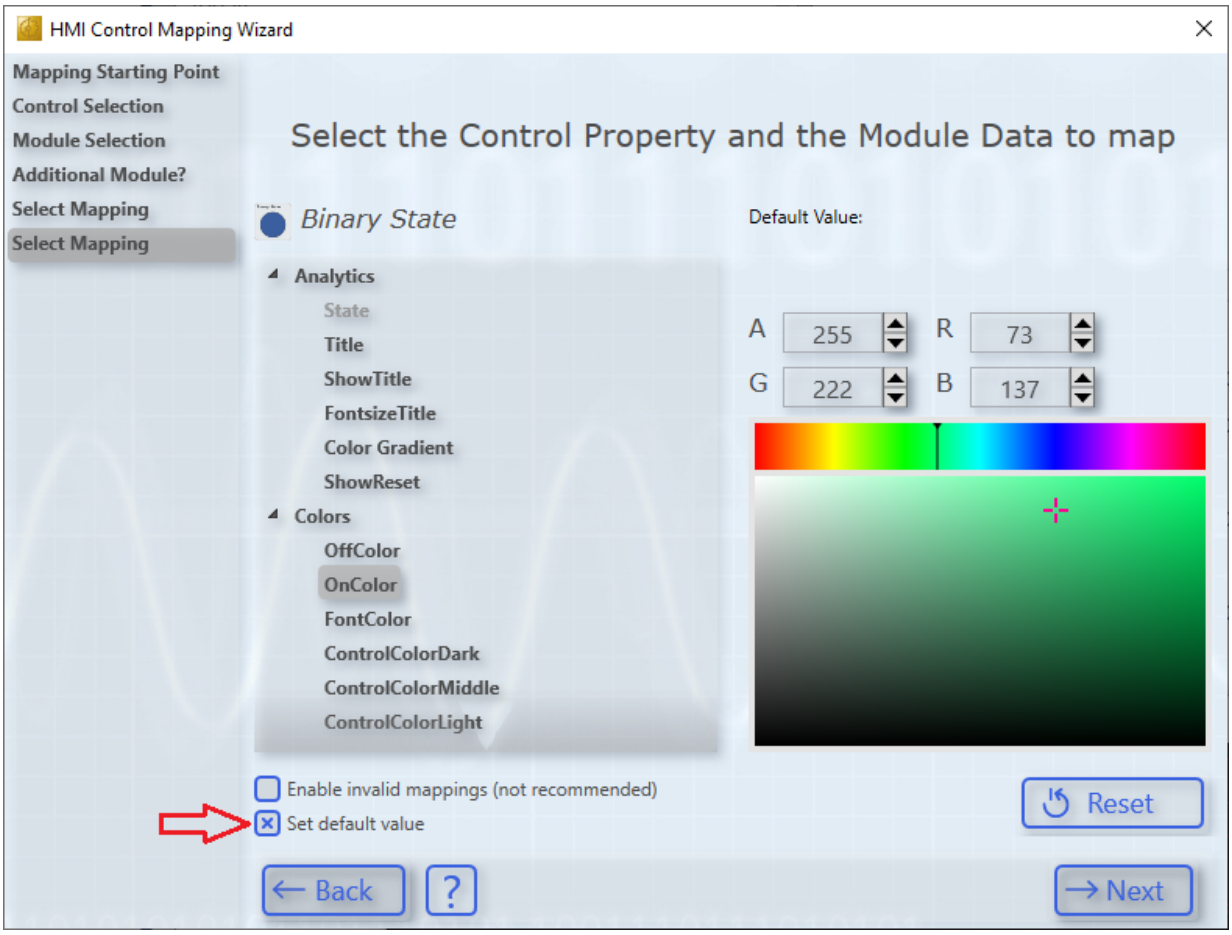


5. Wählen Sie als Nächstes eine Control Eigenschaft aus, die gemappt werden soll. Alle unmöglichen Variablen, die aufgrund ihres Datentyps nicht zugeordnet werden können, sind ausgegraut. Alle anderen können gemappt werden.
   Wählen Sie nun die Modul Variable aus, die Sie mit der Control Eigenschaft mappen wollen.

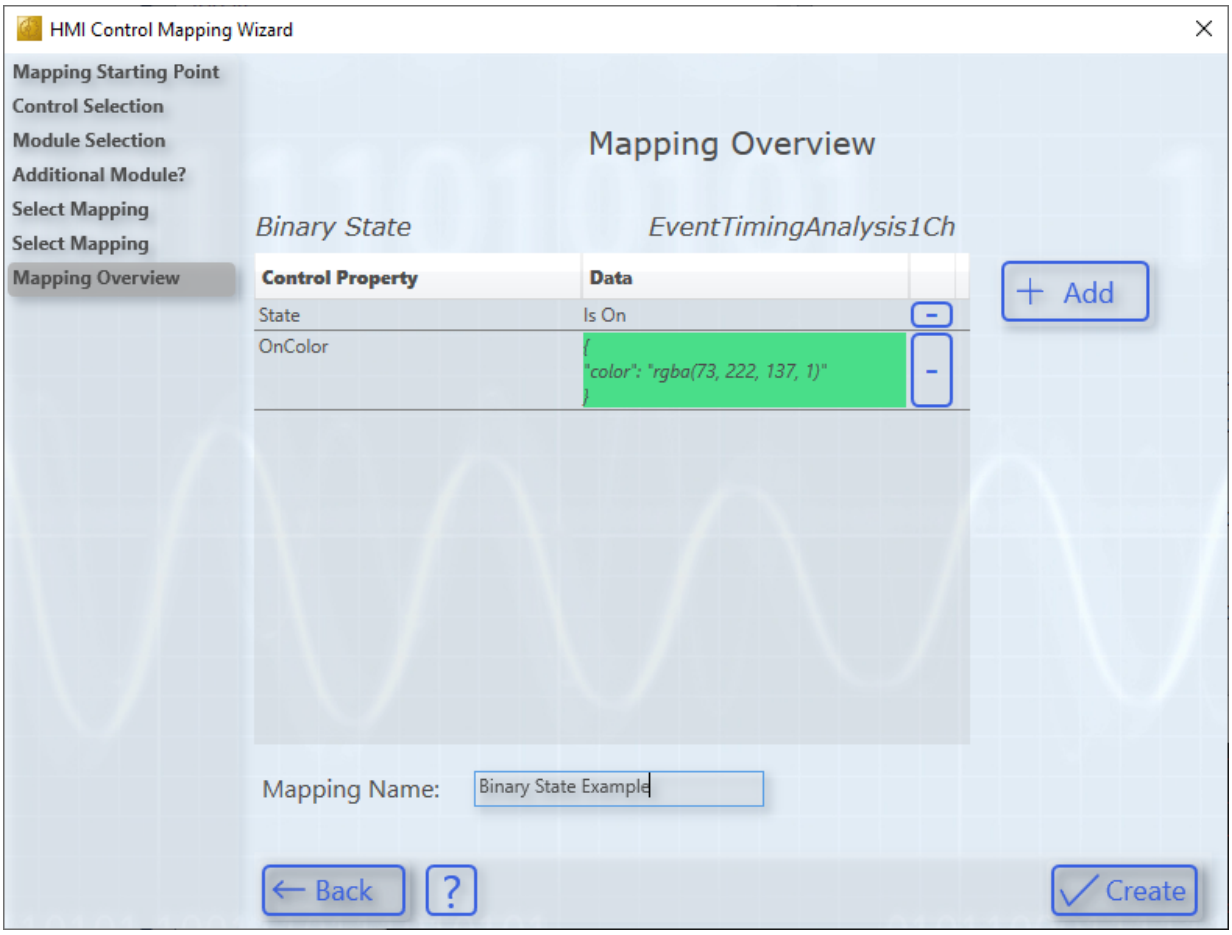


6. Ebenso können Sie die Checkbox **Set default value** aktivieren, um einen Standardwert dem Control zuzuweisen. Damit können beispielsweise Standardfarben geändert oder auch boolesche Werte wie „ShowTitle“ auf „False“ gesetzt werden, wenn in dem Analytics Control kein Titel angezeigt werden soll.

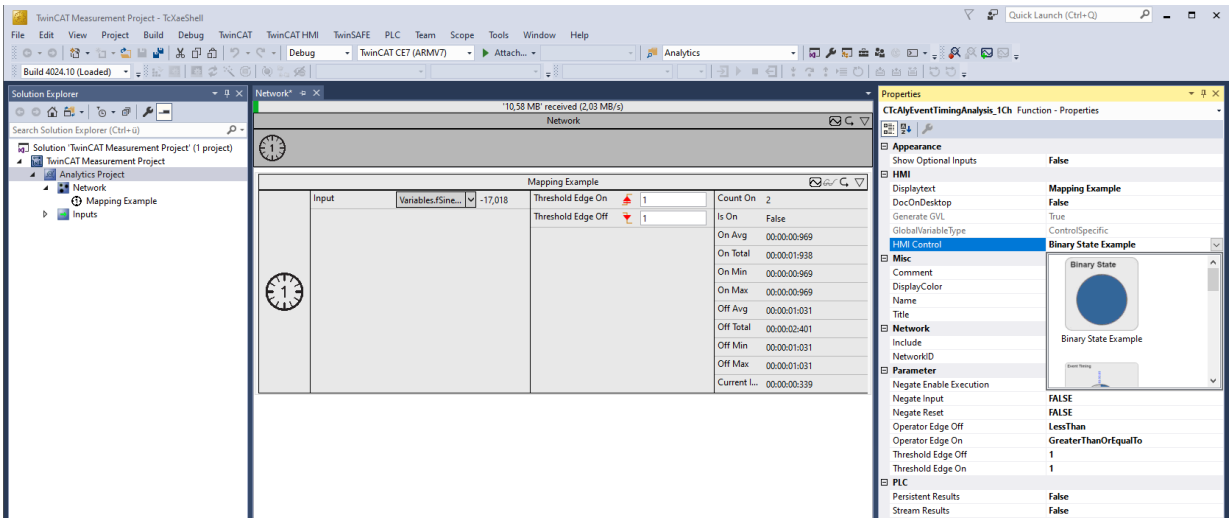
In diesem Beispiel wird die „OnColor“ geändert. Anhand des Datentyps wird eine spezifische Auswahlmöglichkeit zur Verfügung gestellt.

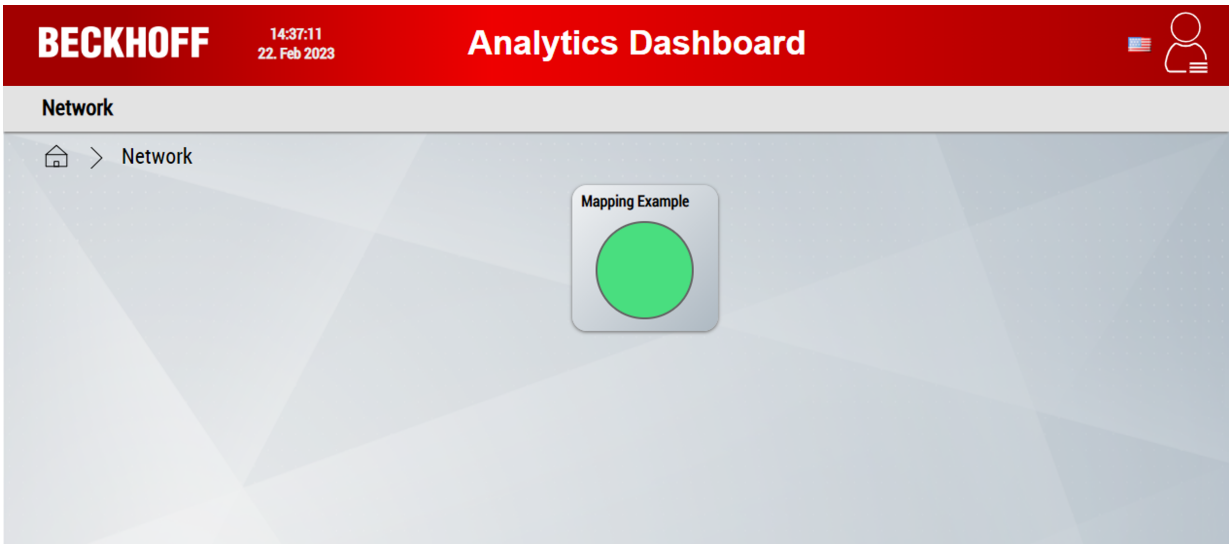


7. Soweit Sie keine weiteren Einträge mehr über **Add** hinzufügen wollen, können Sie über **Create** das Mapping vollenden. Es wird empfohlen einen aussagekräftigen Mapping- Namen zu vergeben.



8. Sie können den Wizard schließen und das Mapping wird automatisch für das Modul verfügbar sein. Wählen Sie es aus und generieren Sie sich Ihr Dashboard.



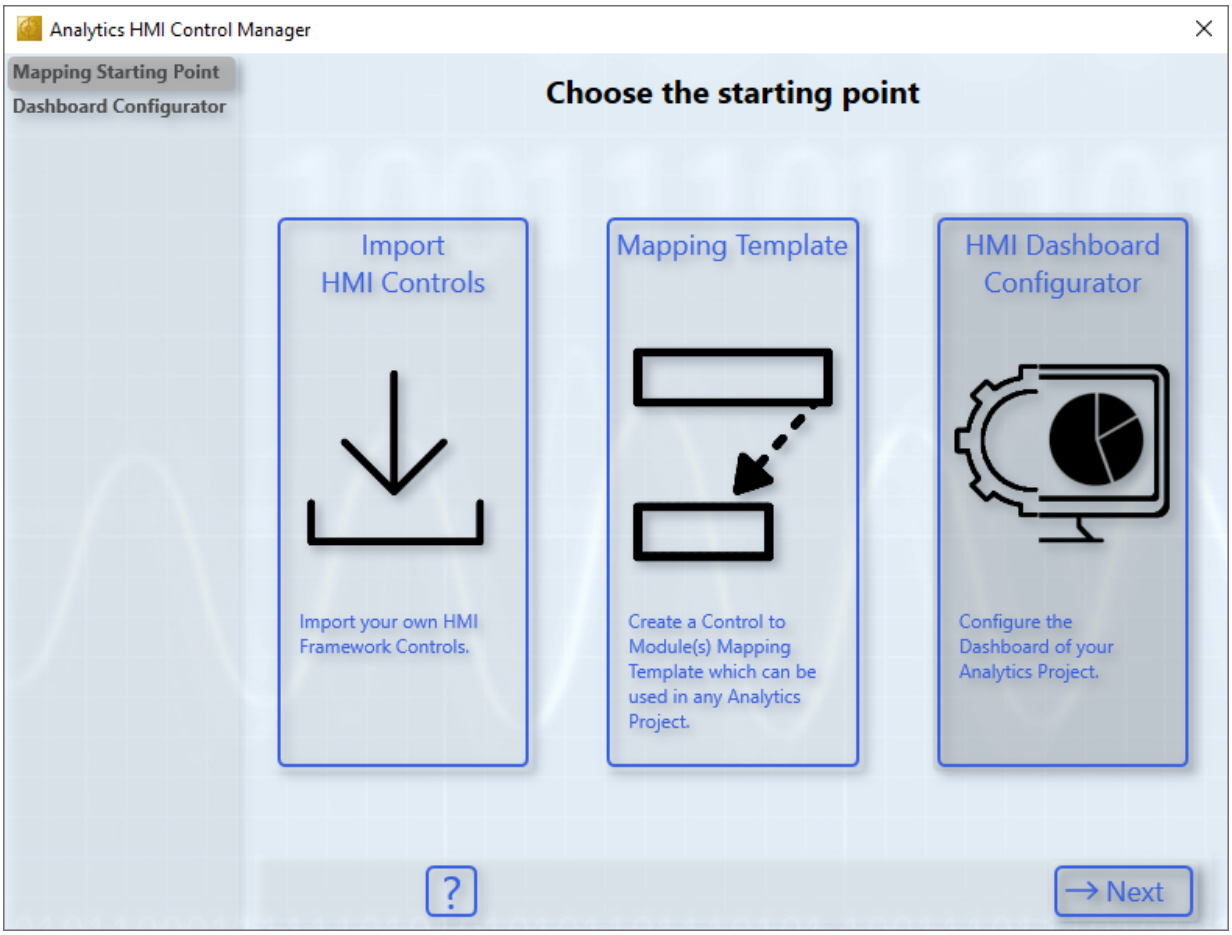


⇨ Das Dashboard ist fertig.

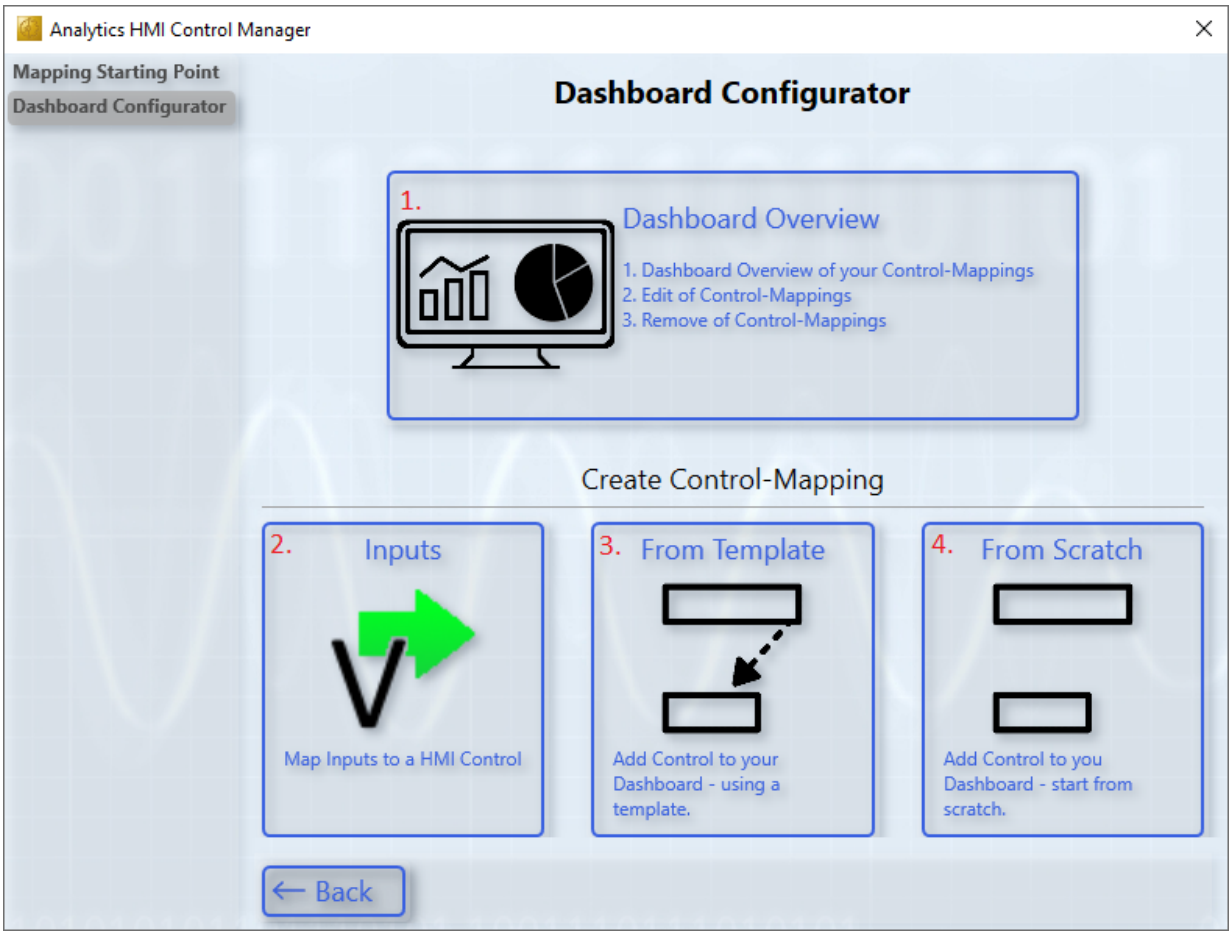
3. Mappen von Controls zu Modul-Instanzen

Neben dem Mappen von Controls zu Modul-Klassen können auch direkt die Modul-Instanzen eines Projekts mit Controls verknüpft werden. Dies ist sowohl über den Dashboard Aufbau und Inhalte im Analytics Projekt verwalten [▶ 436] als auch über den Wizard möglich.

1. Klicken Sie zu Beginn des Wizards auf **HMI Dashboard Configurator**.



2. Dort haben Sie mehrere Auswahlmöglichkeiten.



1. Übersicht und Bearbeitungsmöglichkeiten über alle Controls
   - Diese Funktion ist ebenfalls über das Doppelklicken eines Controls im Solution Explorer möglich.
2. Ein Control hinzufügen, dass Ihre Eingangsdaten anzeigt
   - Diese Funktion ist ebenfalls über den Dashboard Aufbau und Inhalte im Analytics Projekt verwalten [▶ 436] möglich.
3. Ein Control hinzufügen, dass Daten von Modulen anzeigt (Auswahl über ein Template)
   - Diese Funktion ist für ein einzelnes Modul auch über das *Properties* Fenster einstellbar. Zur Darstellung von Daten aus mehreren Modulen ist dies der einzige Weg.
   - Sie sehen direkt welche Daten aus dem Template verknüpft sind und können diese direkt anpassen.
4. Ein Control hinzufügen, dass Daten von Modulen anzeigt (komplett Manuell)
   - Diese Funktion ist ebenfalls über den Dashboard Aufbau und Inhalte im Analytics Projekt verwalten [▶ 436] möglich.

Für die einzelnen Punkte müssen Sie den Schritten des Wizards folgen. Die Schritte sind gleich bzw. sehr ähnlich zu Abschnitt „2. Mappen von Controls zu Modulen“ aufgebaut.

6.8.3 Benutzerverwaltung und Zugriffsrechte konfigurieren

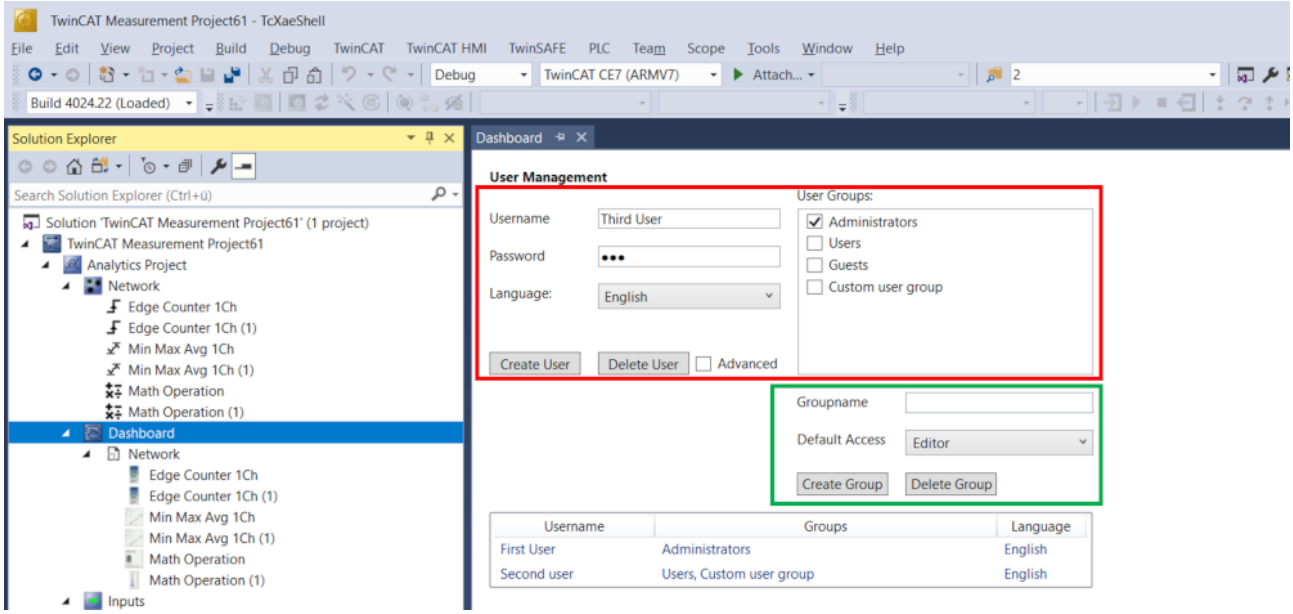
Über das Benutzermanagement lassen sich individuell Benutzer und Benutzergruppen generieren. Für Benutzergruppen können Zugriffsrechte auf Contents und Controls definiert werden.

Beachten Sie, dass das Benutzermanagement und die Zugriffsrechte erst ab der Dashboard-Version 2.0 in das HMI-Projekt übernommen werden.

6.8.3.1 Definition im Analytics Projekt

Definiterte Benutzer, Benutzergruppen und Zugriffsrechte werden bei der Code-Generierung automatisch in das HMI-Projekt übernommen. Im Webbrowser lassen sich nach der Generierung Änderungen an den Benutzern und Zugriffsrechten vornehmen.

6.8.3.1.1 Benutzerverwaltung



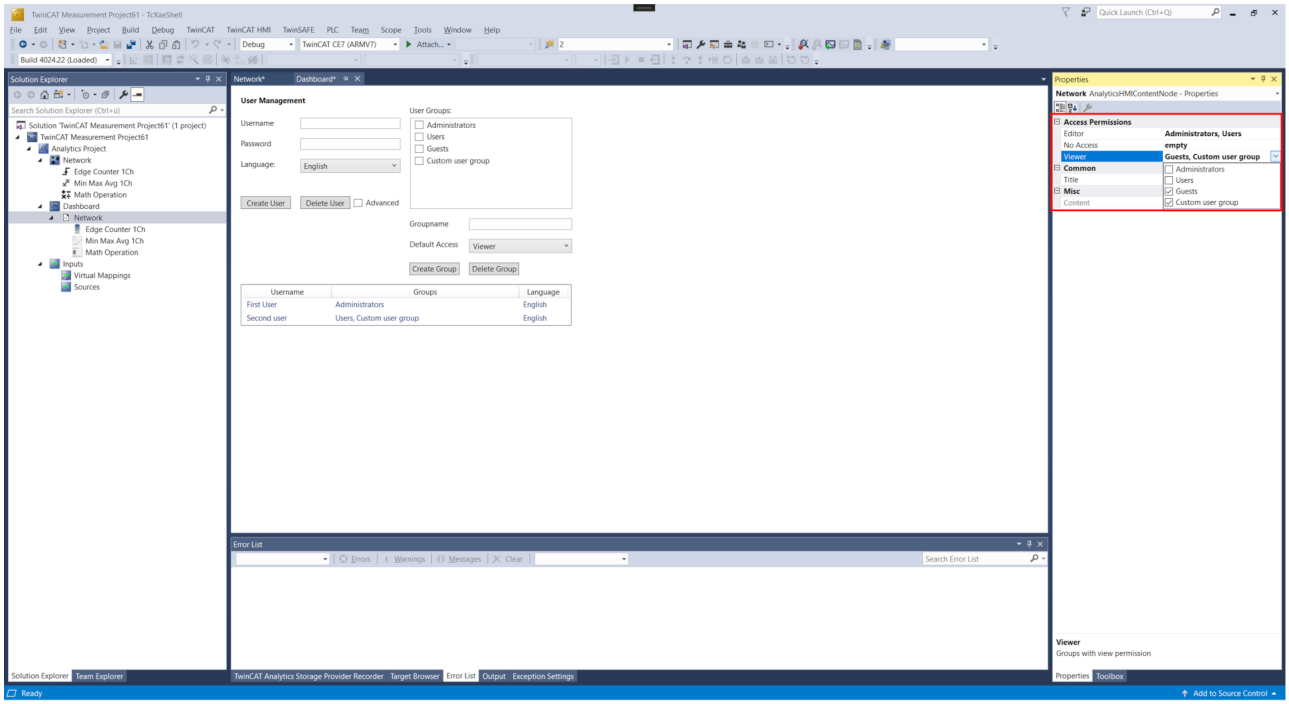
Unter dem Dashboard-Knoten des Analytics-Projekts können Benutzer (rot umrandet) angelegt werden. Für einen neuen Benutzer müssen ein Benutzername und Passwort sowie eine Sprache festgelegt werden. Zusätzlich gilt es dem Benutzer eine oder mehrere Benutzergruppen zuzuweisen, wodurch der Benutzer die Zugriffsrechte der jeweiligen Gruppe(n) erhält.

Standardmäßig gibt es folgende Benutzergruppen:

- Administratoren
- Users
- Guests

Für eine genauere Einteilung lassen sich weitere Benutzergruppen erstellen (grün umrandet). Hierfür muss ein Name und die Zugriffsberechtigung eingegeben werden.

6.8.3.1.2 Zugriffsrechte konfigurieren



Für Objekte unterhalb des Dashboard-Knotens (Contents und Controls) können Zugriffsrechte konfiguriert werden. Über die Access-Permissions (rot umrandet) kann für jede Benutzergruppe das Zugriffsrecht festgelegt werden.

Folgende Zugriffsrechte stehen zur Verfügung:

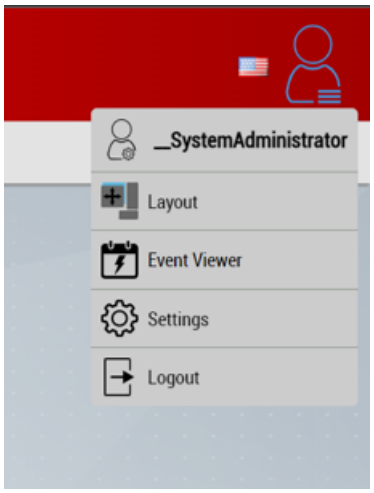
- Editor: Zugriff auf das Objekt gestattet, zudem können Änderungen (wie Layout-Anpassungen bei Content-Objekten) vorgenommen werden.
- Viewer: Zugriff auf das Objekt gestattet – jedoch können keine Änderungen vorgenommen werden.
- No Access: Kein Zugriff auf das Objekt und die dem Objekt im Solution-Explorer untergeordneten Objekte gestattet.

Zugriffsrechte für Content-Objekte lassen sich nachträglich über den Webbrowser anpassen.

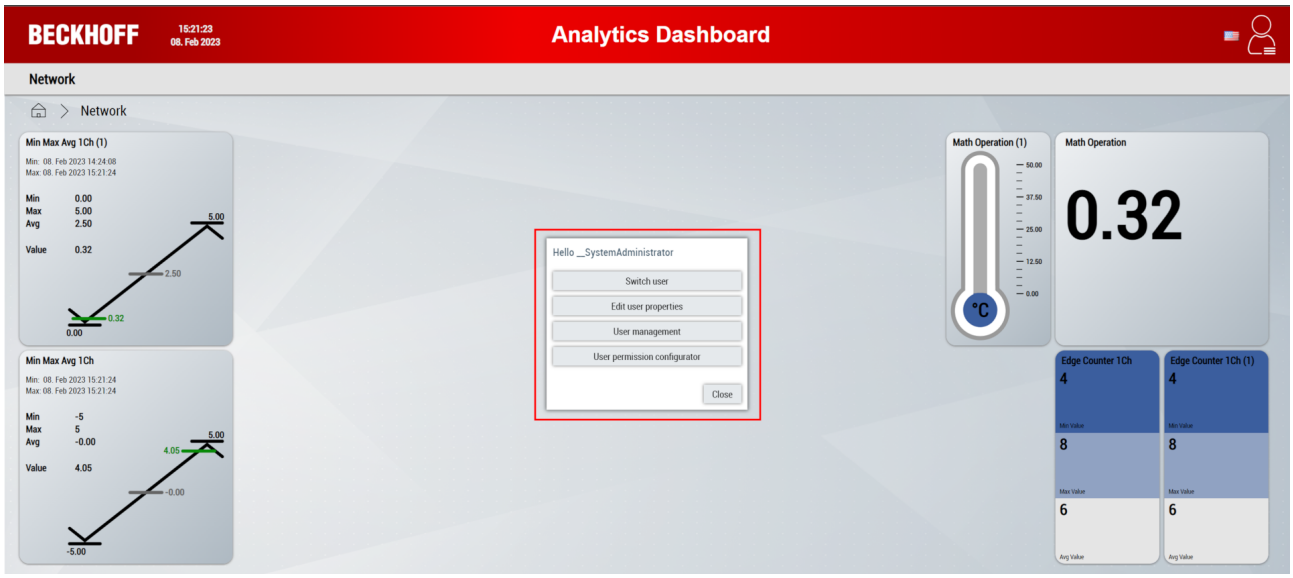
6.8.3.2 Anpassen der Benutzer und Zugriffsrechte im Webbrowser

Beachten Sie, dass das Menü-Control sowie die Konfiguration von Benutzern und Zugriffsrechten erst ab der Dashboard-Version 2.0 verfügbar sind.

Benutzer konfigurieren



Über das Menü-Control können Änderungen an den Benutzern und den Zugriffsrechten vorgenommen werden. Klicken Sie hierfür bitte auf Ihren Benutzernamen (ganz oben im Menü).



Über das Popup (rot umrandet) stehen folgende Optionen zur Verfügung:

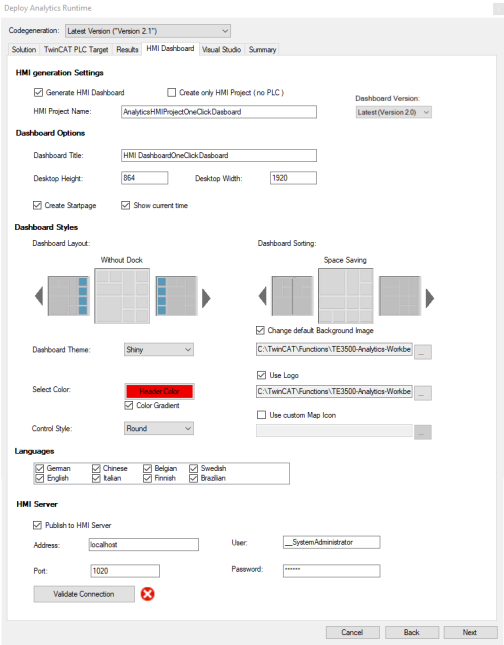
- **Switch user:** Wechsel des angemeldeten Benutzers
- **Edit user properties:** Ändern der Eigenschaften des eigenen Benutzers
- **User management:** Erzeugen von neuen Benutzern, entfernen existierender Benutzer und festlegen von Benutzereigenschaften
- **User permission configurator:** Konfigurieren der Zugriffsrechte von Benutzergruppen auf Content-Objekte

Beachten Sie, dass beim Erzeugen neuer Benutzer individuell konfigurierte Layouts des erzeugenden Benutzers übernommen werden.

6.8.4 Dashboard-Konfiguration

Karteireiter HMI Dashboard

Der Karteireiter **HMI Dashboard** enthält alle Konfigurationen für das Dashboard.



Konfiguration

HMI generation Settings

Generate HMI Dashboard	Aktiviert die Generierung des HMI Dashboards, falls aktiviert. Damit werden die Checkboxen **Create Bootproject** und **Activate PLC Runtime** in dem Karteireiter **TwinCAT PLC Target** automatisch aktiviert/deaktiviert.
Create only HMI Project (No PLC)	Erstellt nur ein HMI-Projekt und kein SPS-Projekt, falls aktiviert.
HMI Project Name	Name des TwinCAT HMI-Projekts.
Dashboard Version	Auswahl der Dashboard-Version.

Dashboard Options

Dashboard Title	Titel des HMI Dashboards, der im Header des Dashboards angezeigt wird.
Desktop Width	Breite des Zielbildschirms in Pixeln.
Desktop Height	Höhe des Zielbildschirms in Pixeln.
Create Start-Page	Erstellt eine Startseite für das Dashboard, auf der eine Karte mit allen Maschinenstandorten angezeigt wird. Die Standortdaten werden aus den Maschinenverwaltungsdaten übernommen.
Show current time	Erstellt eine Uhr im Dashboard-Header, die die aktuelle Ortszeit anzeigt.

Dashboard Styles

Dashboard Layout	Legt das Layout des Dashboards fest. Dock erfordert, dass die „Dock“-Eigenschaft eines Moduls TRUE ist. Dock Left: Fixierte linke Spalte Dock Right: Fixierte rechte Spalte Without Dock: Keine fixierte Spalte
Dashboard Sorting	Legt die Sortierung des Dashboards fest. Space Saving: Ordnet die Controls ohne Freiraum an. Control Type: Ordnet die Controls nach Control-Typ an. Control Size: Ordnet die Controls nach Größe von groß nach klein an. Filled: Ordnet die Controls so an, dass der gesamte Bildschirm ausgefüllt ist. Network Groups: Gruppiert die Controls nach Netzwerk und fasst sie auf einem Bildschirm zusammen. Die Gruppierung beginnt auf der definierten Ebene (0 = Alle, 1 = Erste Ebene…)
Control Distance	Bestimmt den Mindestabstand von Controls im Raster.
Dashboard Theme	Legt das Thema des HMI Dashboards fest, beeinflusst Steuerelemente und Hintergründe: Light: Helles Thema, insbesondere für den Tagmodus. Shiny: Ähnlich wie das helle Thema, Farbverlauf in den Controls. Dark: Dunkles Thema, insbesondere für den Nachtmodus.
Header Color	Legt die Farbe des Dashboard-Headers fest. Dieser enthält einen Farbverlauf, wenn Color Gradient aktiviert ist.
Control Style	Legt den Stil der Controls fest: Round: Die Controls besitzen abgerundete Ecken. Flat: Die Controls besitzen kantige Ecken.
Change default Background Image	Falls aktiviert, kann ein kundenspezifisches Hintergrundbild für das HMI Dashboard festgelegt werden. Wenn nichts festgelegt wird, wird das Standardbild verwendet.
Use Logo	Falls aktiviert, wird dem Dashboard-Header ein Logo hinzugefügt. Es kann ein kundenspezifisches Bild für das Logo festgelegt werden.

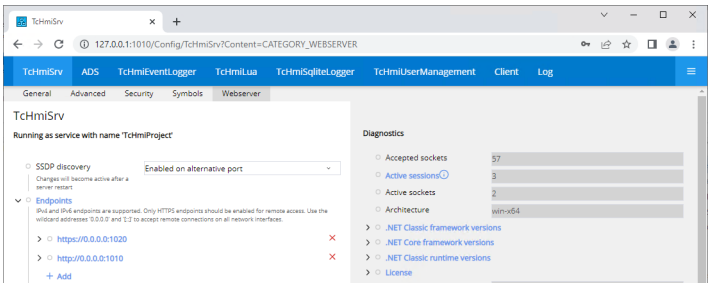
Languages

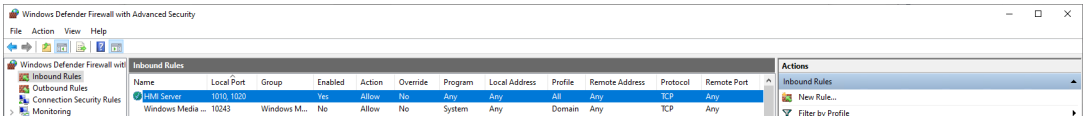
Languages	Auswahl der Sprachen, die im HMI Dashboard verfügbar sind.
Select Default Language	Standardsprache für das Dashboard.

HMI Server

Publish to TwinCAT HMI Server	Veröffentlicht das Dashboard an einen TwinCAT HMI Server, falls aktiviert.
Address	Geben Sie die IP-Adresse oder den Hostnamen des TwinCAT HMI Servers ein.
Port	Geben Sie den Port des TwinCAT HMI Servers ein (standardmäßig 1010 ohne Verschlüsselung und 1020 mit Verschlüsselung).
User	Geben Sie den Benutzernamen ein.
Password	Geben Sie das Passwort ein. (Das Administratorpasswort muss einmalig auf dem HMI Server über die Konfigurationsseite festgelegt werden.)
Validate Connection	Drücken Sie die Schaltfläche, um Ihre Serververbindung zu validieren.

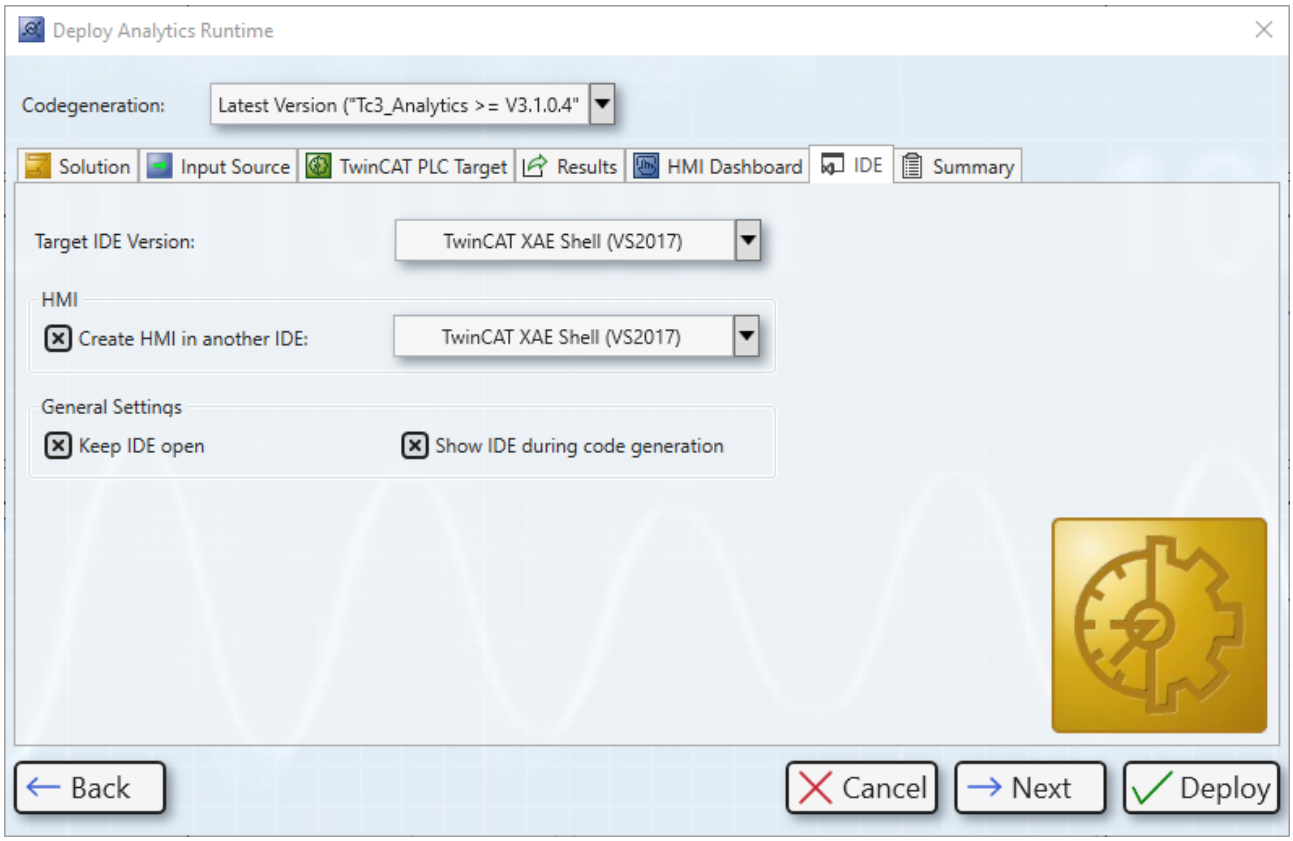
Damit zu einem Remote HMI Server veröffentlicht werden kann, muss der HMI Port und die IP des Entwicklungsrechners im HMI Server freigegeben werden. Ebenfalls muss eine Inbound Rule mit den HMI Ports in der Windows Firewall vom Remote-PC eingerichtet werden.





Karteireiter IDE

Auf dem Karteireiter **IDE** kann ausgewählt werden, welche IDE (Visual Studio® oder TwinCAT XAE Shell) für die Generierung von SPS und HMI verwendet werden soll. Zudem besteht die Möglichkeit, HMI und SPS in zwei verschiedene Lösungen zu generieren.

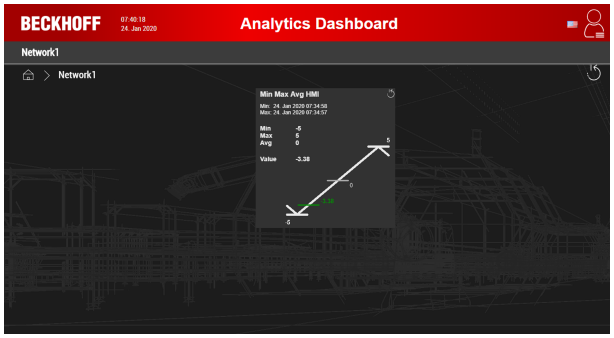


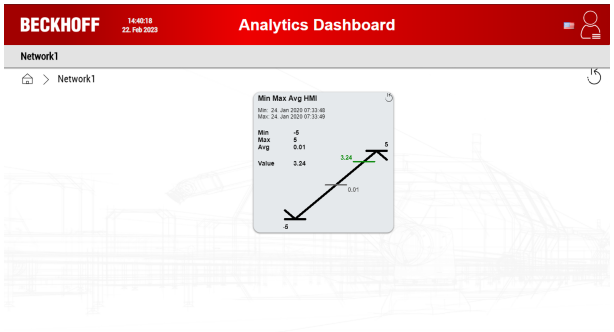
Keep IDE open	Lässt das IDE nach der Generierung geöffnet, falls aktiviert.
Target IDE Version	Die IDE-Zielversion für SPS und HMI.
Create HMI in another IDE	Falls aktiviert, wird die HMI in eine zweite Lösung generiert. Es kann eine andere IDE ausgewählt werden.
Anforderung: TwinCAT bzw. TwinCAT HMI muss in der ausgewählten IDE installiert sein.	

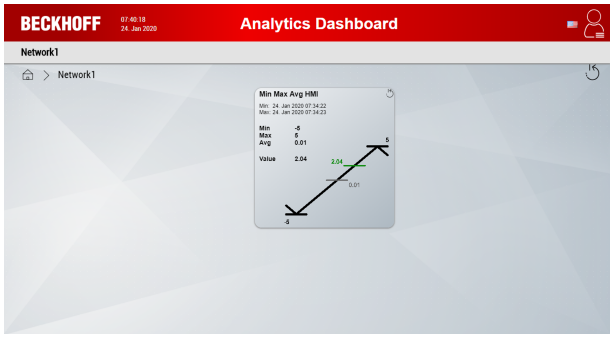
Klicken Sie auf **Next**, um den Karteireiter **Summary** anzuzeigen, und anschließend auf **Deploy**, um die Generierung des Dashboards zu starten.

Impressionen

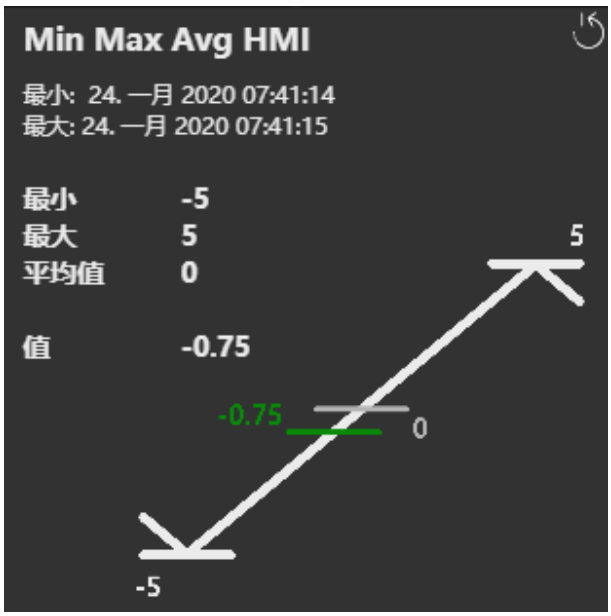
Themen

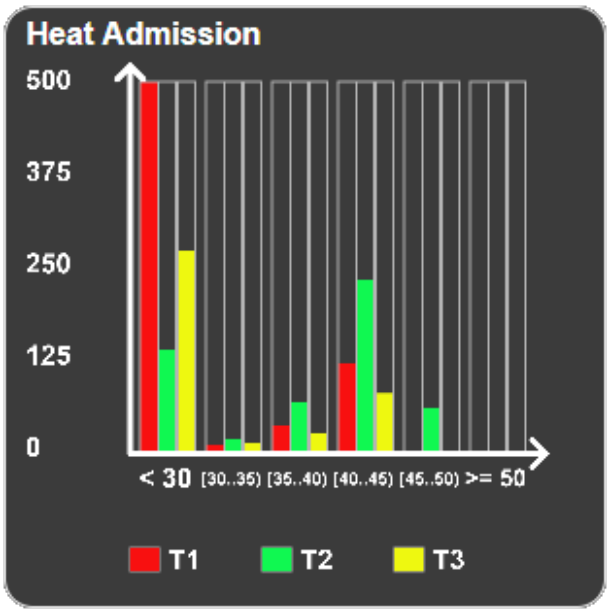




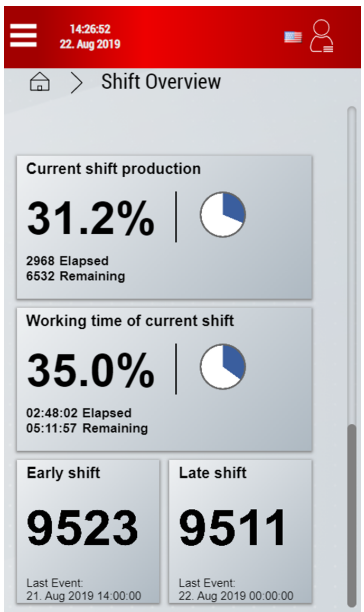


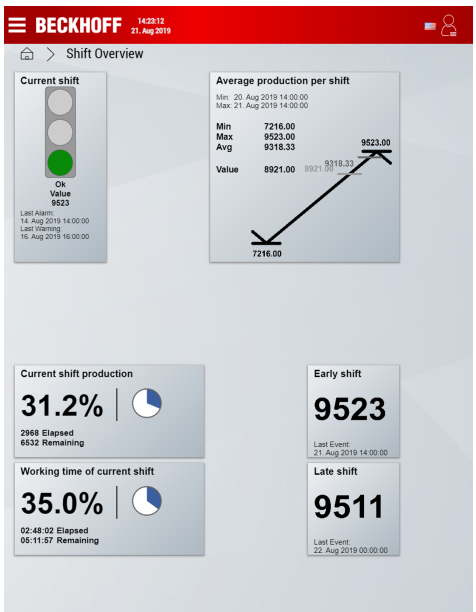
Stile



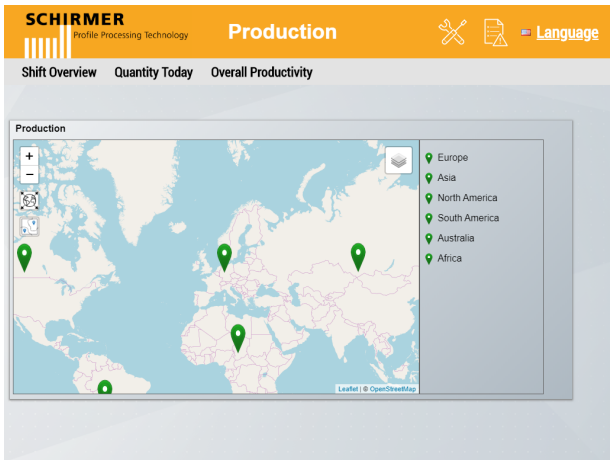


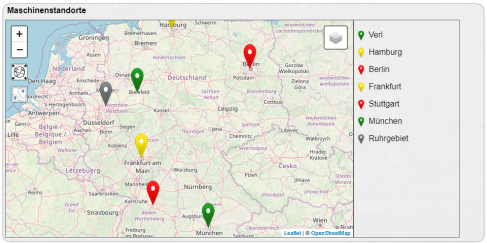
Ansichten für Mobilgeräte



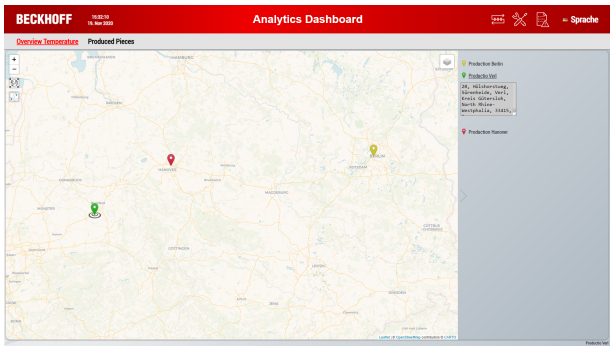


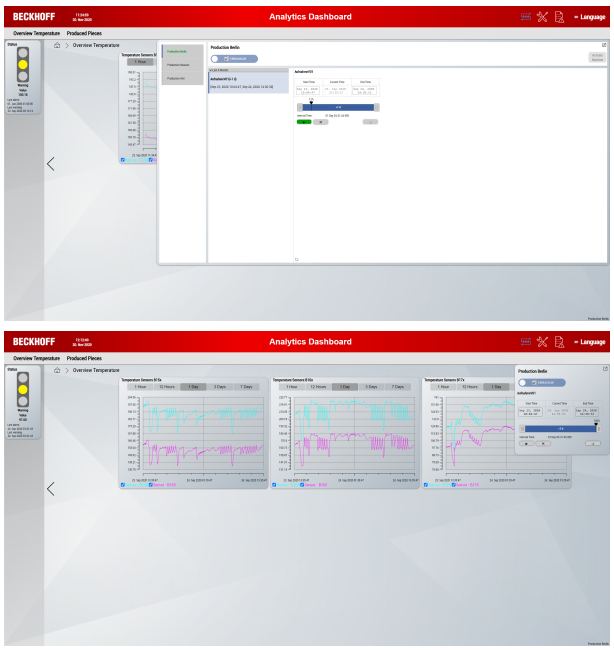
Map auf kundenspezifischem Dashboard





Historische Daten und Maschinenumschaltung





6.8.5 Ändern eines generierten Dashboards im HMI-Engineering

TwinCAT 3 HMI-Projekt

Das Ergebnis der Dashboard-Generierung ist ein vollständiges TwinCAT 3 HMI-Projekt. Daher können alle Möglichkeiten, die das TwinCAT 3 HMI Engineering bietet, genutzt und einbezogen werden.

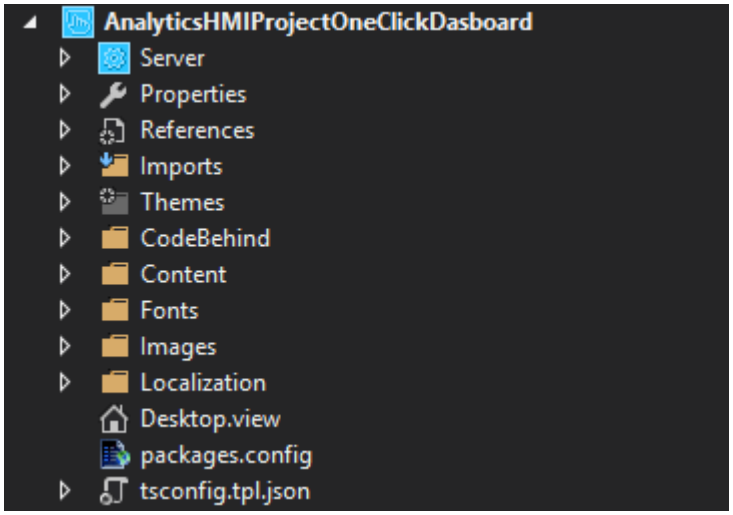


Abb. 5:

Beachten Sie, dass viele Dashboard-Anpassungen auch im Webbrowser zur Laufzeit vorgenommen werden können.

6.8.6 Konfiguration des Dashboards zur Laufzeit im Webbrowser

Zur Laufzeit können im Webbrowser unterschiedliche Konfigurationen vorgenommen werden:

- Erstellen individueller benutzerspezifischer Layouts
- Verwalten von Benutzern (siehe Benutzerverwaltung und Zugriffsrechte konfigurieren)
- Anpassen der Zugriffsrechte auf Contents (siehe Benutzerverwaltung und Zugriffsrechte konfigurieren)
- Ändern von Control-Eigenschaften

- Anpassen von Parametern
- Ändern von globalen Dashboard-Optionen

Beachten Sie, dass bei der Dashboard-Version 1.0 nicht alle Konfigurationsmöglichkeiten verfügbar sind.

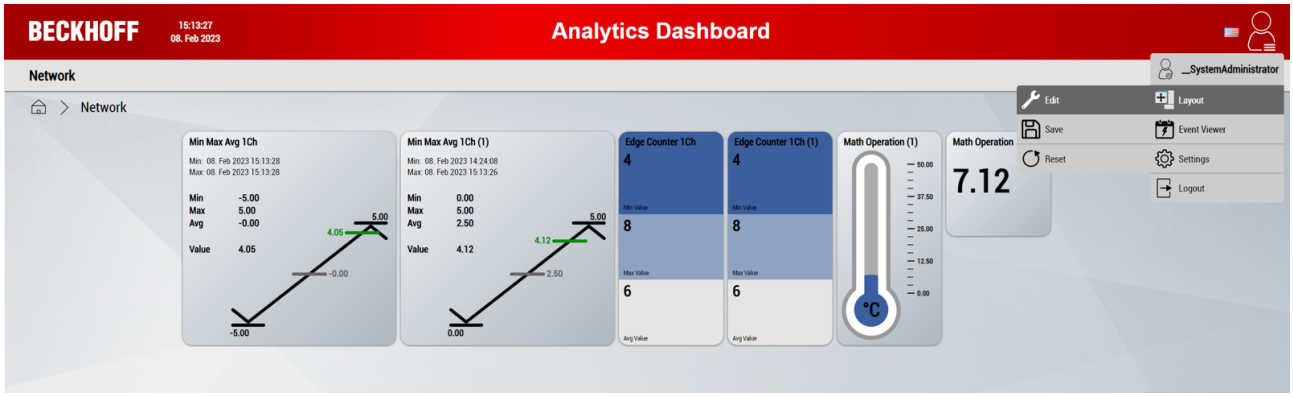
6.8.6.1 Anpassen von Layouts mit der Interactive-Funktion

Mit der Interactive-Funktion lassen sich Controls innerhalb eines Contents individuell benutzerspezifisch anordnen. Das angepasste Layout wird zentral in der HMI-Server-Extension „TcHmiAnalytics“ abgespeichert.

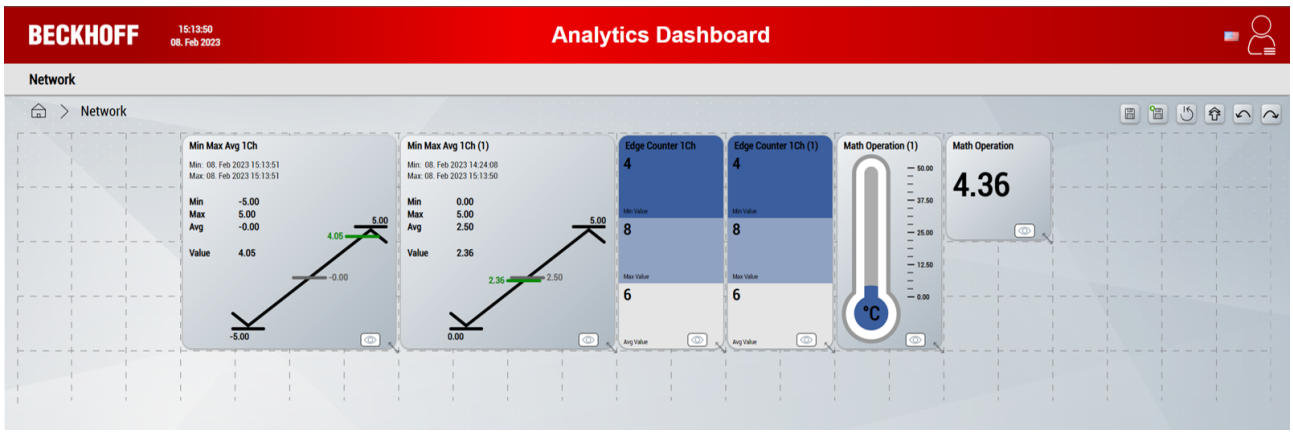
Beachten Sie, dass die Bearbeitung von Layouts erst ab der Dashboard-Version 2.0 verfügbar ist. Zudem kann das Layout nur dann bearbeitet werden, wenn der Benutzer über die notwendigen Zugriffsrechte verfügt.

Bearbeitungsmodus

Die Bearbeitung eines Layouts geschieht in einem speziellen Bearbeitungsmodus. Der Bearbeitungsmodus kann über das Menü-Control aufgerufen werden. Alternativ kann der Bearbeitungsmodus auch mit der Tastenkombination „STRG“ + „E“ gestartet werden.



Im Bearbeitungsmodus ist im Hintergrund ein Gitternetz sichtbar, auf welchem die Controls beliebig angeordnet werden können.

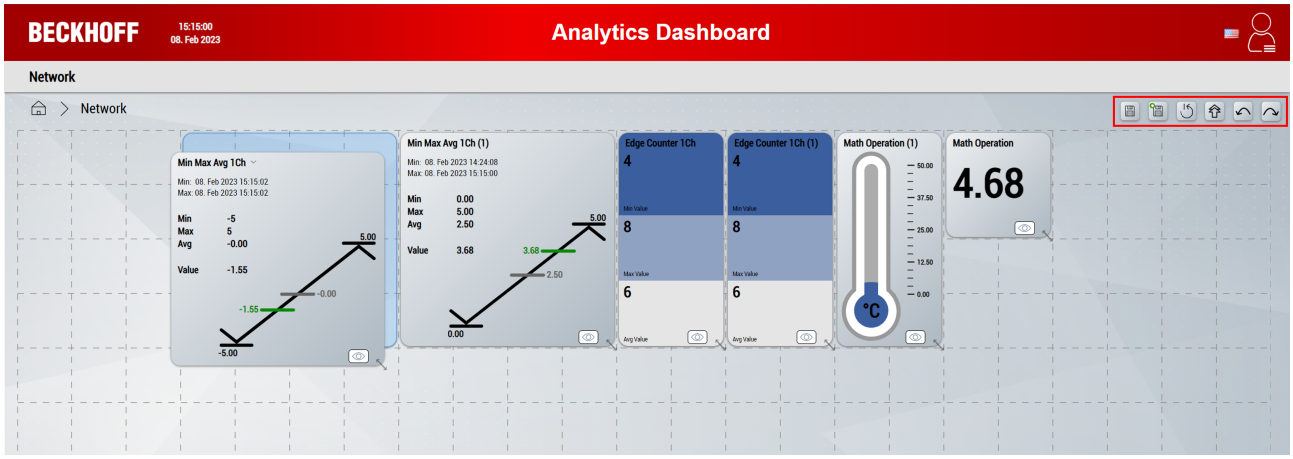


Layouts bearbeiten

Folgende Möglichkeiten stehen zur Verfügung:

- **Positionieren:** Für die Positionierung muss auf ein Control geklickt bzw. gedrückt werden. So wird es aus seiner Verankerung gelöst und ist verschiebbar. Der hellblaue Rahmen zeigt die neue Position innerhalb des Layouts an. Lässt man das Control los, verankert sich dieses an seiner neuen Position.

- **Skalieren:** Mithilfe des Pfeils in der unteren rechten Ecke eines Controls kann dessen Größe verändert werden.
- **Controls ausblenden:** Durch den Sichtbarkeits-Button (unten rechts innerhalb eines Controls) kann dieses ein- und ausgeblendet werden. Ausgeblendete Controls werden im Bearbeitungsmodus halb-transparent dargestellt.



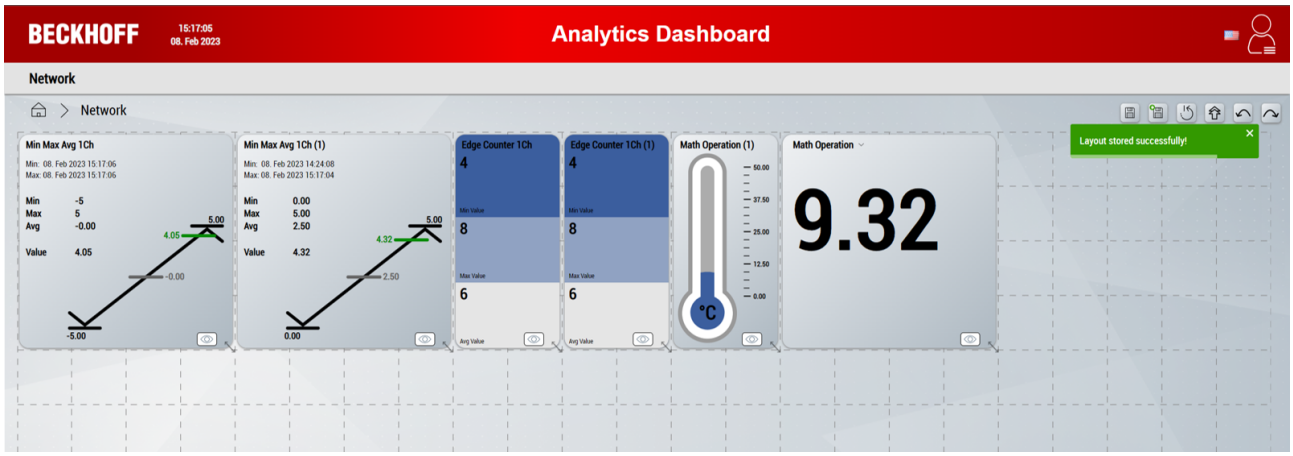
Bearbeitungsmenü

Im Bearbeitungsmodus können über ein zusätzliches Menü (rot umrandet) unterschiedliche Aktionen vorgenommen werden:

- Speichern des Layouts (alternativ Tastenkombination „STRG“ + „S“)
- Speichern des Layouts und Verlassen des Bearbeitungsmodus (alternativ Tastenkombination „STRG“ + „Q“)
- Zurücksetzen der durchgeführten Änderungen (alternativ Tastenkombination „STRG“ + „R“)
- Controls nach oben verschieben, sodass freie Zeilen entfernt werden (alternativ Tastenkombination „STRG“ + „U“)
- Letzte Aktion rückgängig machen (alternativ Tastenkombination „STRG“ + „Z“)
- Letzte Aktion wiederholen (alternativ Tastenkombination „STRG“ + „Y“)

Layouts speichern

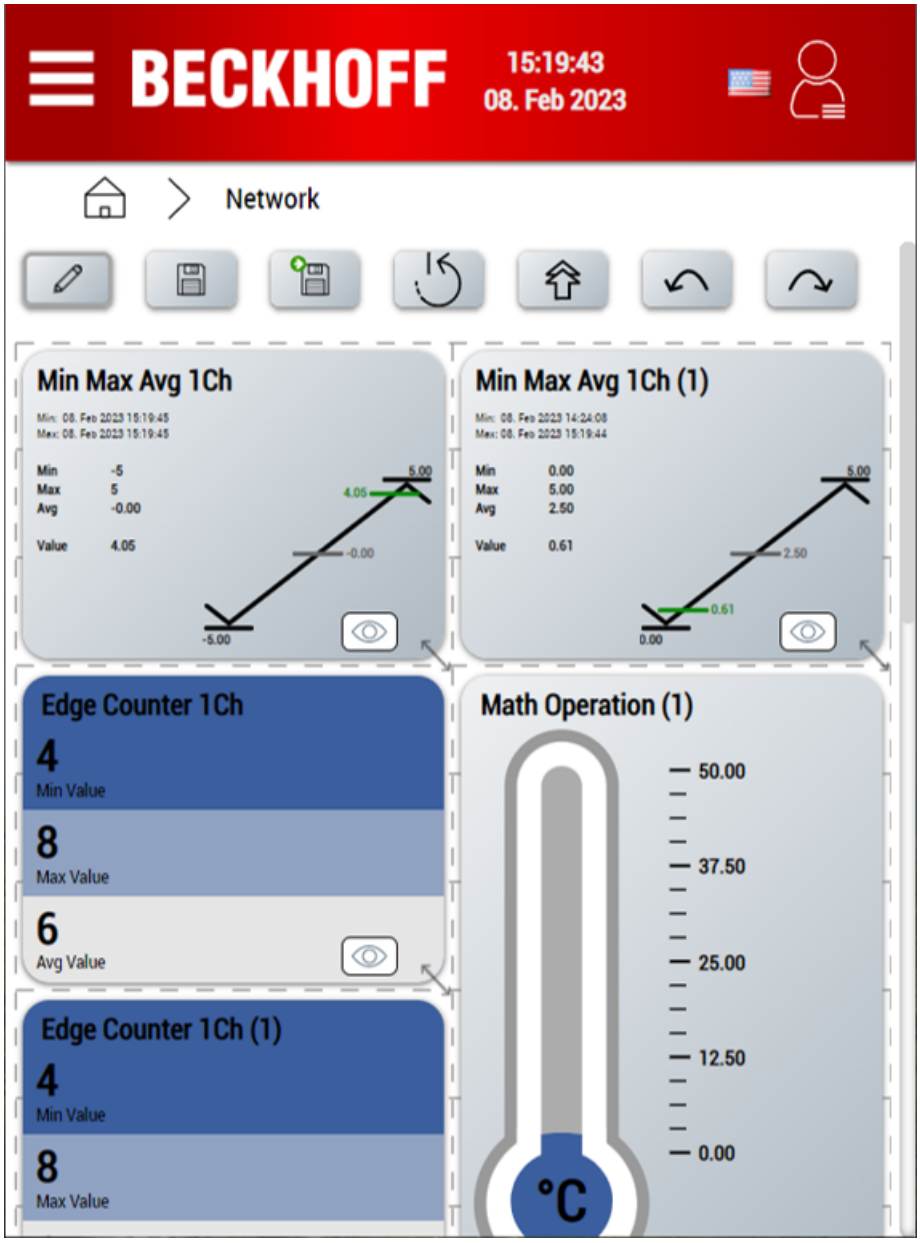
Angepasste Layouts können über das Bearbeitungsmenü oder über das Menü-Control gespeichert werden. Ein Popup zeigt das Ergebnis des Speichervorgangs.



Wenn ein Fehler beim Speichern auftritt, prüfen Sie in der Server-Konfiguration, ob die Extension „TcHmiAnalytics“ aktiviert ist.

Mobiles Layout

Für mobile Geräte kann zusätzlich ein benutzerspezifisches mobiles Layout generiert werden. Dieses ist unabhängig von dem normalen Layout und besitzt lediglich 2 Spalten. Die Bearbeitung und das Speichern erfolgt auf dieselbe Art.



6.8.6.2 Ändern von Control-Eigenschaften im Browser

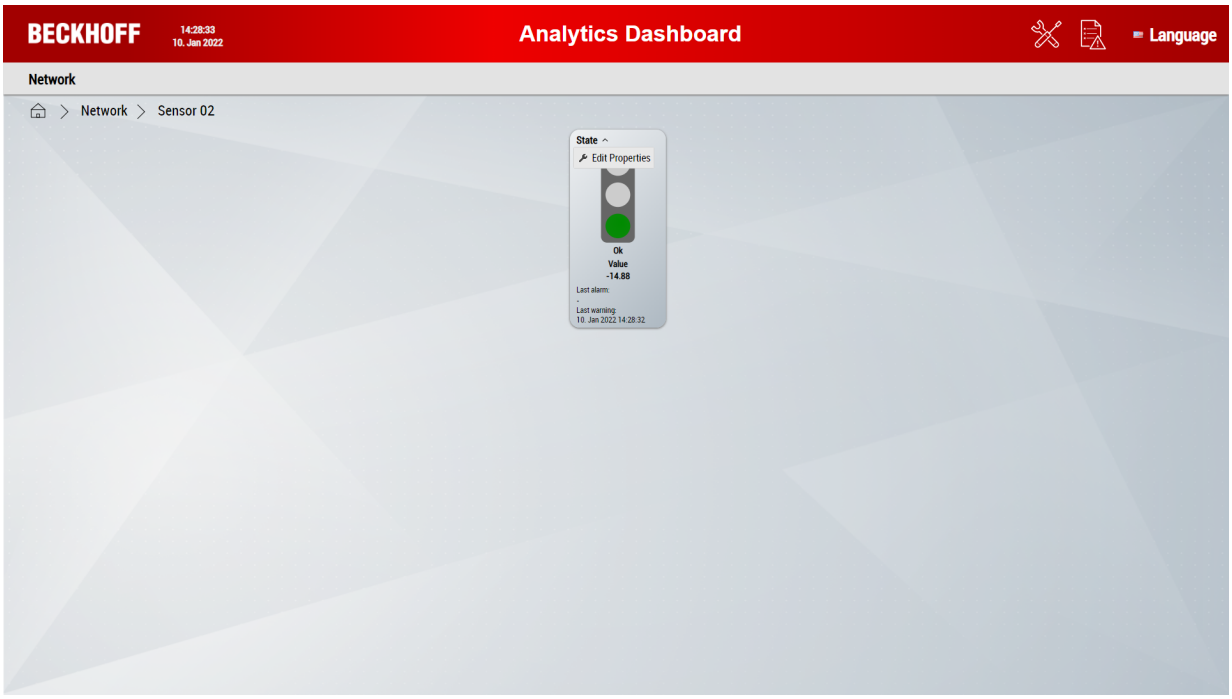
Empfehlung: Änderungen im Analytics Workbench Projekt durchführen

In der Regel können die hier vorgenommen Änderungen bereits in der Analytics Workbench vorgenommen werden, bevor Sie das Dashboard generieren (siehe Mapping Wizard oder auch Dashboard-Knoten). Dies hat den Vorteil, dass die Änderungen bei einer erneuten Dashboard-Generierung gespeichert bleiben.

Jedes Analytics Control besitzt ausgewählte Eigenschaften, die dynamisch im Browser geändert werden können. Diese Möglichkeit ist ab der Analytics Version 3.4.3145.0 bzw. mit der Control Paket Version 1.1.31 verfügbar. Sie können damit bei älteren HMI Projekten einfach das NuGet Paket aktualisieren.

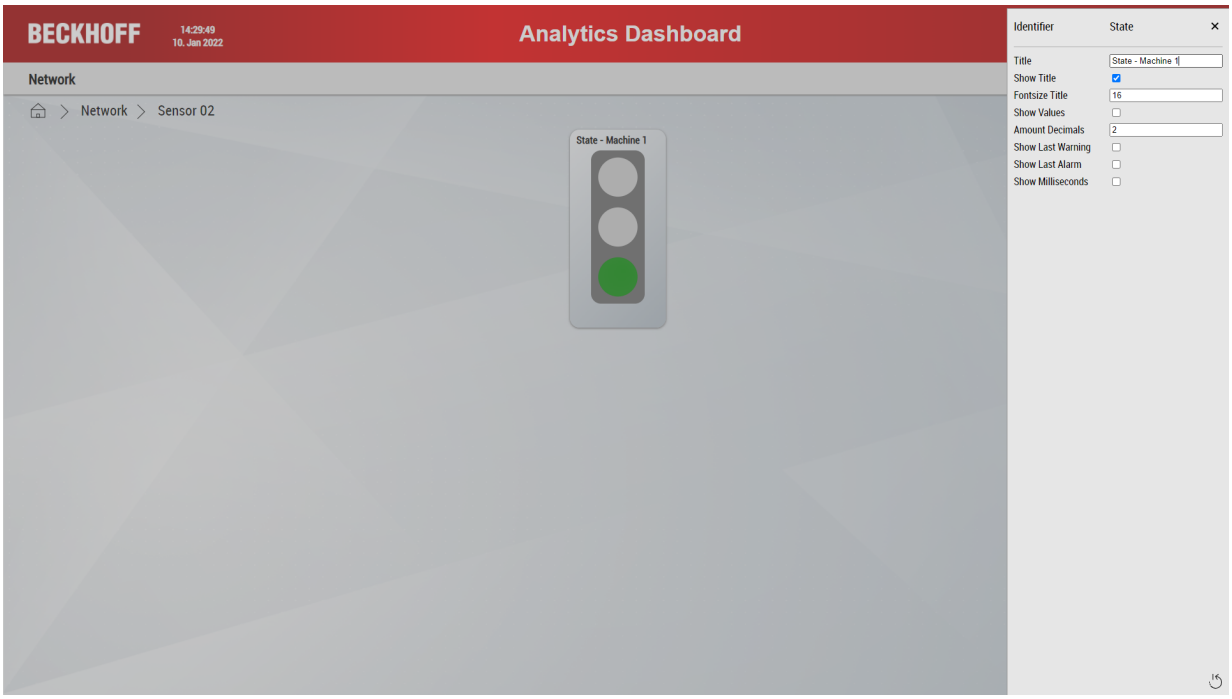
1. Öffnen Sie ein generiertes HMI Projekt und klicken Sie auf den Titel eines Controls, bei dem Sie eine Eigenschaft ändern wollen.

2. Dort klappt sich ein Menü aus, bei dem Sie auf **Edit Properties** klicken.



⇨ Jedes Control besitzt unterschiedliche Eigenschaften, die angepasst werden können. Hier werden zu Demonstrationszwecken die Eigenschaften des *Traffic Light* Controls geändert

3. Ändern Sie den Titel und haken die unteren drei Auswahlfelder ab, um lediglich die Ampel mit dem geänderten Titel anzuzeigen.

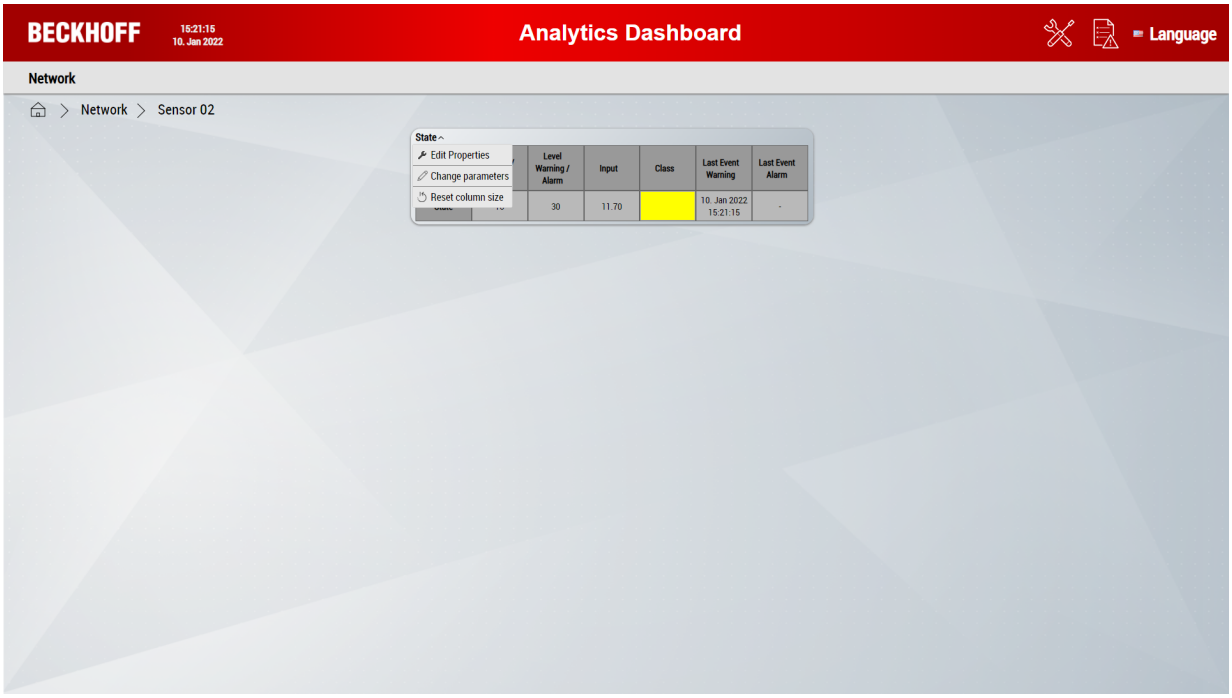


⇨ Die vorgenommenen Änderungen werden Client-Seitig (auf dem Gerät, auf dem Sie das Dashboard öffnen) gespeichert und lediglich dort angezeigt. Zukünftig wird die Speicherung über den HMI Server durchgeführt, wodurch Änderungen global auf jedem Gerät durchgeführt werden.

6.8.6.3 Ändern von Parametern im Browser

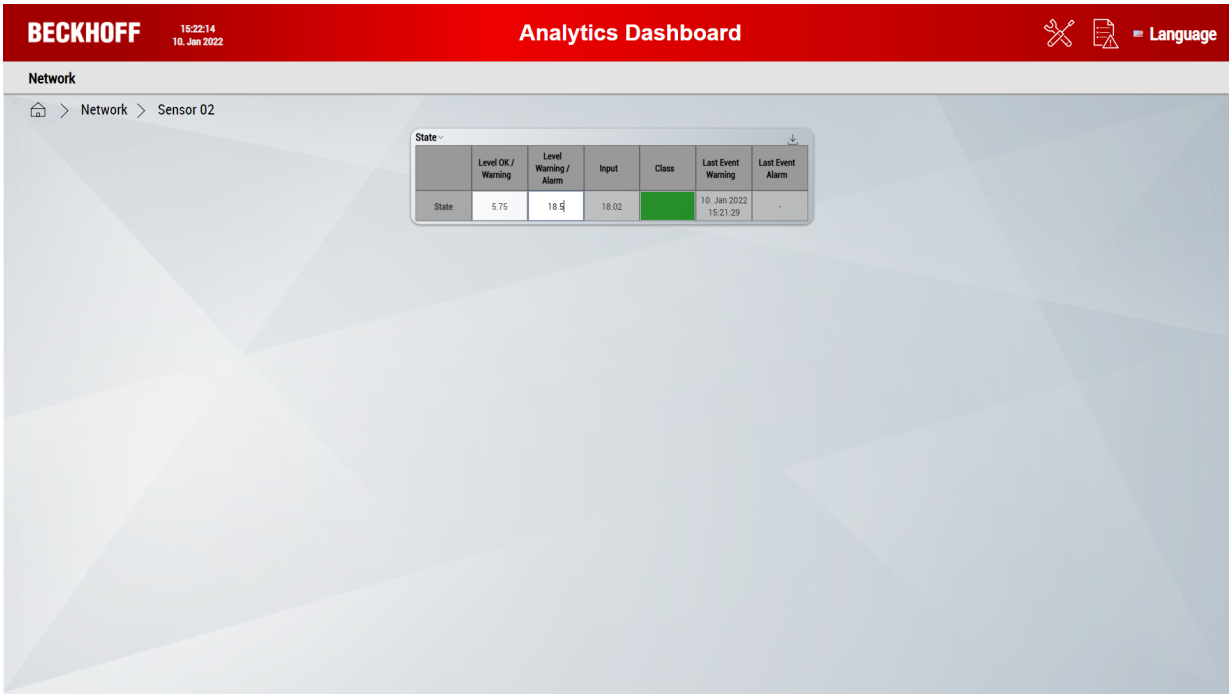
Numerische Parameter wie Grenzwerte können dynamisch im Dashboard geändert werden. Dies ist mit dem Data Table Control möglich. Die Werte werden dort persistent zurück in die SPS geschrieben und sind damit auch nach einem Neustart verfügbar.

1. Öffnen Sie ein generiertes HMI Projekt und klicken Sie auf den Titel des Tabellen Controls, bei dem Sie ein Parameter ändern wollen.
2. Dort klappt sich ein Menü aus, bei dem Sie auf **Change parameters** klicken. Ein Modul ohne Parameter hat diesen Menüeintrag nicht.

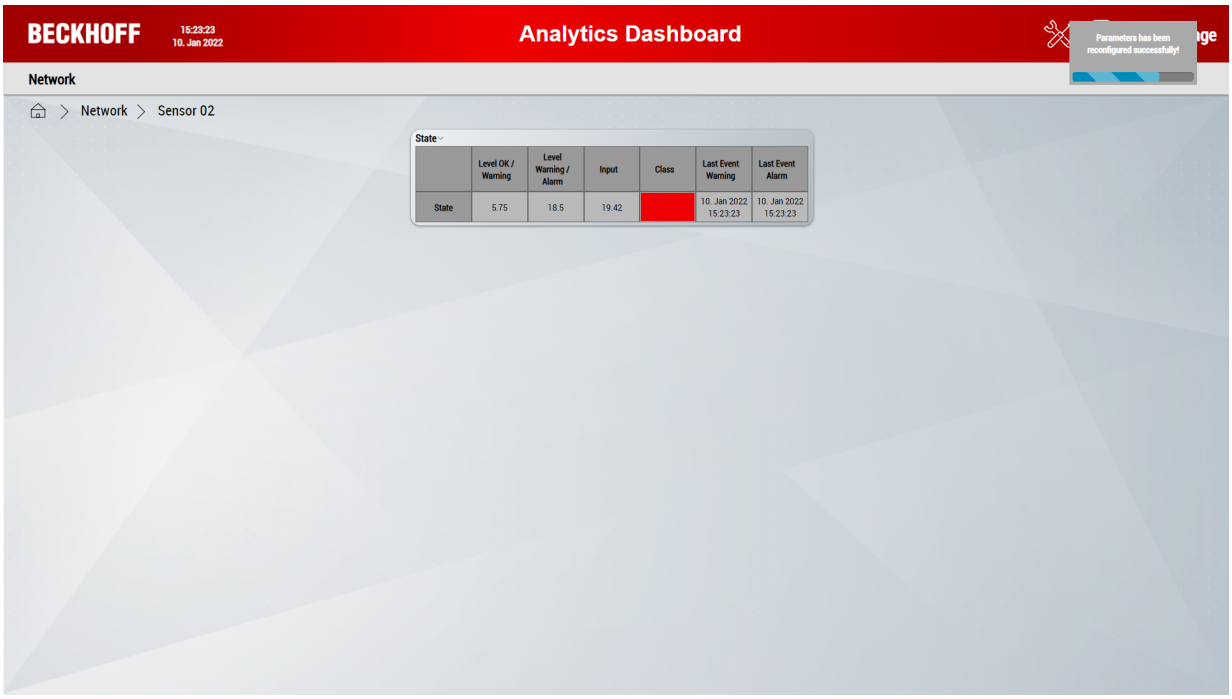


⇨ Die änderbaren Felder wechseln zu einem Eingabefeld mit weißem Hintergrund.

3. Ändern Sie dort die Werte und klicken Sie anschließend oben rechts in der Ecke auf das Speichern-Symbol.

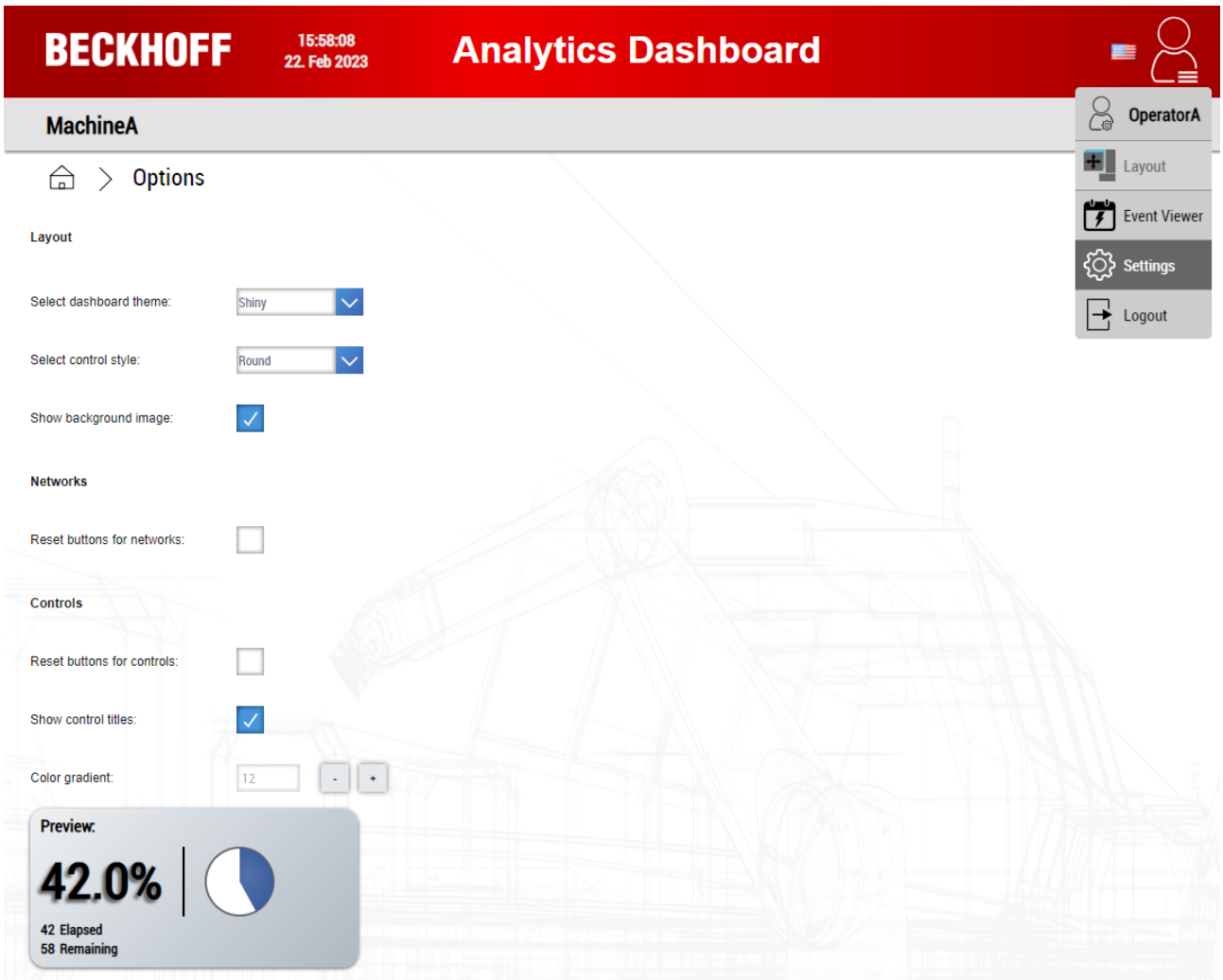


⇨ Nach dem Speichern der Parameter wird Ihnen eine Meldung im oberen rechten Bereich des Dashboards angezeigt. Dort wird Ihnen mitgeteilt, ob die Neukonfiguration funktioniert hat.



6.8.6.4 Anpassen von globalen Dashboard-Optionen

Auf der Options-Seite können generelle Einstellungen für das Dashboard bearbeitet werden.



Layout	
Select dashboard theme	Ändern Sie das Dashboard-Thema zwischen Shiny, Dark und Light.
Select control style	Ändern Sie den Steuerelementstil zwischen Flat und Round.
Show background image	Zeigt das Standard- oder ein kundenspezifisches Hintergrundbild an.
Networks	
Reset buttons for networks	Aktiviert Reset-Schaltflächen für ganze Netzwerke. Zurücksetzen aller Controls innerhalb des ausgewählten Netzwerks.
Controls	
Reset buttons for controls	Aktiviert Reset-Schaltflächen für Controls. Zurücksetzen eines einzelnen Controls.
Show control titles	Aktiviert Control-Titel für alle Controls.

6.8.7 Erzeugen von Diagrammen im Webbrowser

Mit der neuen Interactive-Chart-Funktion können Sie Ihre Dashboards beliebig mit Diagrammen erweitern. Das Hinzufügen sowie die Konfiguration von Diagrammen sind direkt im Webbrowser möglich. Das Engineering ist hierfür nicht erforderlich. Als Datenquelle für die Diagramme können Daten, welche mit dem Analytics Storage Provider historisiert wurden, verwendet werden. Bei der Konfiguration können Sie die darzustellenden Variablen, die Zeitbereiche sowie den Diagrammtyp auswählen.

Analytics Storage Provider Update erforderlich

Damit die Daten korrekt geladen werden können, ist ein **TwinCAT Analytics Storage Provider >= Version 1.0.12.3 erforderlich**

Für die Verwendung der Interactive-Chart-Funktion werden folgende NuGet-Packages benötigt:

Beckhoff.TwinCAT.Analytics.Controls >= 1.35.0

Beckhoff.TwinCAT.Analytics.ControlsExtension >= 1.35.0

Bei bereits generierten HMI-Projekten lässt sich die Interactive-Chart-Funktion durch das Updaten der NuGet-Packages integrieren.

6.8.7.1 Verwalten der Datenquellen

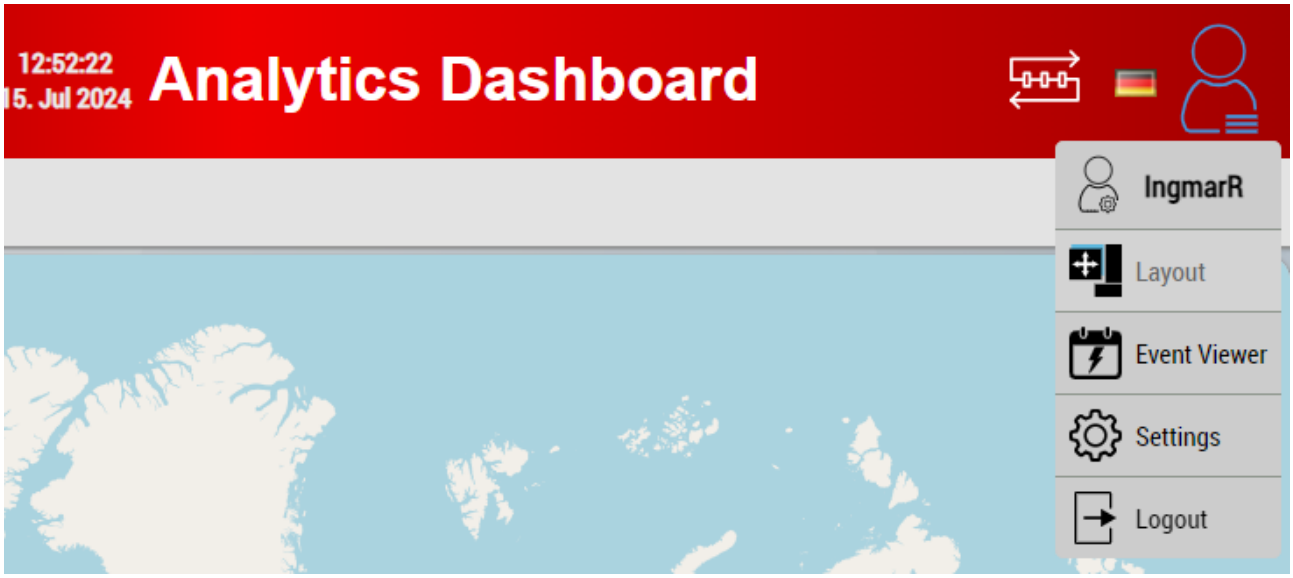
Als Datenquelle für die Diagramme können Daten, welche mit dem Analytics Storage Provider historisiert wurden, verwendet werden. Die Verwaltung der Datenquellen kann direkt über das Dashboard vorgenommen werden.

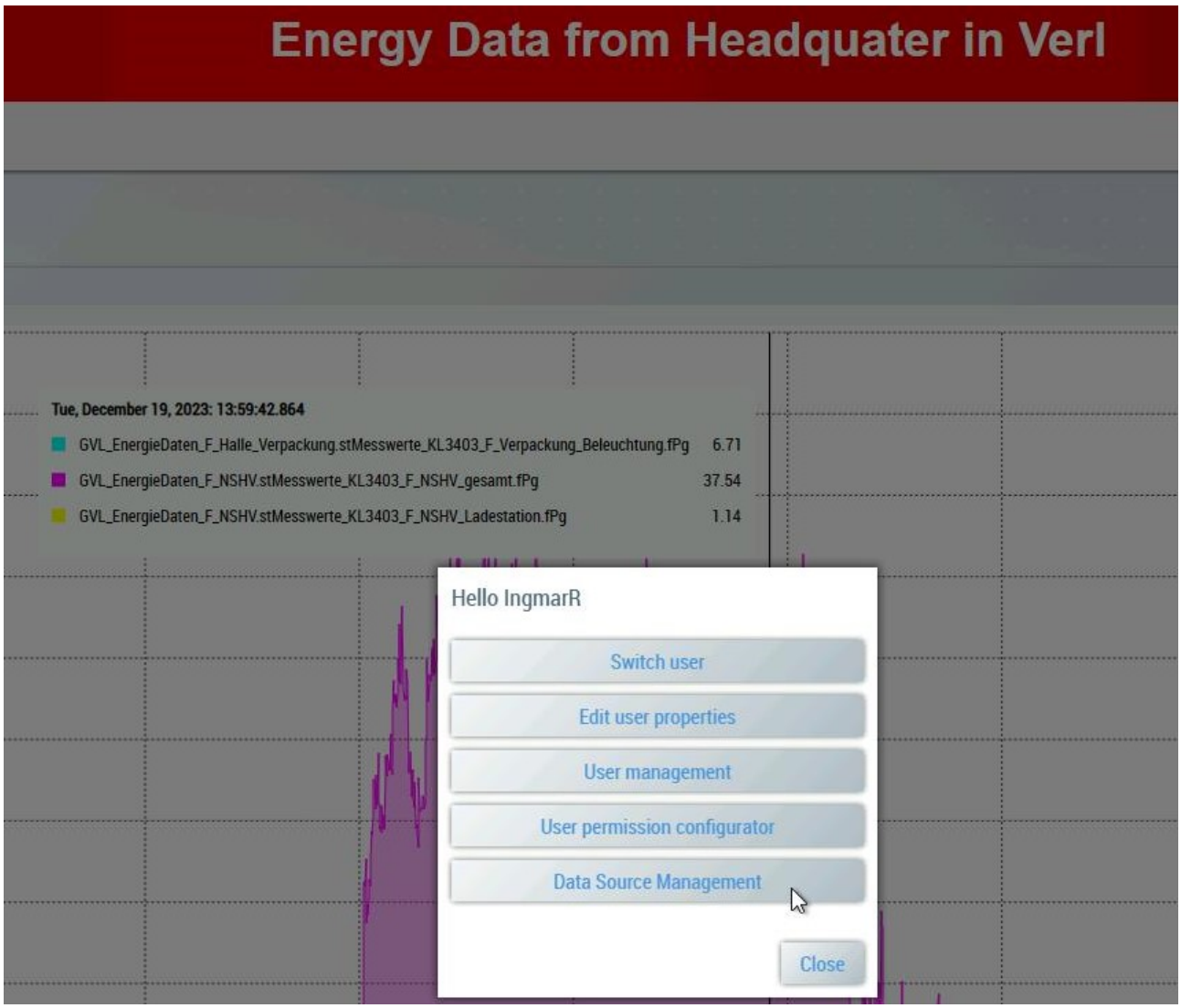
Automatisches Hinzufügen von historischen Sources bei der Codegenerierung

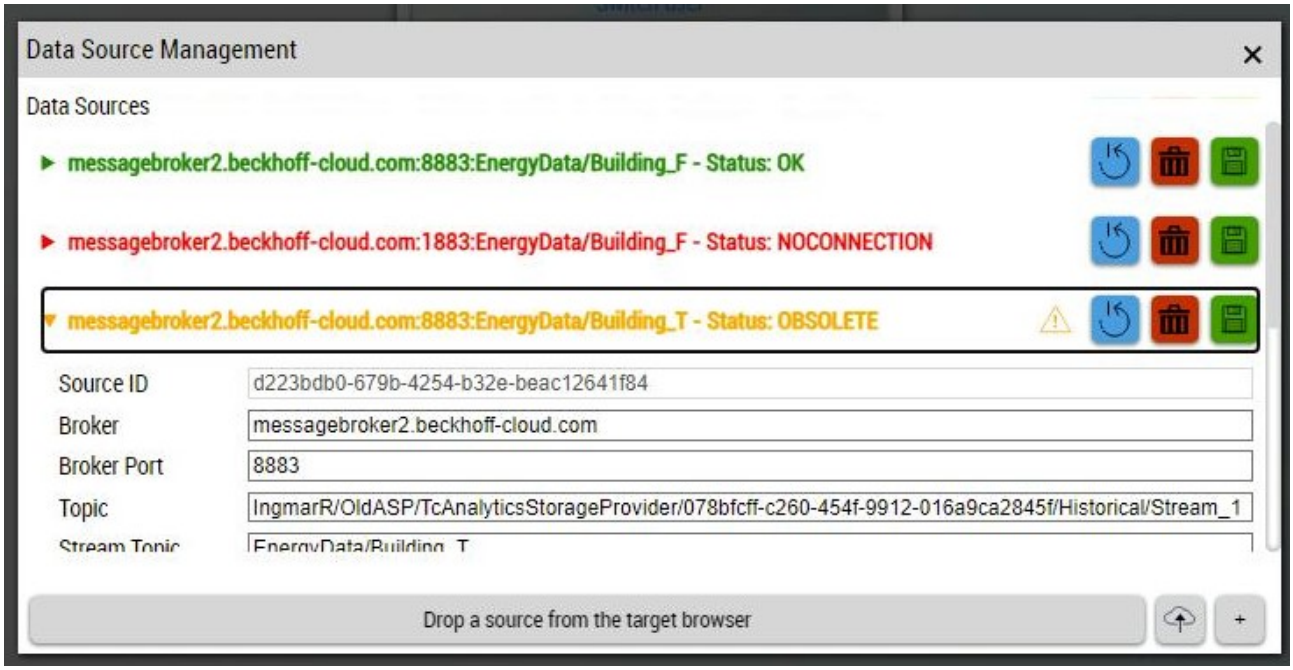
Ab TwinCAT Analytics Workbench Version >= 49.0 werden bei der Codegenerierung automatisch alle historischen Datenquellen aus dem Projekt in das Dashboard übernommen. Dabei spielt es keine Rolle, ob diese im Projekt aktiv verwendet werden oder lediglich als Quellen hinzugefügt wurden. Beachten Sie bitte, dass Livedaten-Quellen nicht verwendet werden können.

Öffnen der Datenquellenverwaltung

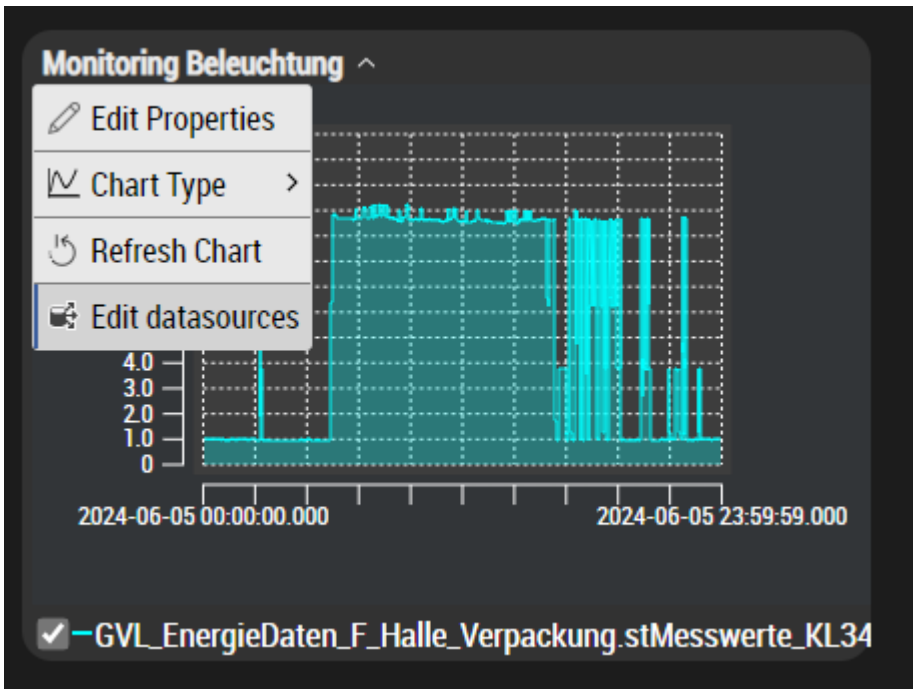
Die Verwaltung von Datenquellen kann nur durch Benutzer der Gruppe „__SystemAdministrators“ oder „Administrators“ vorgenommen werden. Die Datenquellenverwaltung können Sie über das Menü-Icon in der oberen rechten Ecke öffnen: **Menü - <Benutzername> - Data Source Management**.







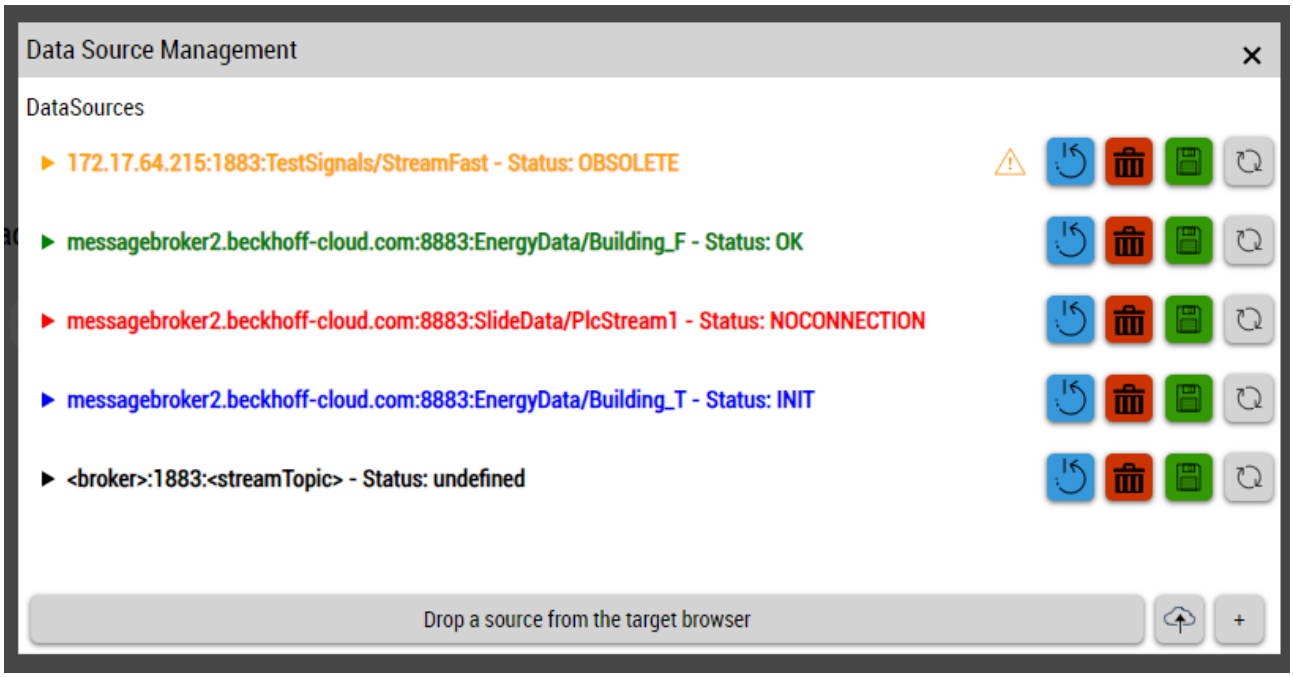
Alternativ können Sie die Datenquellenverwaltung auch über ein existierendes **Interactive-Chart** öffnen.



Übersicht der Datenquellen

In der Datenquellenverwaltung werden alle vorhanden Datenquellen dargestellt. Zusätzlich wird zu jeder Datenquelle der Status angezeigt. Dieser hat folgende Bedeutungen:

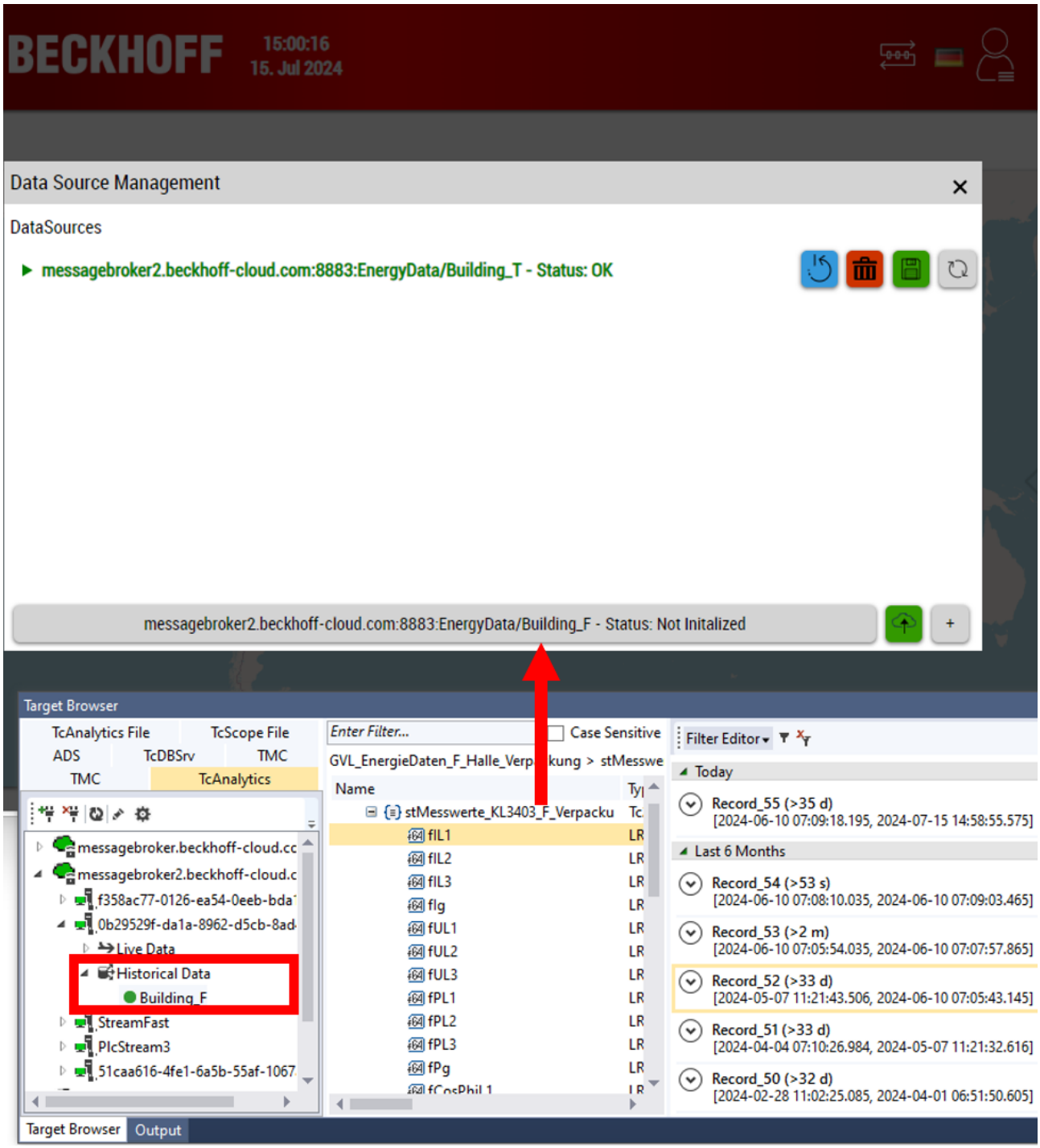
- OK: Die Datenquelle wurde korrekt konfiguriert.
- NOCONNECTION: Die Datenquelle wurde fehlerhaft konfiguriert. In diesem Fall sollten die Verbindungsparameter überprüft und korrigiert werden.
- OBSOLETE: Der Storage Provider der Daten ist veraltet und sollte mindestens auf Version >= 1.0.12.3 geupdatet werden. Andernfalls können Daten nicht korrekt abgefragt werden.
- INIT: Es wurde eine neue Konfiguration an die Server-Extension gesendet. Diese versucht sich mit der Datenquelle zu verbinden.
- UNDEFINED: Die Konfiguration ist lediglich lokal angelegt und wurde noch nicht in der Server-Extension gespeichert.
- NOSOURCEHANDLER: Die Konfiguration konnte nicht korrekt in der Server-Extension geladen werden und sollte erneut gespeichert werden.



Anlegen einer Datenquelle

Neue Datenquellen können einfach per Drag-and-Drop aus dem TargetBrowser hinzugefügt werden.
Wählen Sie unter dem Reiter **TcAnalytics** historische Datenquellen aus.
Es werden automatisch alle Symbole der Datenquelle hinzugefügt.

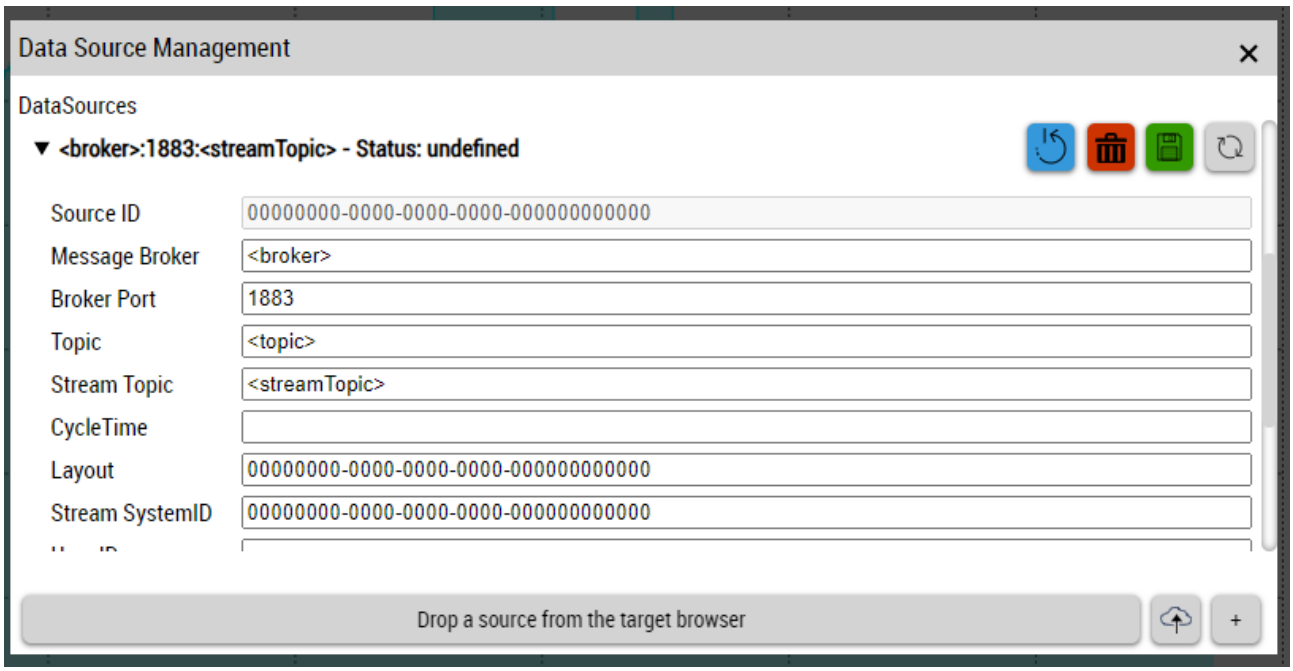
Wurden die Parameter erfolgreich ausgelesen, leuchtet das Upload-Icon grün. Über einen Klick auf das Upload-Icon wird die Konfiguration an die Server-Extension gesendet.



Steht der Target Browser auf dem Gerät nicht zur Verfügung, können Sie eine leere Datenquelle über das Plus-Icon in der unteren rechten Ecke anlegen. Diese muss manuell konfiguriert werden.

Konfigurieren und Ändern einer Datenquelle

Mit dem Pfeil links von einer Datenquelle können Sie diese aufklappen und bearbeiten.



Folgende Parameter können gesetzt werden:

Parameter	Beschreibung
Source ID	Eindeutige ID einer Datenquelle. Diese ist einzigartig und wird von der Server-Extension vergeben.
Message Broker	Adresse des Message Brokers
Broker Port	Port des Message Brokers
Topic	MQTT-Topic des historischen Streams „…/TcAnalyticsStorageProvider/…/Historical/Stream…“
StreamTopic	Originales Topic des aufgezeichneten Streams.
Cycle Time	Zykluszeit der Datenquelle in 100-Nanosekunden.
Layout	Layout des Streams
Stream SystemID	SystemID des aufzeichneten Streams.
UserID	Benutzername für den Zugriff auf den Message Broker.
Passwort	Passwort für den Zugriff auf den Message Broker.
Mit Zertifikaten	Wenn der Wert angehakt ist, werden die ausgewählten Zertifikatsdateien verwendet.
CA	CA-Datei (kann zur Server-Extension hochgeladen werden).
CERT	CERT-Datei (kann zur Server-Extension hochgeladen werden).
KEY	KEY-Datei (kann zur Server-Extension hochgeladen werden) .
Key Password	Passwort zu der KEY-Datei.

Die Änderungen können über das Speicher-Icon (grün) an die Server-Extension gesendet werden. Alternativ können die lokal gemachten Änderungen über das Reset-Icon (blau) wieder rückgängig gemacht werden. Über das Refresh-Icon (grau) kann der Status der Datenquelle erneut abgefragt werden.

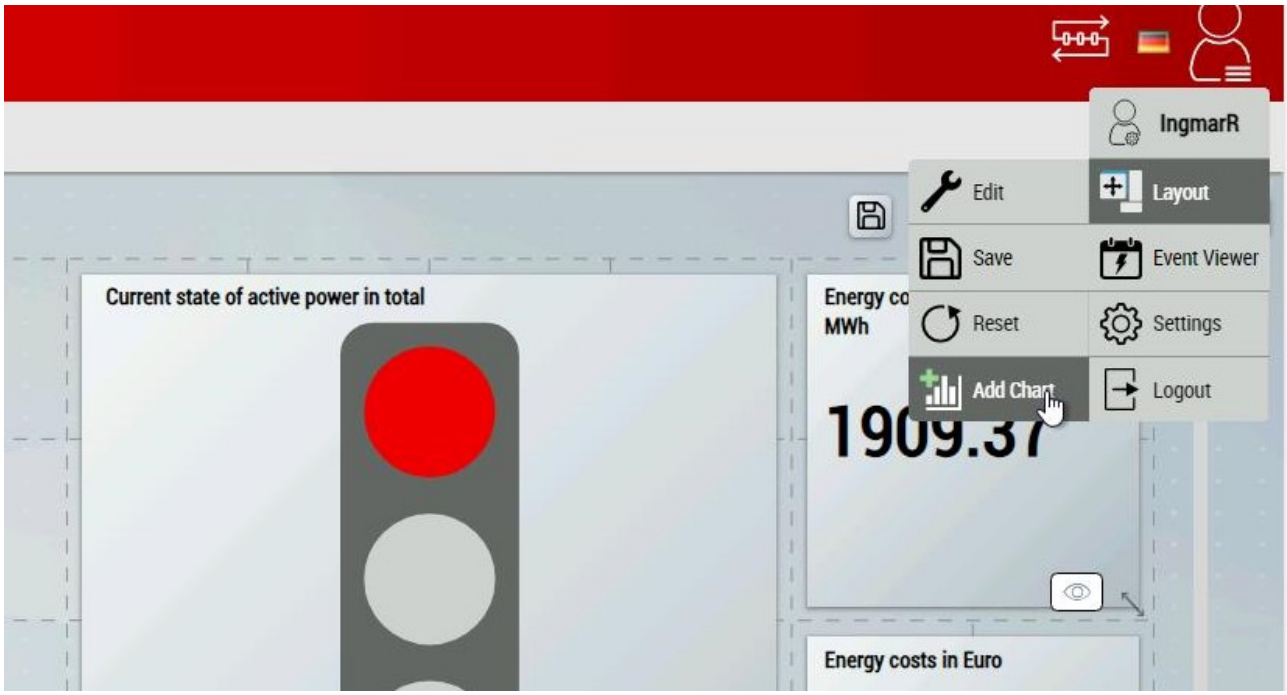
Löschen einer Datenquelle

Eine Datenquelle kann über das dazugehörige Löschen-Icon (rot) gelöscht werden. Bitte beachten Sie, dass alle Diagramme, welche diese Datenquelle verwenden, nicht mehr funktionieren werden. Bei einer Änderung von Parametern, wie beispielsweise der Broker-Adresse oder der Erneuerung von Zertifikaten, ist es empfehlenswert, bestehende Datenquellen anzupassen und diese nicht zu löschen.

6.8.7.2 Hinzufügen und Entfernen von Diagrammen

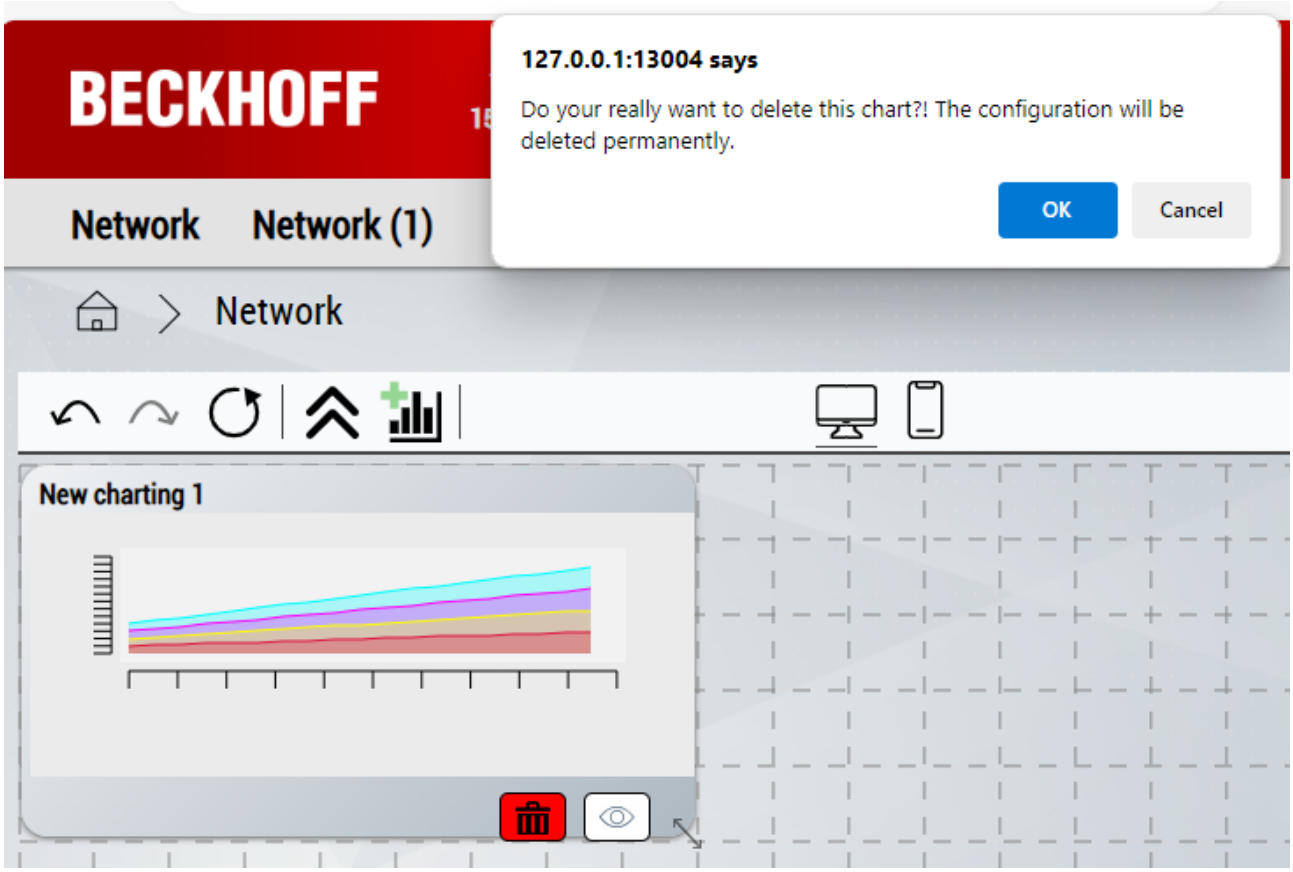
Im Webbrowser

Fügen Sie neue Diagramme über das Menü-Icon hinzu. Klicken Sie unter der Kategorie Layout auf **Add Chart.** Das neue Diagramm wird an der ersten freien Position im Layout eingefügt.



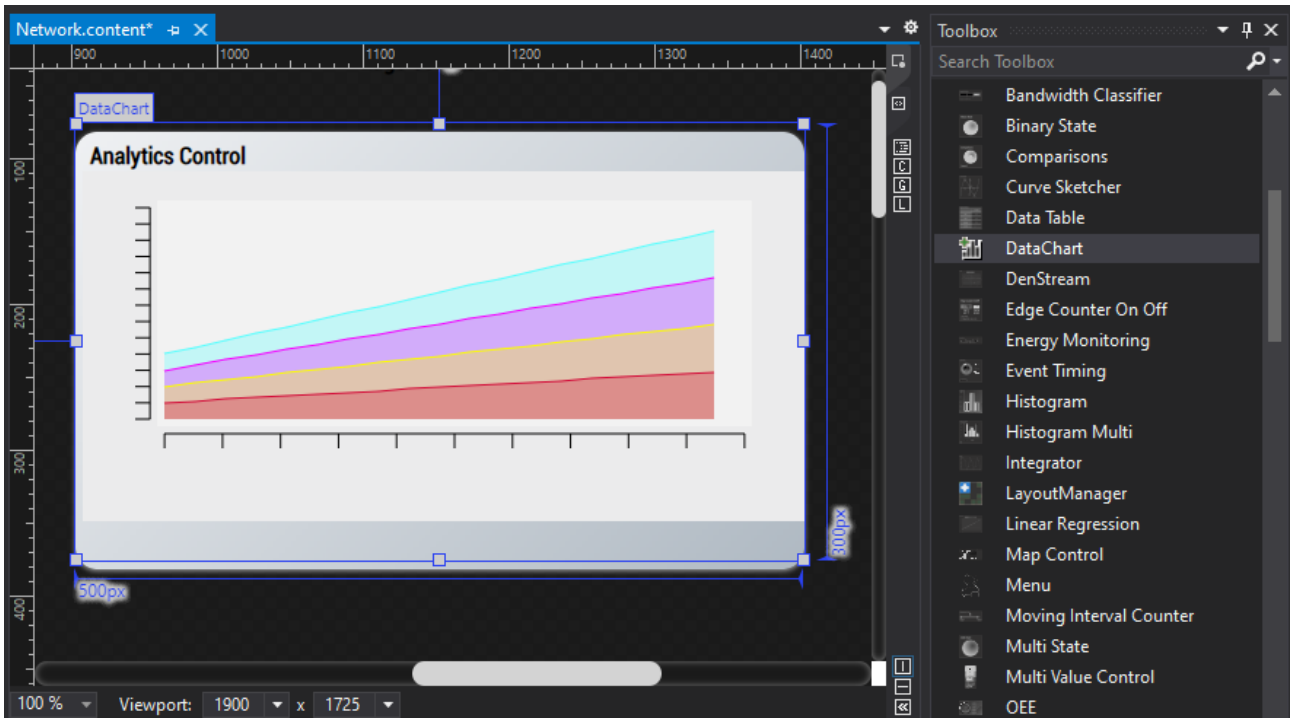
Das Hinzufügen von Diagrammen ist nur möglich, wenn der Benutzer über die Berechtigung zum Bearbeiten des Layouts verfügt.

Das hinzugefügte Diagramm ist benutzerspezifisch und nur für den aktuellen Benutzer einsehbar. Es kann beliebig in seiner Position und Größe innerhalb des Layouts verschoben werden. Das Entfernen eines Diagramms ist im Bearbeitungsmodus über das Papierkorb-Icon in der unteren rechen Ecke möglich.



Im HMI-Engineering

Diagramme können ebenfalls über das HMI-Engineering hinzugefügt werden. Diese Diagramme sind standardmäßig für alle Anwender vorhanden und können nicht entfernt werden. Die Konfiguration [▶ 490] im Engineering hinzugefügten Diagramme erfolgt im Webbrowser.

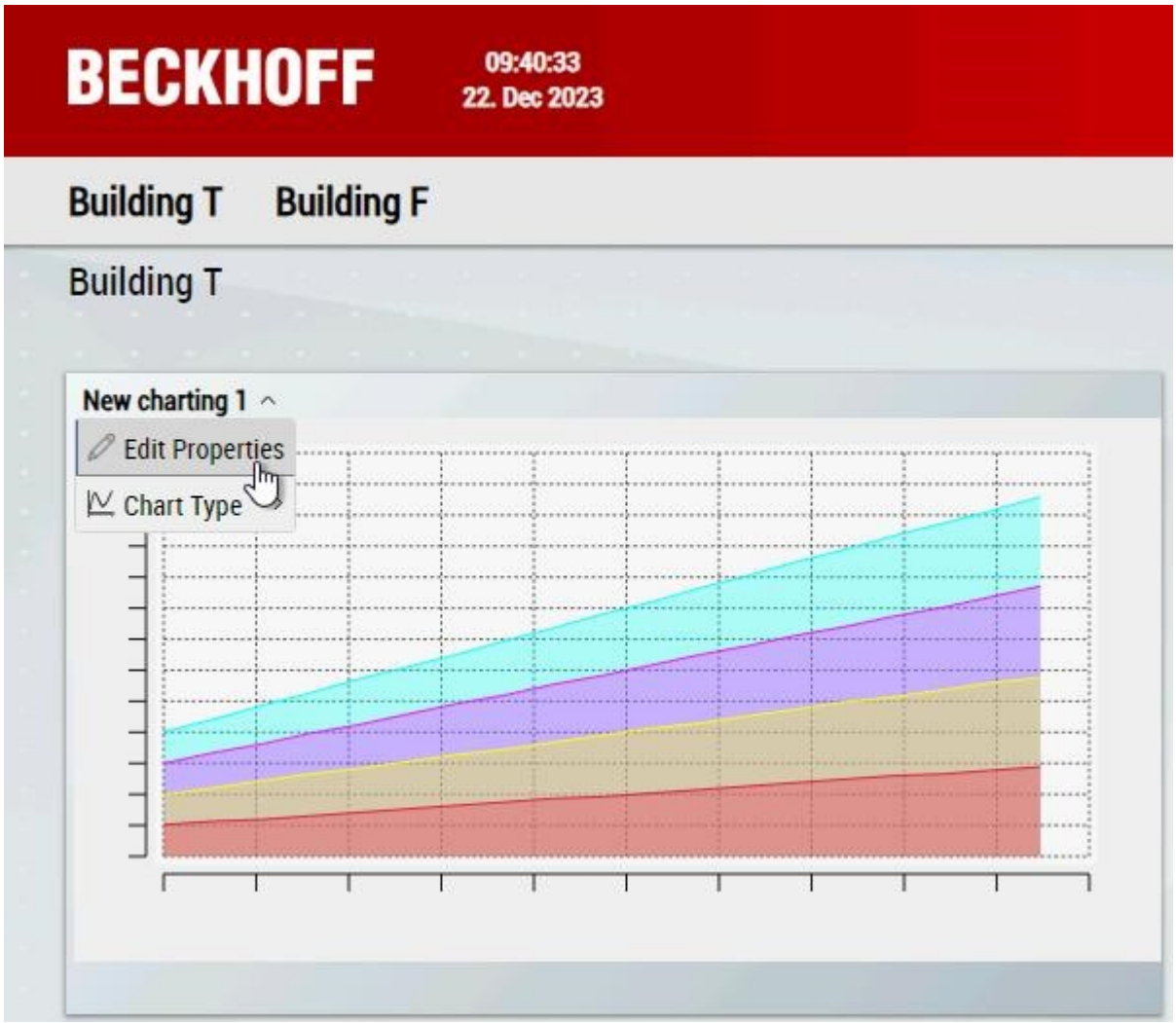


6.8.7.3 Konfiguration eines Diagramms

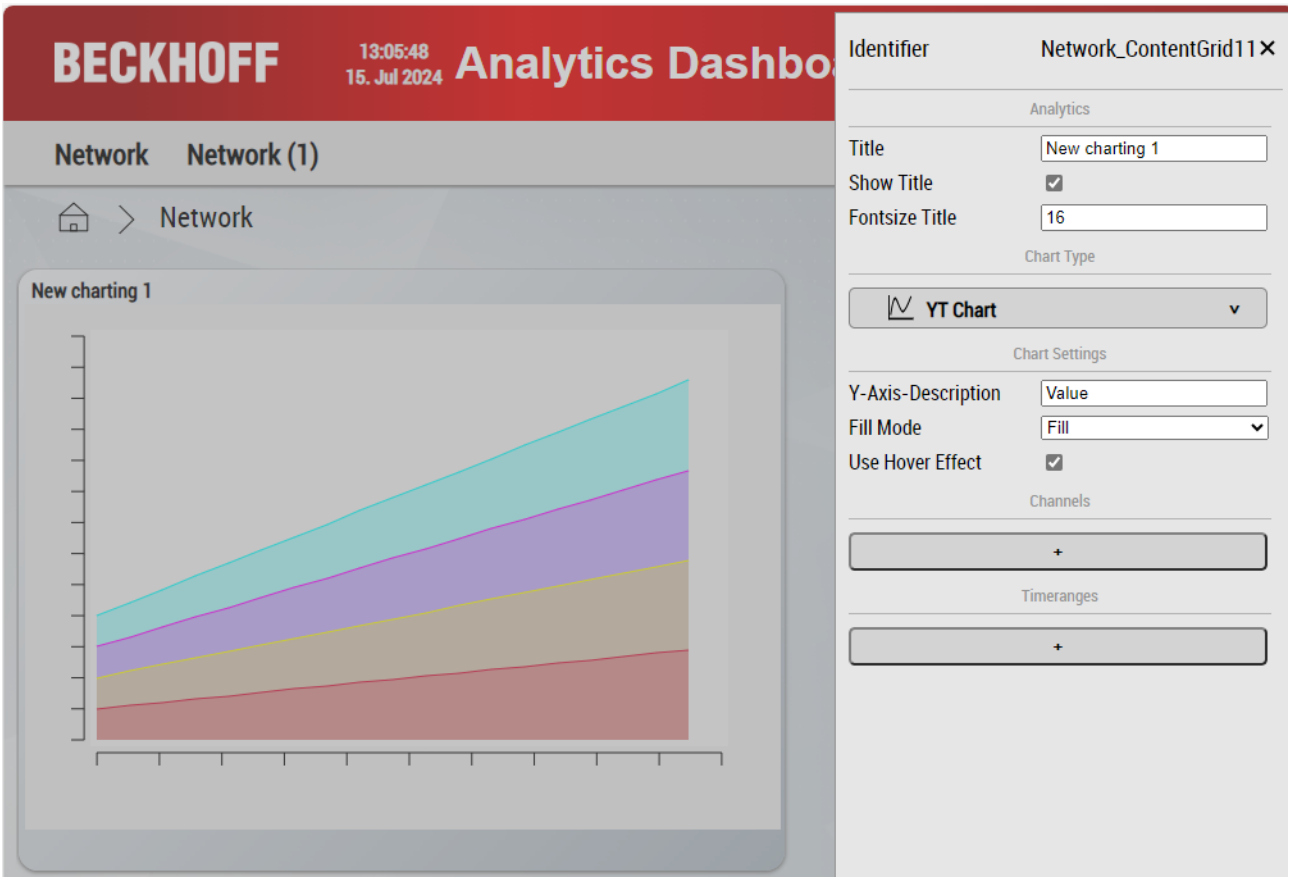
Die Konfiguration eines Diagramms erfolgt im Webbrowser. Sie können alle Einstellungen (wie Diagrammtyp, Variablen und Zeiträume) über das Eigenschaften-Fenster des Diagramms bearbeiten.

Öffnen der Diagrammeigenschaften

Öffnen Sie die Diagrammeigenschaften über einen Klick auf den Titel des Chart-Controls.



Mit einem Klick auf **Edit Properties** öffnen Sie die Diagrammeigenschaften.

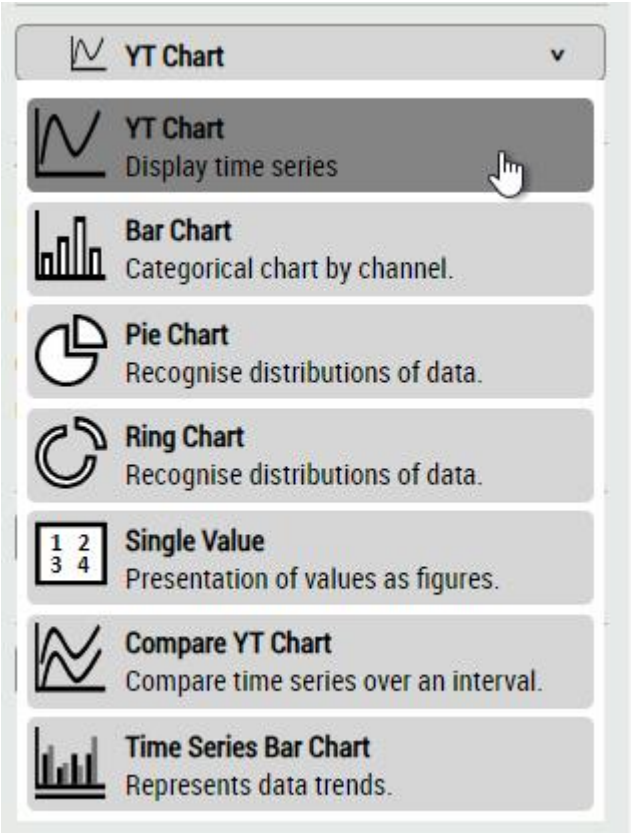


Die Diagrammeigenschaften sind in 4 Kategorien unterteilt.

- Chart Type: Auswahl des Diagrammtyps
- Chart Settings: Konfiguration des Diagramms
- Channels: Auswahl der Variablen und deren Darstellung.
- Timeranges: Auswahl der darzustellenden Zeitbereiche.

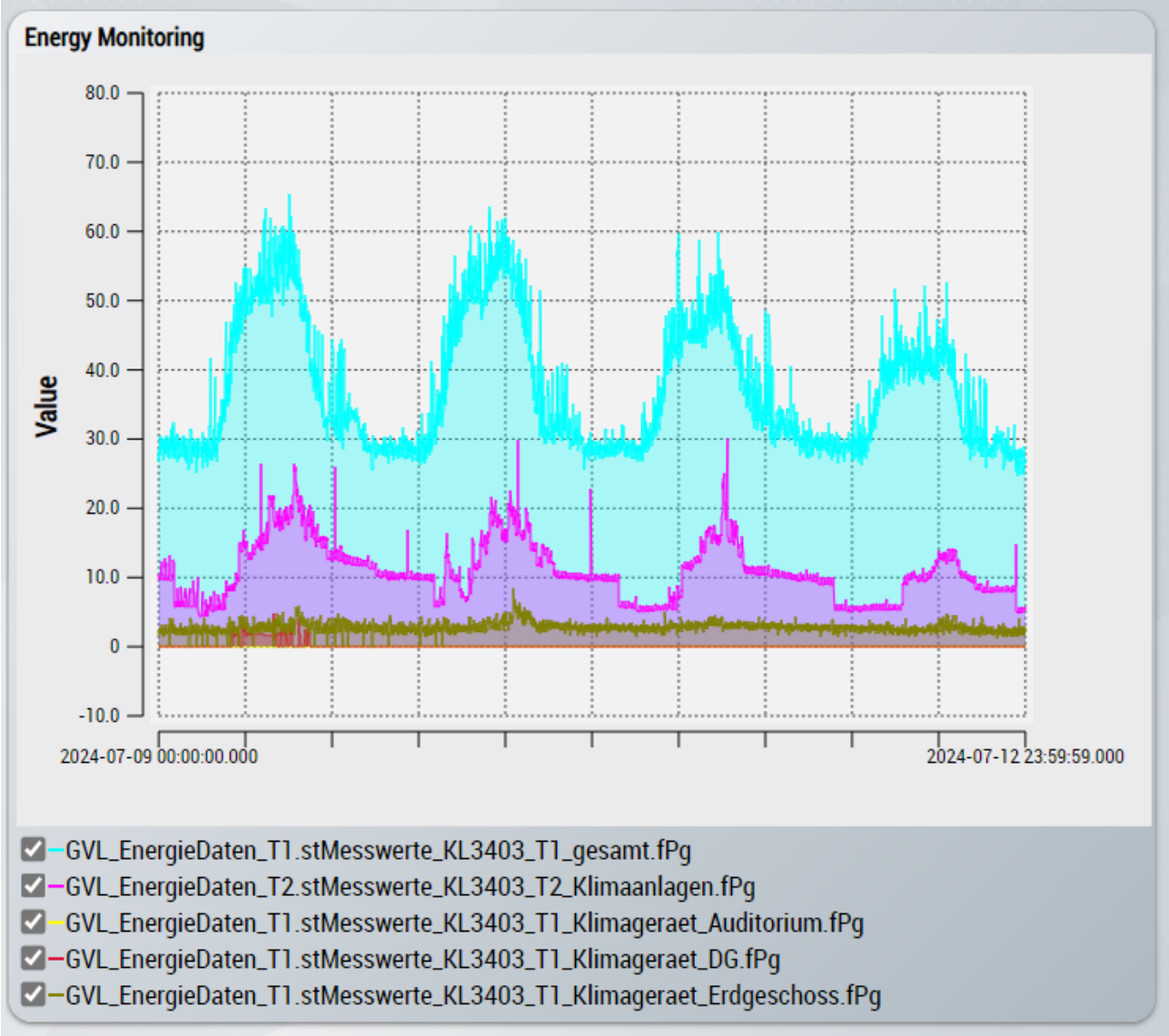
6.8.7.3.1 Diagrammtyp

Den Diagrammtyp können Sie über die Eigenschaften des Diagramms einstellen.



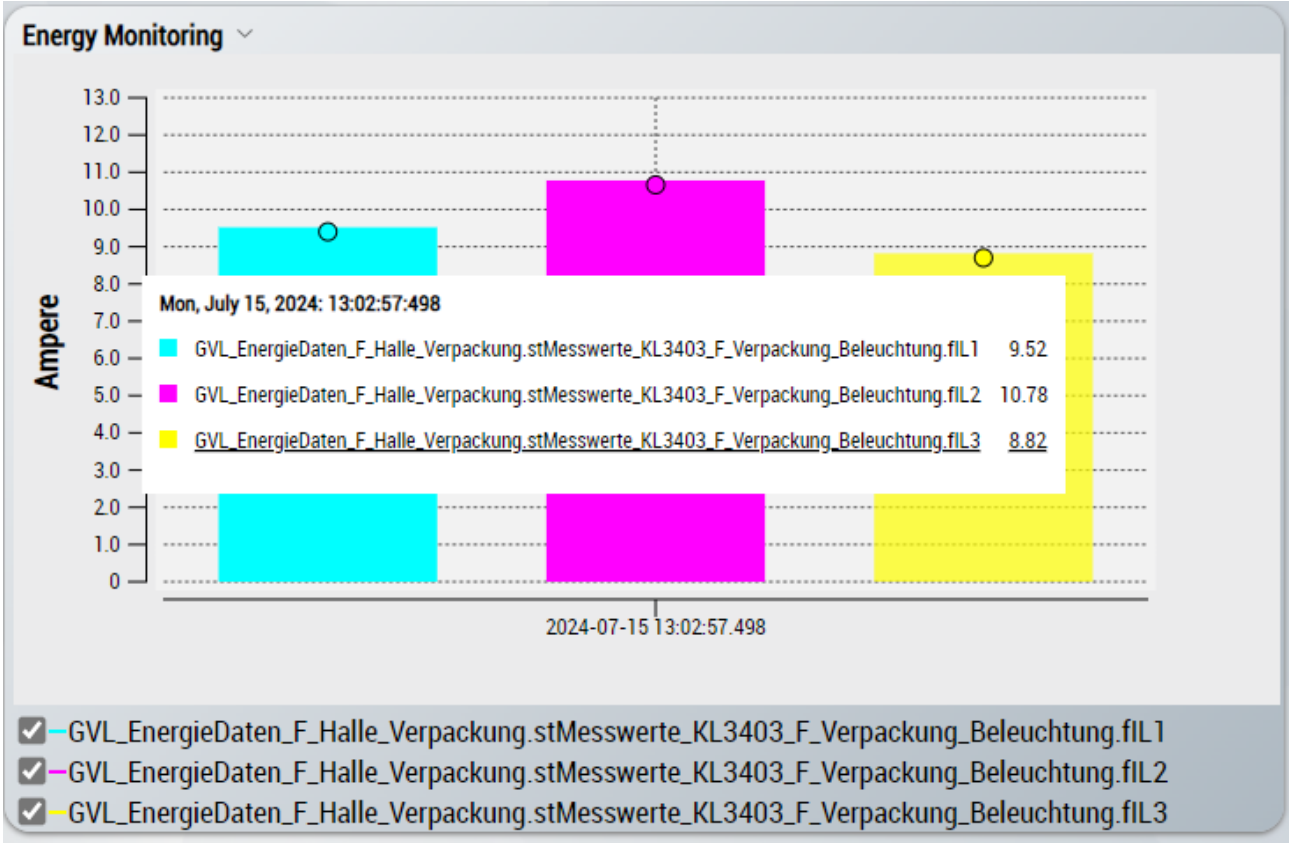
Sie können folgende Diagrammtypen verwenden:

YT-Chart



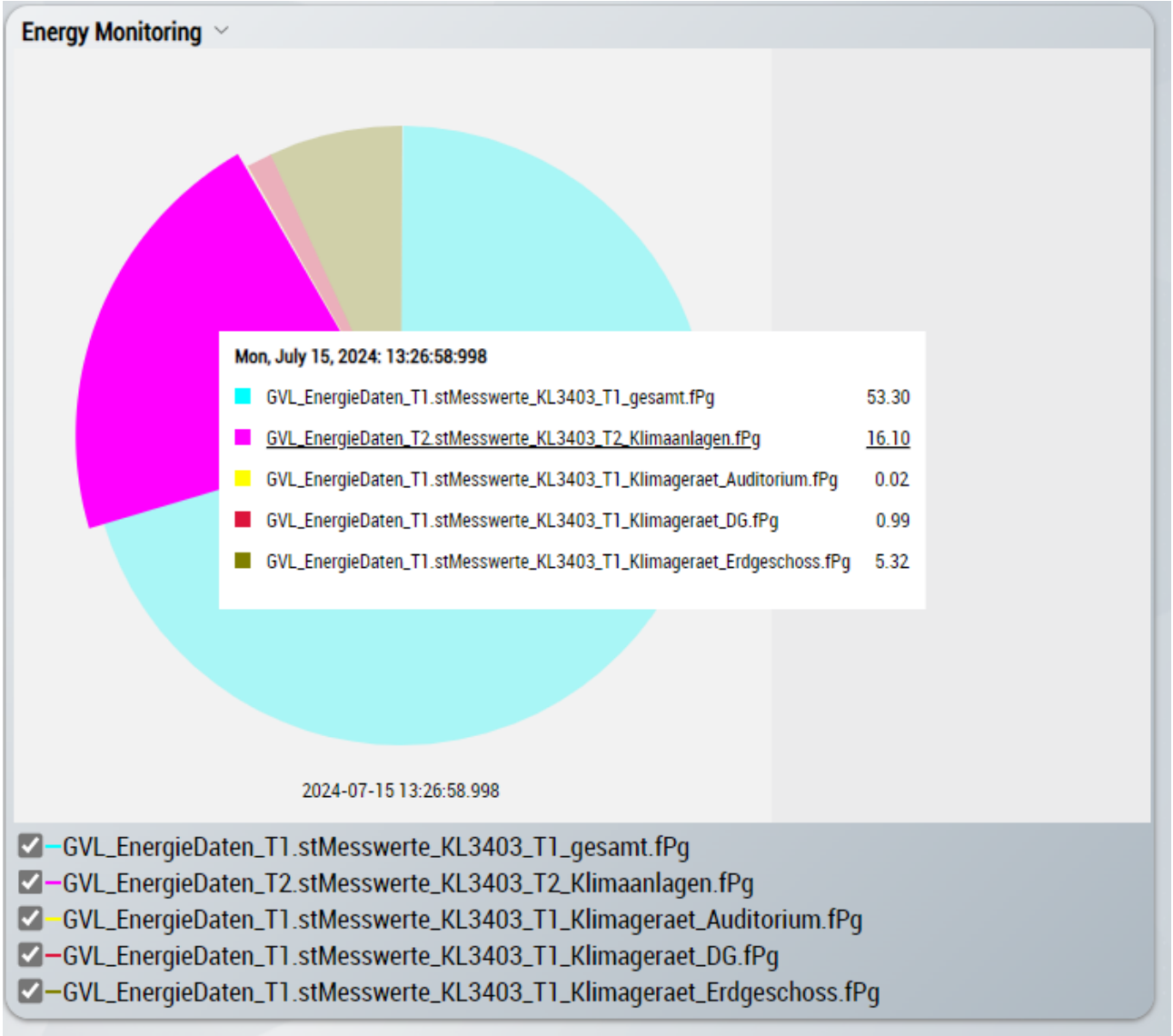
Darstellung der Zeit auf der X-Achse und der entsprechenden Werte auf der Y-Achse.

Bar-Chart



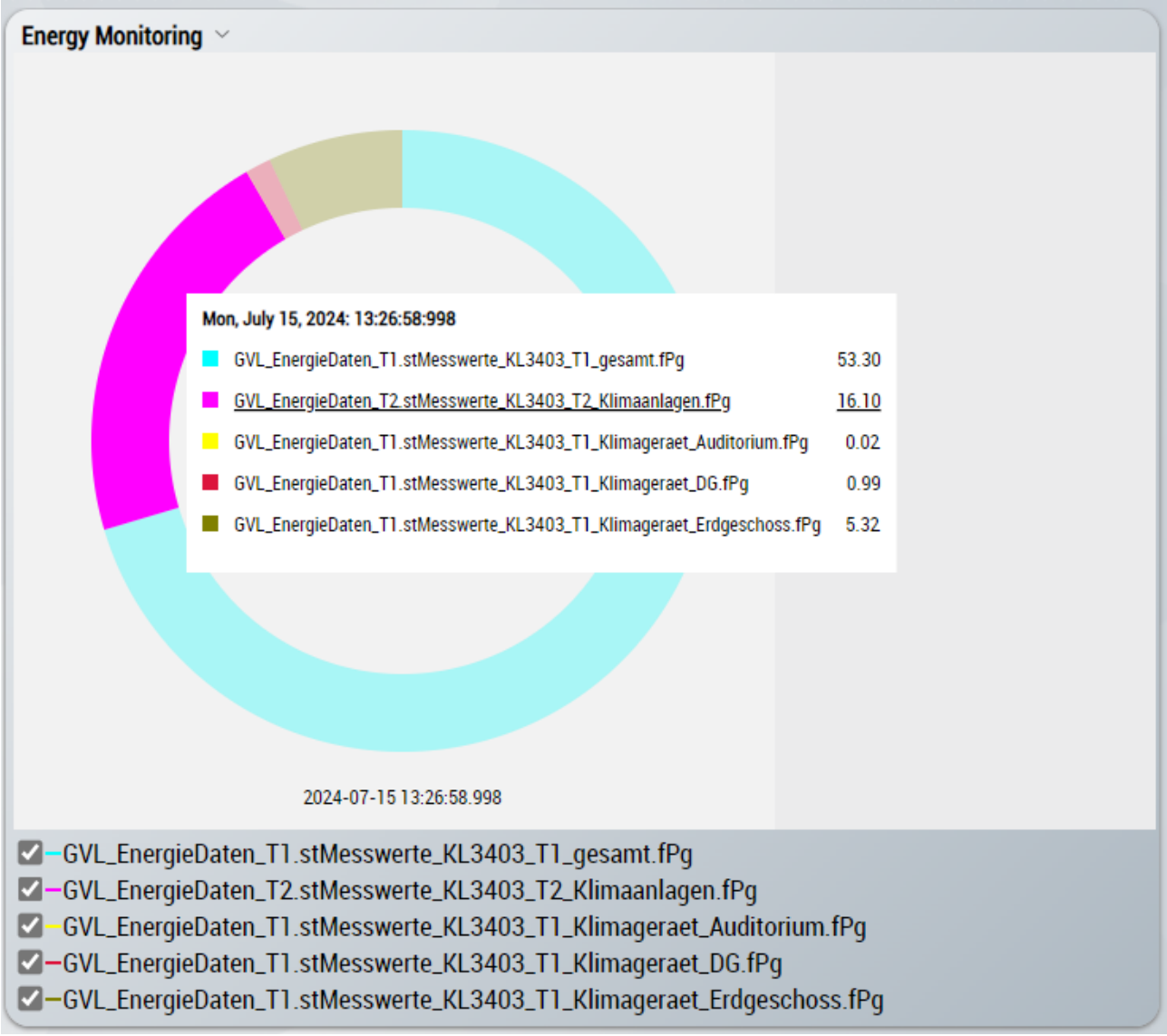
Pro ausgewähltem Zeitbereich und pro Variable wird ein Balken dargestellt. Der Balken stellt jeweils den letzten Wert des Zeitbereichs dar.

Pie-Chart



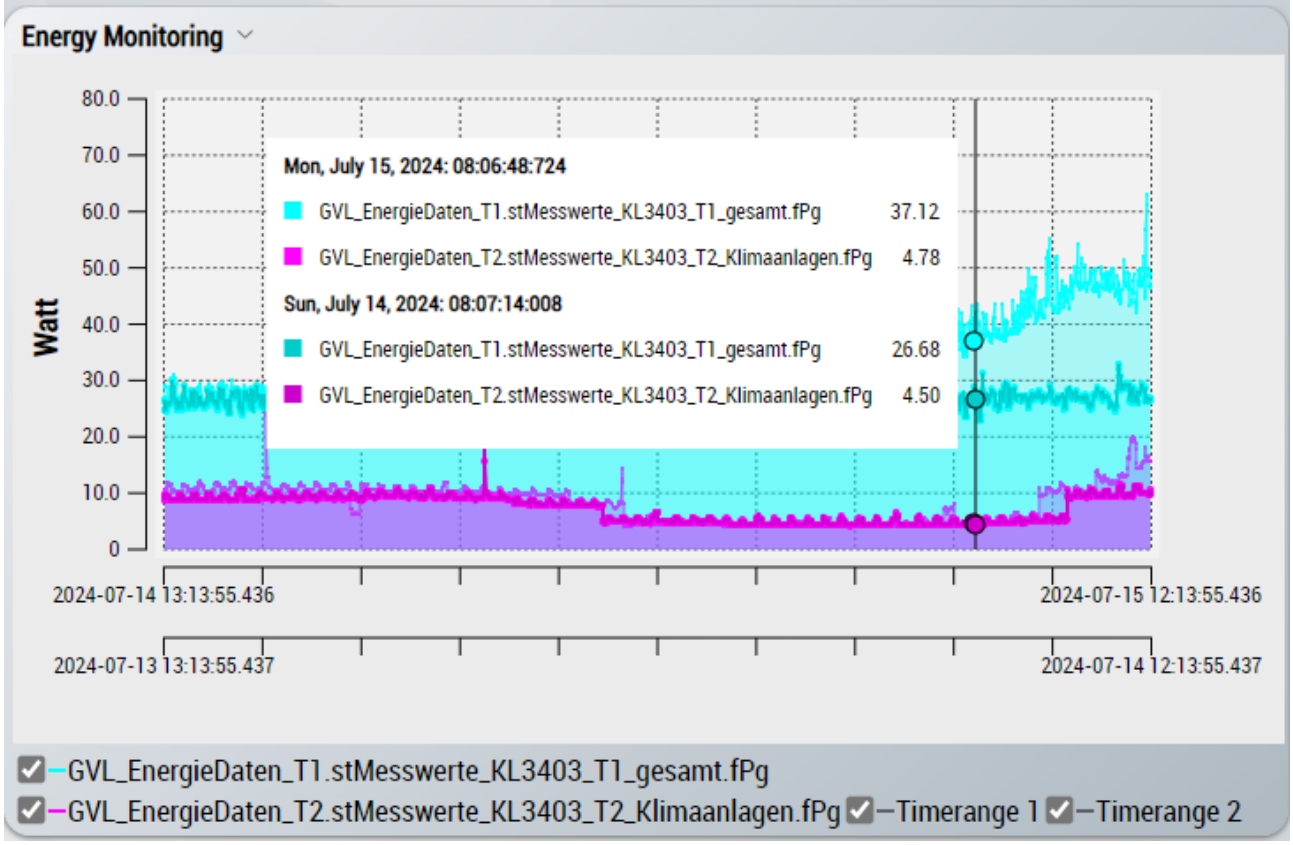
Pro ausgewähltem Zeitbereich wird ein Kuchen dargestellt, welcher jeweils das letzte Sample des Zeitbereichs darstellt. Negative Variablenwerte werden nicht dargestellt.

Ring-Chart



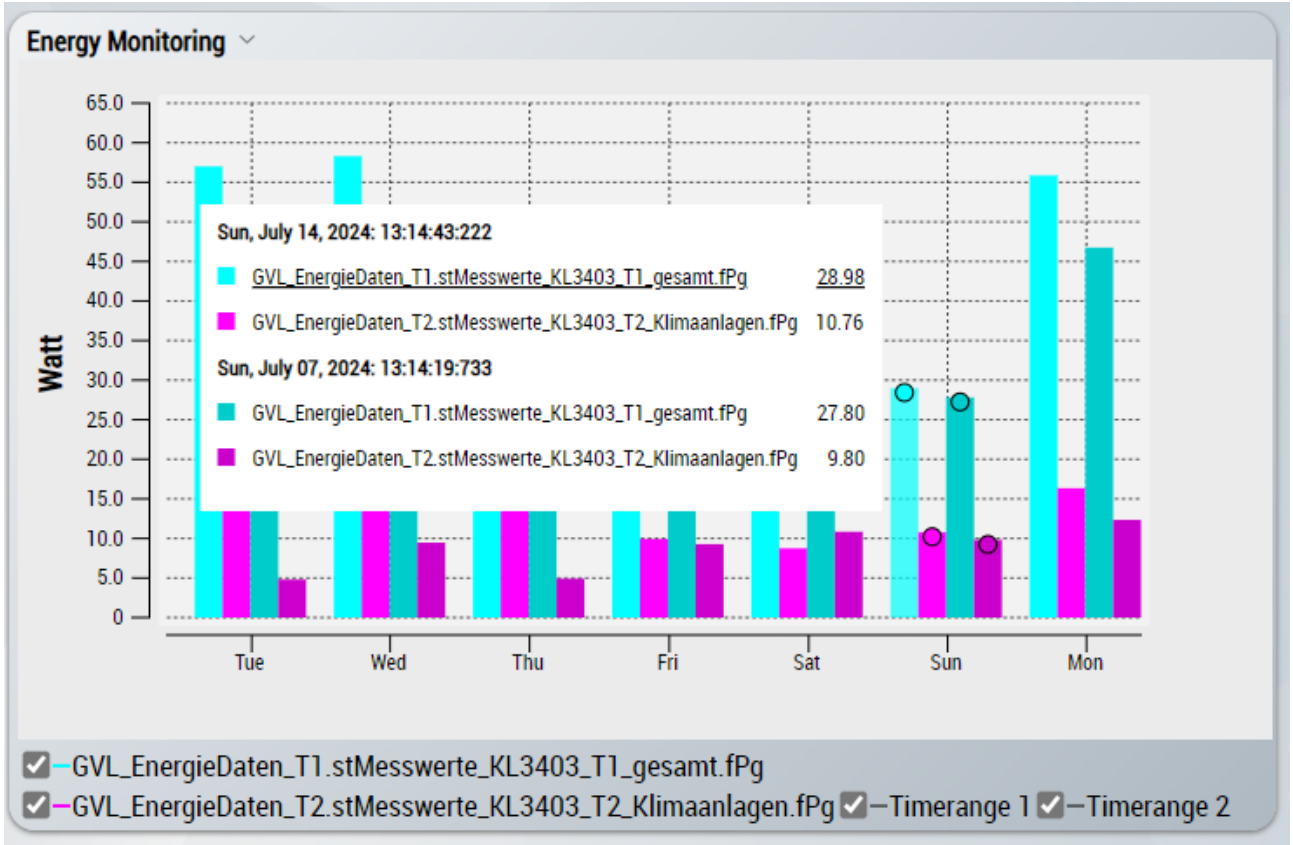
Pro ausgewähltem Zeitbereich wird ein Ring dargestellt, welcher jeweils das letzte Sample des Zeitbereichs darstellt. Negative Variablenwerte werden nicht dargestellt.

Compare-YT-Chart



Geeignet zum Vergleichen mehrerer Zeitreihen. Hierbei werden die Daten über das ausgewählte Vergleichsintervall miteinander vergleichen. Für den Vergleich stehen mehrere Vergleichsmodi zur Verfügung. Auf der X-Achse wird die Vergleichszeitspanne dargestellt.

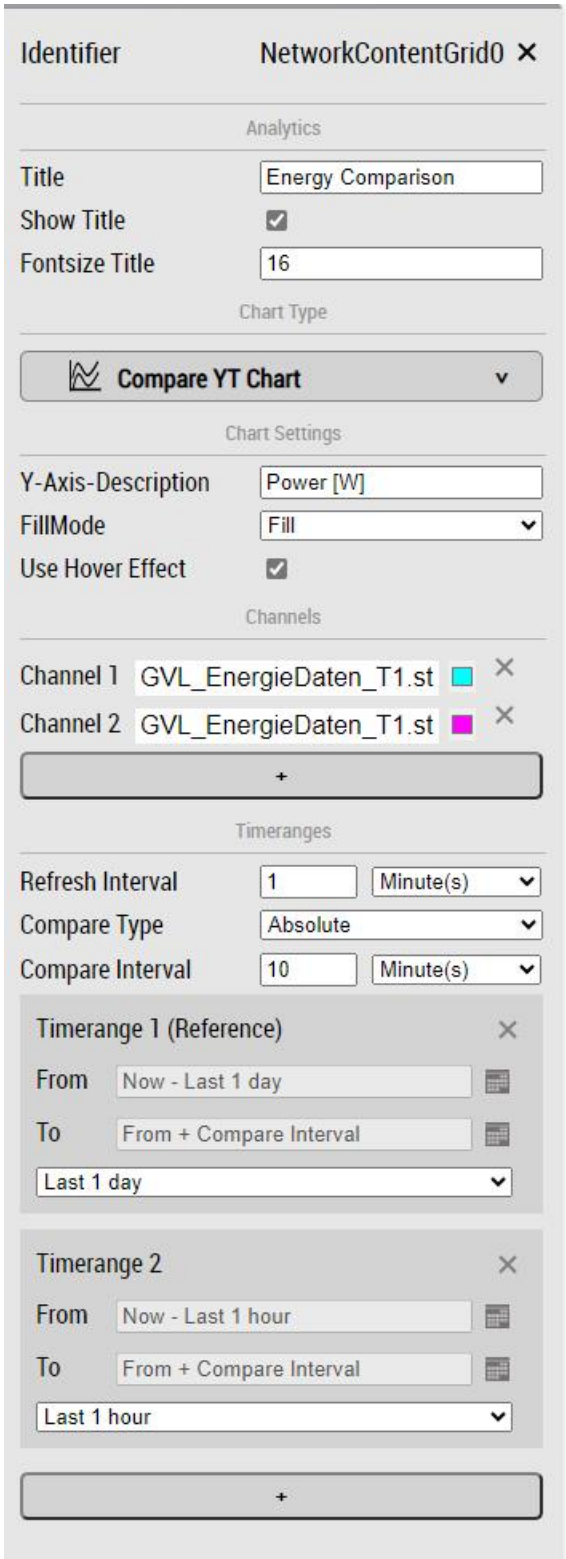
Time-Series-Bar-Chart



Geeignet zum Vergleichen mehrerer Zeitreihen. Hierbei werden die Daten über das ausgewählte Vergleichsintervall miteinander vergleichen. Für den Vergleich stehen mehrere Vergleichsmodi zur Verfügung. Pro ausgewählter Zeitspanne und pro Variable werden basierend auf der Vergleichszeitspanne Balkendiagramme erzeugt.

6.8.7.3.2 Allgemeine Einstellungen

Im Bereich **Chart Settings** können Sie allgemeine Diagrammeinstellungen vornehmen.

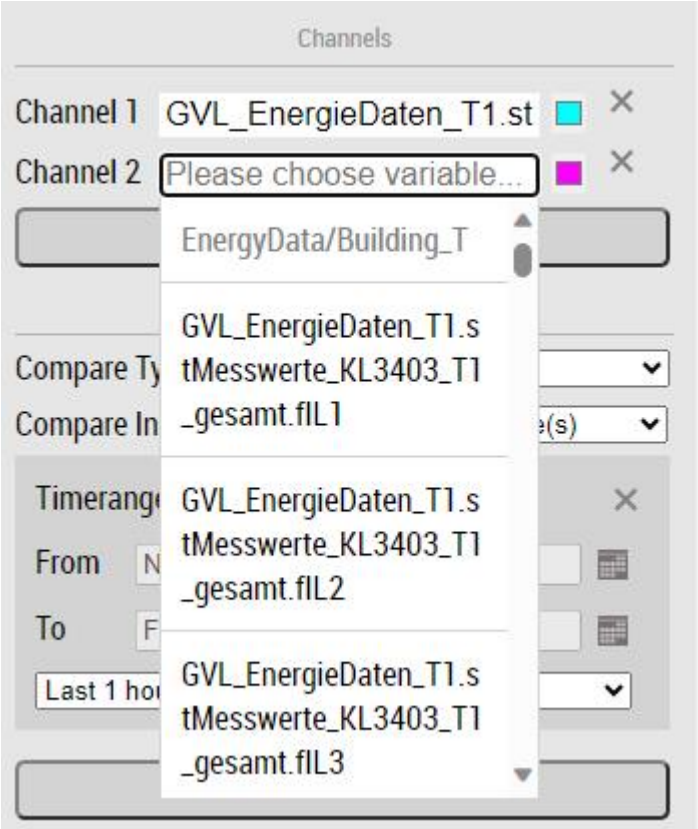


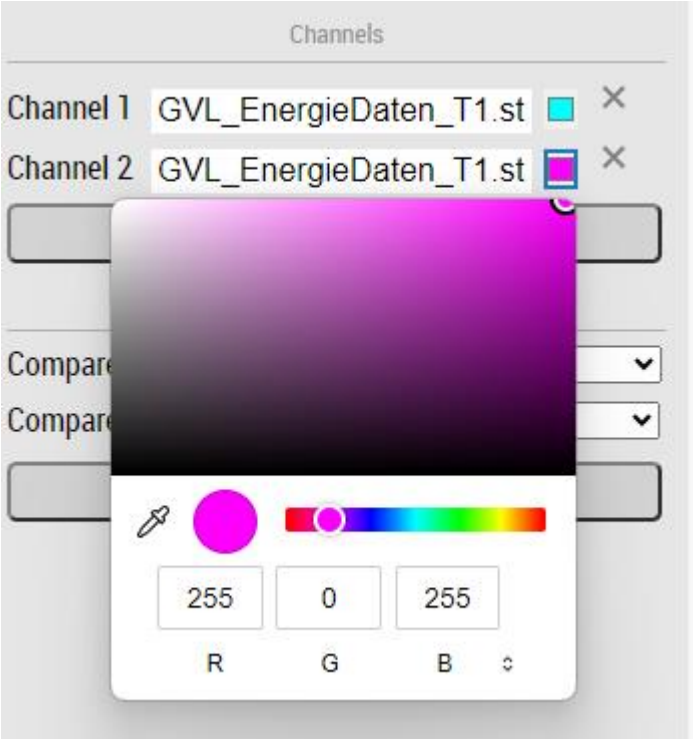
Folgende Optionen können hier gesetzt werden:

Property	Beschreibung
Y-Axis-Description	Beschriftung der Y-Achse
FillMode	Auswahl, ob die Graphen ausgefüllt sein sollen.
Use Hover Effekt	Wenn diese Option ausgewählt ist und sich der Mauszeiger im Diagramm befindet, werden in einem Fenster die genauen Zeitpunkte und Variablenwerte dargestellt.

6.8.7.3.3 Hinzufügen von Variablen

Im Bereich **Channels** können Sie die darzustellenden Variablen auswählen. Fügen Sie über das Plus-Icon einen neuen Kanal für das Diagramm hinzu. Hinterlegen Sie über ein Dropdown-Fenster eine Variable für den Kanal. Vergeben Sie über das Farb-Icon eine individuelle Farbe für den Kanal. Entfernen Sie den Kanal über das X-Icon.





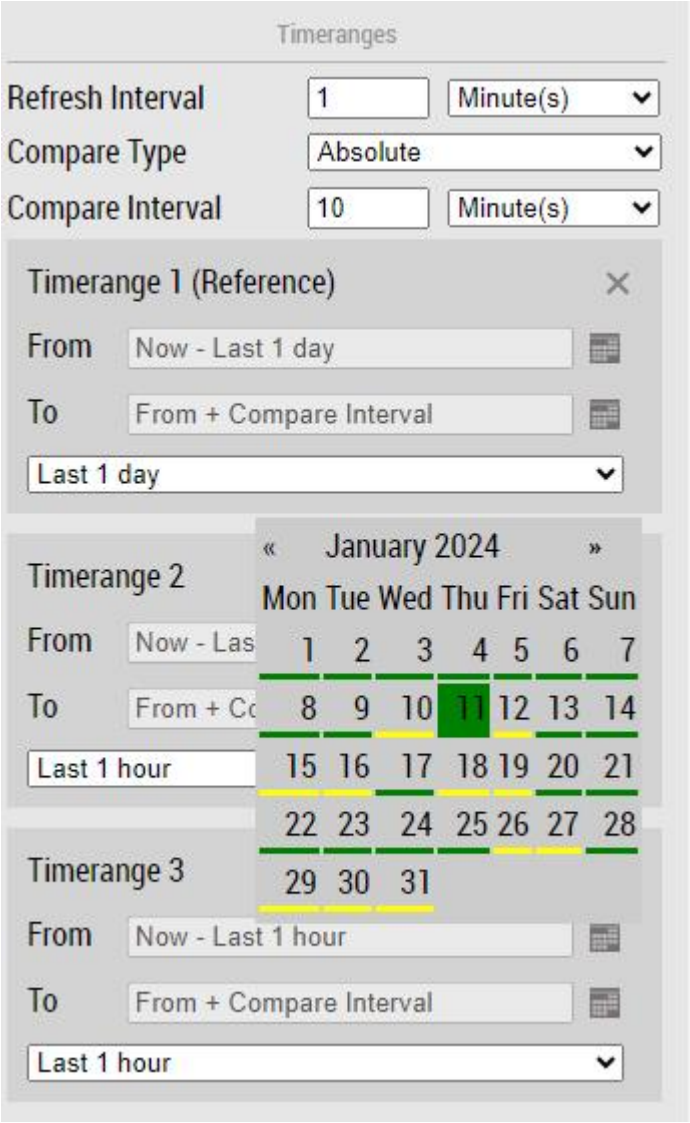
6.8.7.3.4 Hinzufügen von Zeitbereichen

Im Bereich „Timeranges“ können die darzustellenden Zeitbereichen ausgewählt werden. Über das Plus-Icon kann ein neuer Zeitbereich für das Diagramm hinzugefügt werden.

Absolute Zeitbereiche

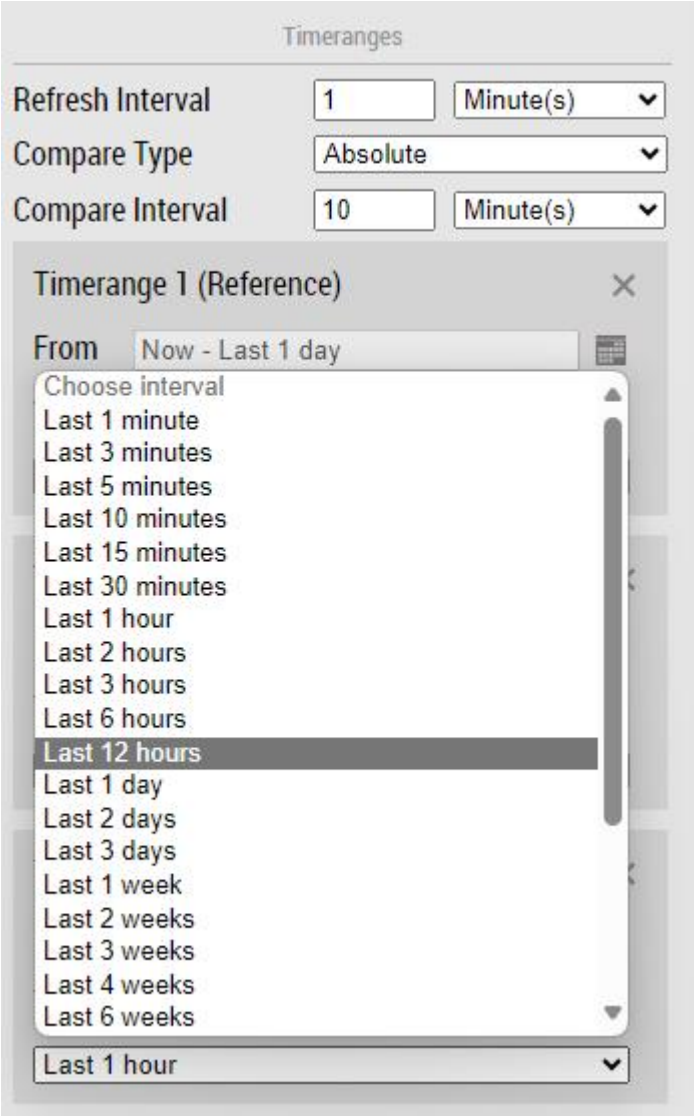
Bei einem Zeitbereich können über die Parameter „From“ und „To“ der Start und Endzeitpunkt ausgewählt werden. Die Datumsauswahl kann über den Kalender (Öffnet sich durch einen Klick auf das Kalender-Icon) vorgenommen werden. Im Kalender sind die Tage farbig markiert:

- Grün: Tag wurde vollständig aufgezeichnet.
- Gelb: Tag wurde nur teilweise aufgezeichnet.
- Grau: Keine Daten für diesen Tag vorhanden.



Relative Zeitbereiche

Alternativ zu absoluten Zeitbereichen können auch relative Zeitbereiche ausgewählt werden. Die Auswahl eines relativen Zeitbereichs kann über das Dropdown-Fenster vorgenommen werden. Sobald mindestens ein relativer Zeitbereich ausgewählt wurde, kann über das Property „Refresh Interval“ festgelegt werden, in welchem Zeitabstand sich das Diagramm aktualisiert.



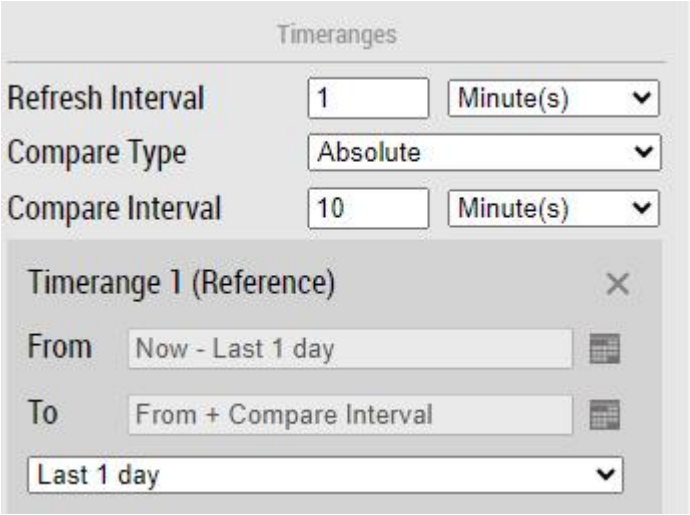
Automatisches Setzen des Endzeitpunkts

Bei den Diagrammtypen Compare-YT-Chart und Time-Series-Bar-Chart wird der Endzeitpunkt eines Zeitbereichs basierend auf dem konfigurierten Vergleichsintervall gesetzt. Eine manuelle Auswahl des Endzeitpunkts ist in diesen Diagrammtypen nicht möglich.

6.8.7.4 Vergleichen von Daten

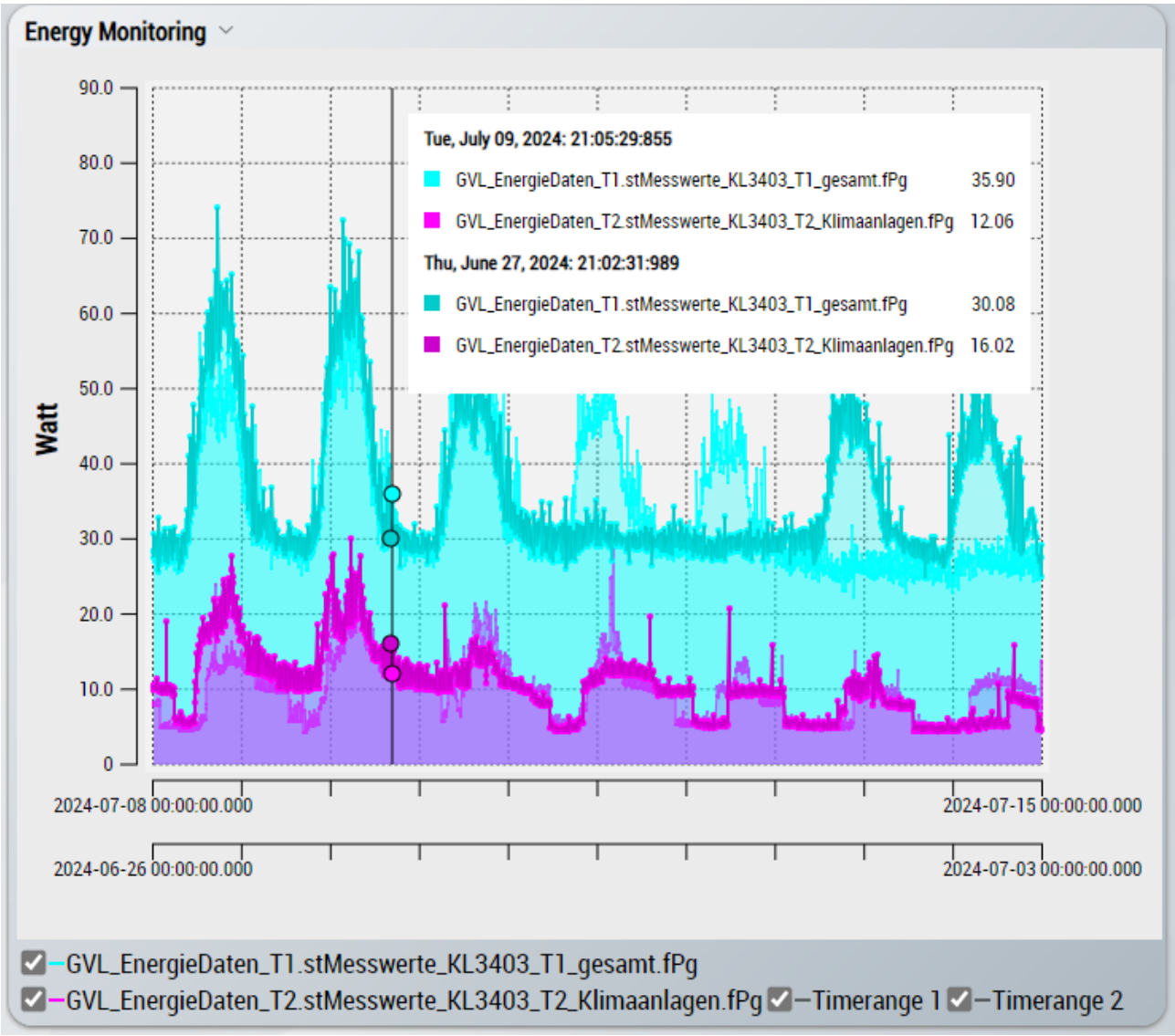
Mittels der Diagrammtypen Compare-YT-Chart und TimeSeries-Bar-Chart können Daten aus verschiedenen Zeitbereichen miteinander verglichen werden. Hierfür können in den Einstellungen bis zu drei Zeitbereiche hinzugefügt werden. Der erste Zeitbereich dient hierbei als Referenz für die Darstellung.

Das Vergleichen der Daten geschieht über eine definierte Zeitspanne (z. B: 3 Stunden, 1 Woche, …) Die Länge der Vergleichszeitspanne kann im Bereich „Timeranges“ konfiguriert werden, wenn als Diagrammtyp Compare-YT-Chart oder TimeSeries-Bar-Chart ausgewählt ist.



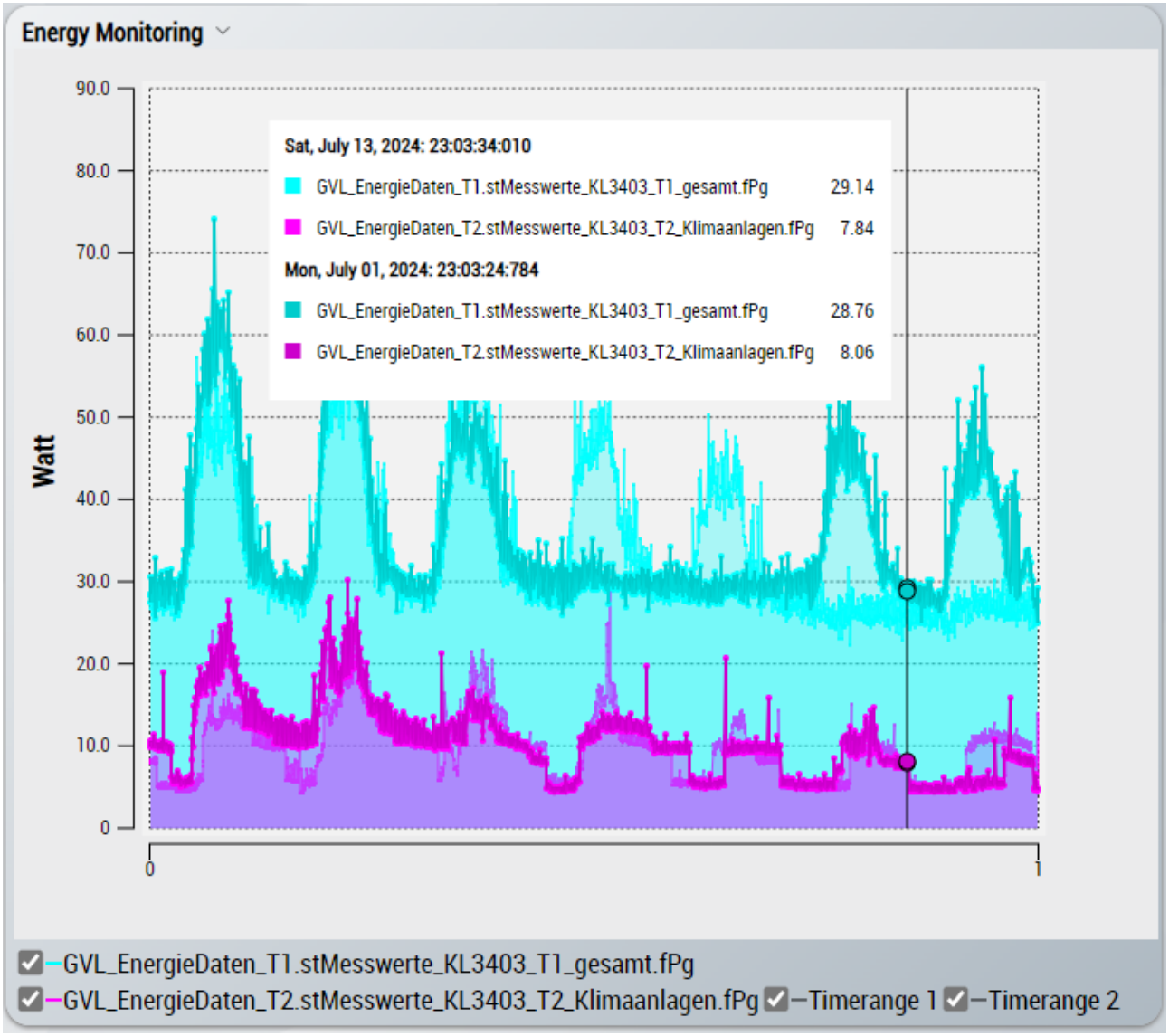
Zusätzlich zu der Zeitspanne kann ein Vergleichsmodus festgelegt werden. Darüber lässt sich einstellen, ob die Daten einfach übereinander dargestellt werden sollen, oder ob sie zueinander synchronisiert dargestellt werden sollen. Dies entspricht einer Verschiebung auf der Zeitachse im Diagramm. Die verfügbaren Vergleichsmodi sind:

Absolut



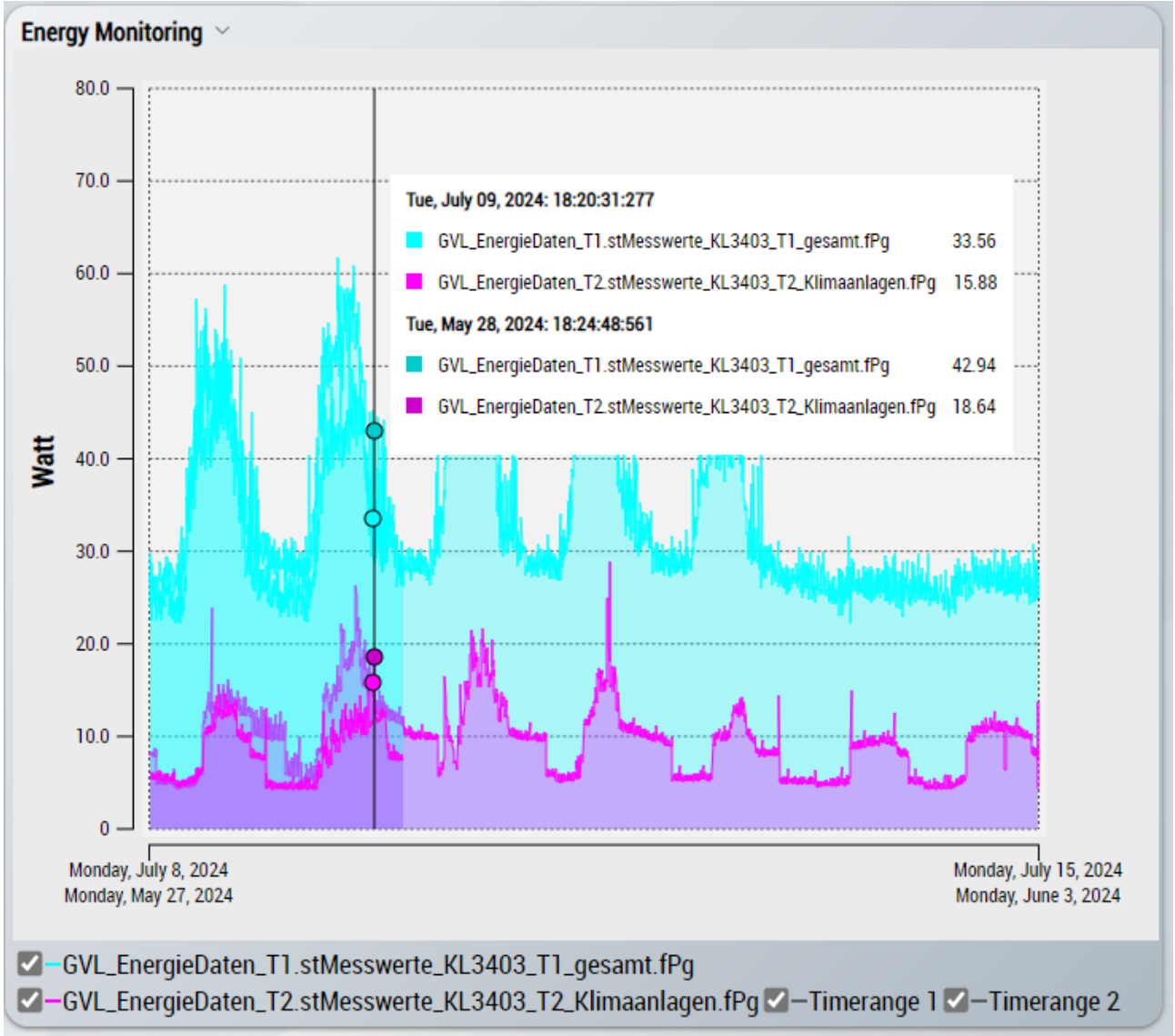
Bei diesem Modus werden alle ausgewählten Zeitbereiche übereinander visualisiert. Auf den X-Achsen wird die absolute Zeit dargestellt. Es findet keine Verschiebung auf der Zeitachse statt. Der Ursprung ist hierbei immer der Startzeitpunkt einer Zeitspanne.

Relativ



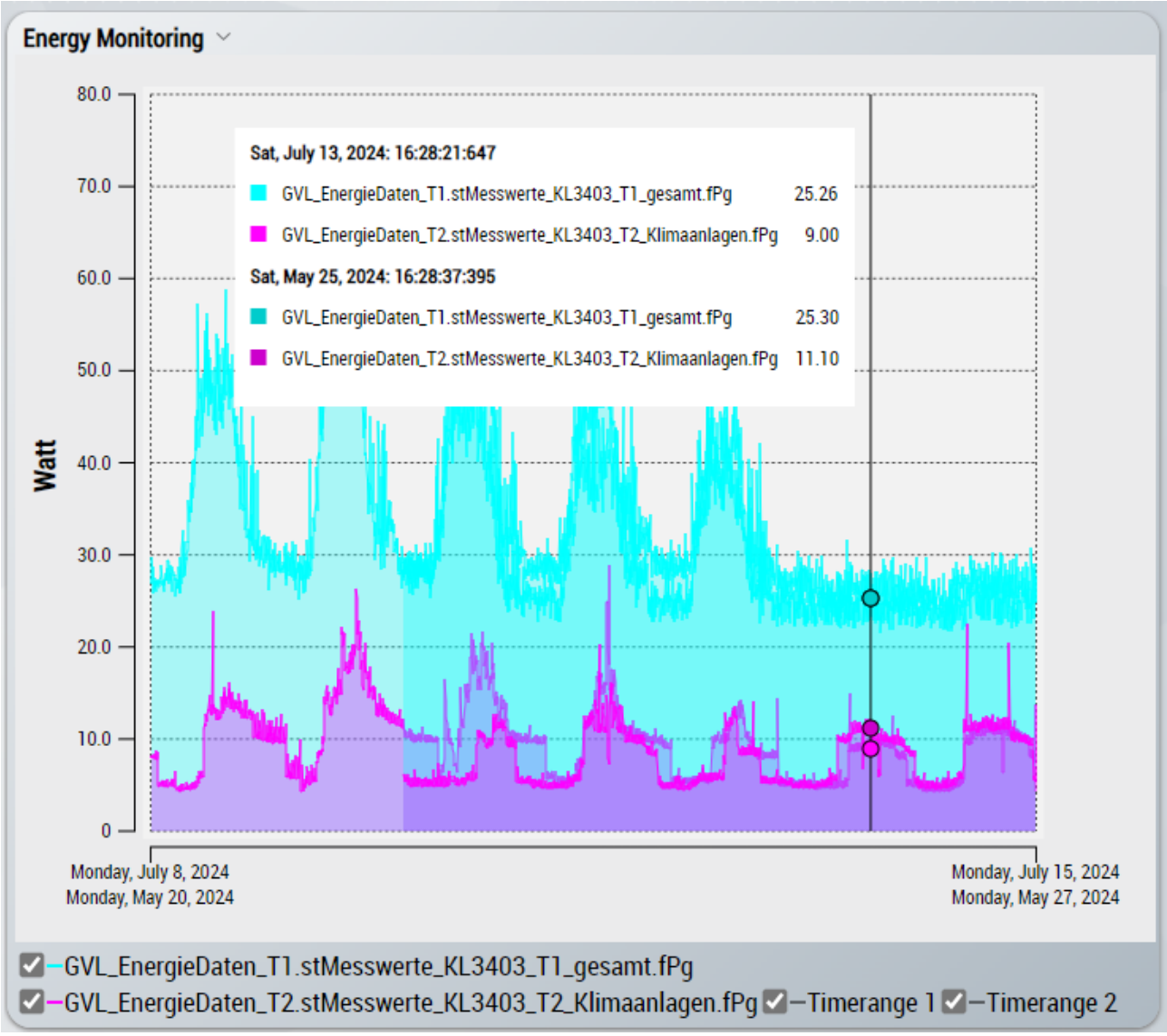
Bei diesem Modus werden alle ausgewählten Zeitbereiche übereinander visualisiert. Auf den X-Achsen wird die relativ vergangene Zeitspanne dargestellt. Es findet keine Verschiebung auf der Zeitachse statt.

Match Start



In diesem Modus werden die Daten zeitlich passend übereinandergelegt. Der Startzeitpunkt des ersten Zeitbereichs dient hierbei als Referenz. Die Daten der weiteren Zeitpunkte werden so verschoben, dass die Daten am Anfang des Diagramms zeitlich passend übereinanderliegen. Die Grundlage hierfür ist die gewählte Vergleichszeitspanne (siehe Tabelle).

Match End



In diesem Modus werden die Daten zeitlich passend übereinandergelegt. Der Endzeitpunkt des ersten Zeitbereichs dient hierbei als Referenz. Die Daten der weiteren Zeitpunkte werden so verschoben, dass die Daten am Ende des Diagramms zeitlich passend übereinanderliegen. Die Grundlage hierfür ist die gewählte Vergleichszeitspanne (siehe Tabelle).

Vergleichszeitspanne	Vorgehen
Second(s)	Vergleich der Millisekunden der Daten. Sobald die gleiche Millisekunde auftritt, werden die Daten übereinandergelegt.
Minute(s)	Vergleich der Sekunden der Daten. Sobald die gleiche Sekunde auftritt, werden die Daten übereinandergelegt.
Hour(s)	Vergleich der Minuten der Daten. Sobald die gleiche Minute auftritt, werden die Daten übereinandergelegt.
Day(s)	Vergleich der Stunde der Daten. Sobald die gleiche Stunde auftritt, werden die Daten übereinandergelegt.
Week(s)	Vergleich des Wochentages der Daten. Sobald der gleiche Wochentag auftritt, werden die Daten übereinandergelegt.
Month(s)	Vergleich des Tages der Daten. Sobald der gleiche Tag (Datum) auftritt, werden die Daten übereinandergelegt. Ein Monat entspricht immer 31 Tagen.
Year(s)	Vergleich des Monats der Daten. Sobald der gleiche Monat auftritt, werden die Daten übereinandergelegt.

Verschiebung der Daten

Da nicht sichergestellt ist, dass die Daten mit der gleichen Zykluszeit aufgezeichnet werden oder die Daten mit der gleichen Abtastrate abgefragt werden, ist es möglich, dass die Werte nicht immer exakt übereinanderliegen.

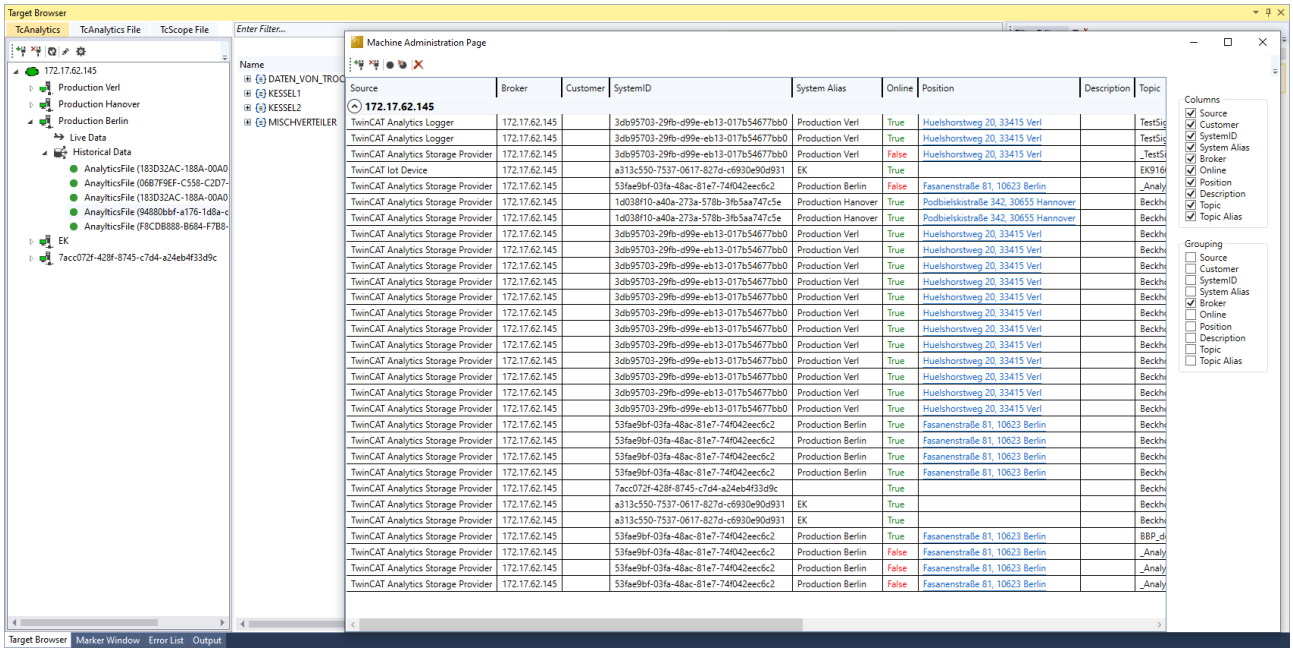
6.8.8 Umschaltung mehrerer Maschinen im HMI Dashboard

In TwinCAT Analytics können Sie verschiedene Datenströme von mehreren Maschinen verwenden, die Sie in einer Analyse umschalten können. Dies geht sowohl in der Analytics Workbench über die Virtual Input Source als auch in der fertig generierten SPS und HMI. Dabei können sowohl live als auch historische Daten verwendet werden. Für jeden Datenstrom können Sie entweder im Analytics Logger oder in der Machine Administration [▶ 29] eine kurze Beschreibung und den Standort hinzufügen.

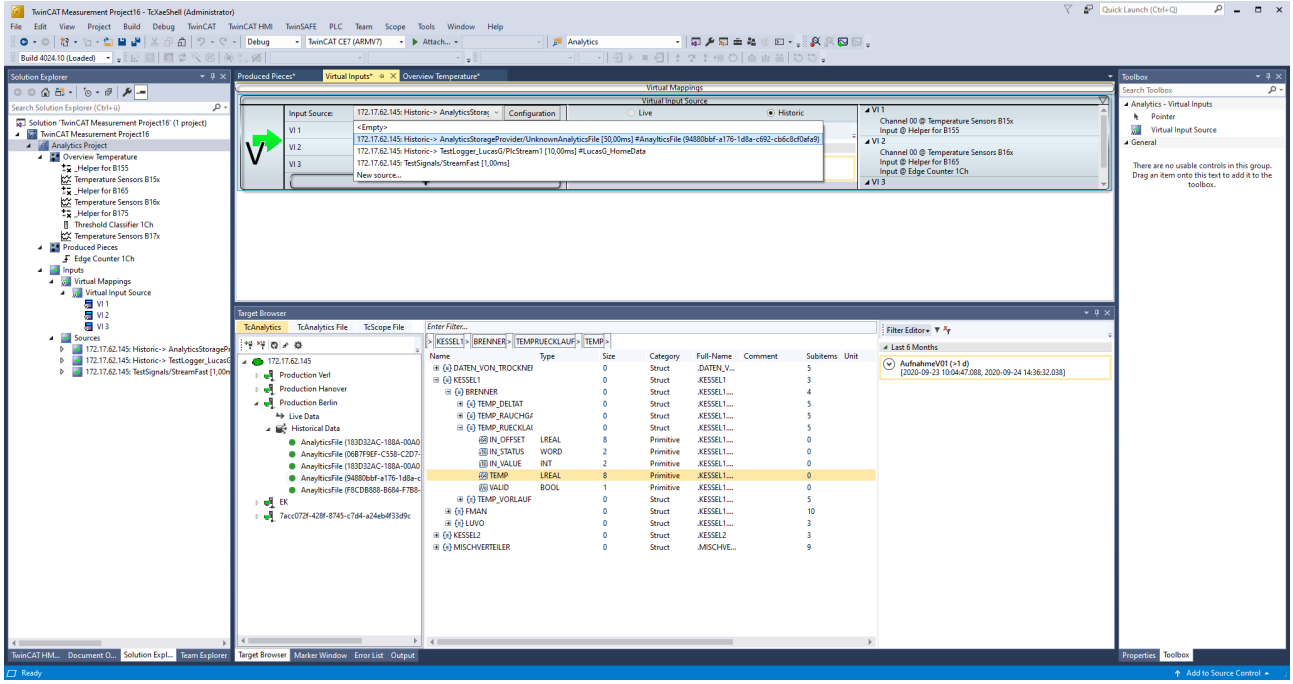
Seite Machine Administration

Falls nicht bereits in der Übersicht geschehen, legen Sie die Metadaten Ihrer Maschine auf der Seite Analyse der Daten [▶ 29] fest. Öffnen Sie den TwinCAT Target Browser (**TwinCAT > Target Browser > Target Browser**) und klicken Sie auf das Zahnrad-Symbol. Nun können Sie den Standort Ihrer Maschine, eine kurze Beschreibung und den Namen Ihrer Maschine eingeben. Beachten Sie, dass bereits vorhandene Daten in einem Analytics Projekt diese Metainformationen nicht bekommen, da die Informationen erst beim Drag & Drop vom Target Browser übertragen werden. Sie können in diesem Fall den Datenstrom löschen und neu anlegen.

Für jeden Datenstrom, den Sie in Ihrer Analytics-Konfiguration verwenden, erzeugt das Runtime-Deployment [▶ 365] einen Karteneintrag. Diese Karteneinträge werden als Eingangsvariable für die allgemeine Karte auf der Startseite Ihres HMI Dashboards verwendet.

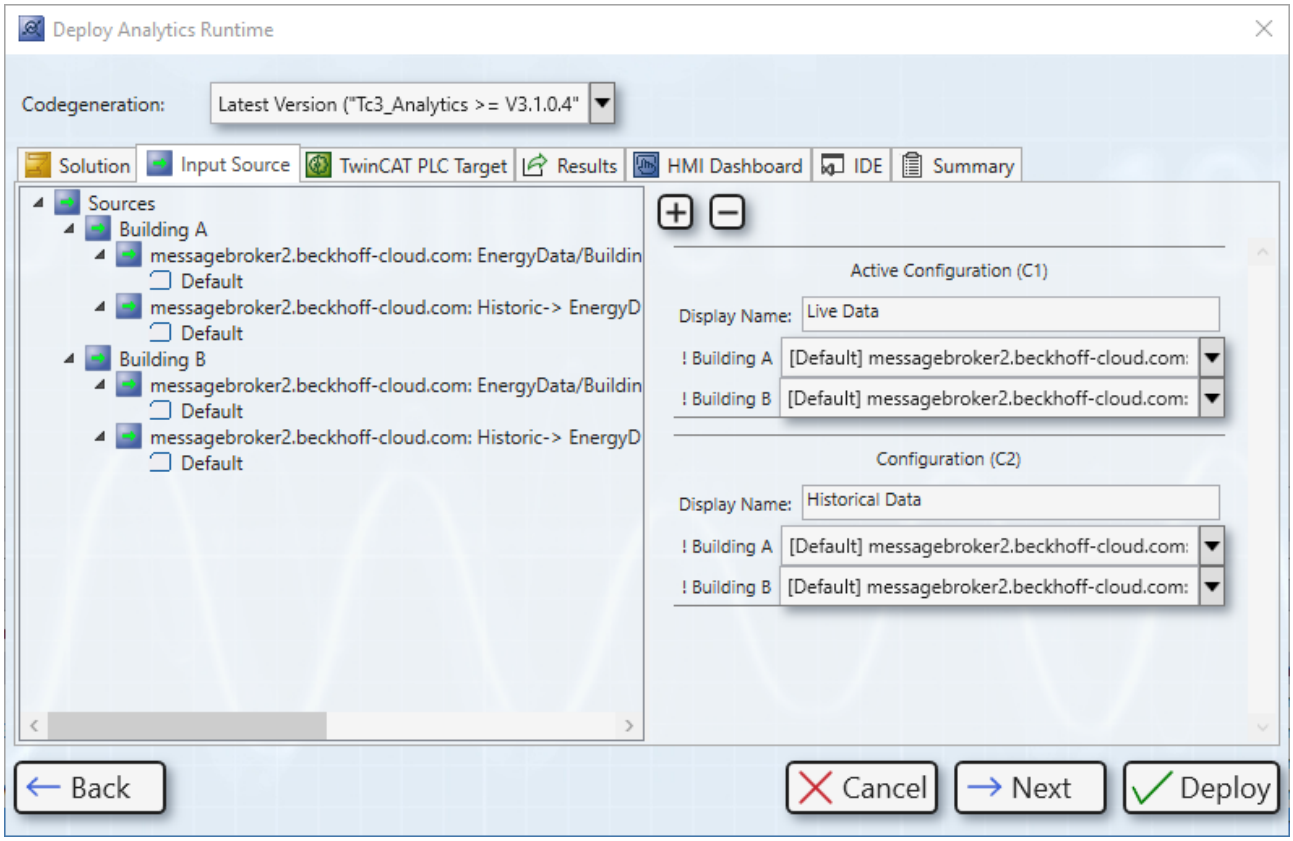


In der Analyse der Daten [▶ 29] werden die Datenströme unter **Sources** gelistet. Diese entstehen, indem Sie aus dem TwinCAT Target Browser von verschiedenen Quellen Daten verwenden. In der Virtual Input Source werden diese Quellen aufgelistet, eine Umschaltung zwischen den Quellen ist möglich. Prüfen Sie, ob alle einzelnen Inputs verknüpft sind und keiner auf **Empty** steht, klicken Sie dazu einmal auf jede Source (jeweils eine Umschaltung in der Workbench). Anschließend können Sie eine HMI mit SPS generieren.

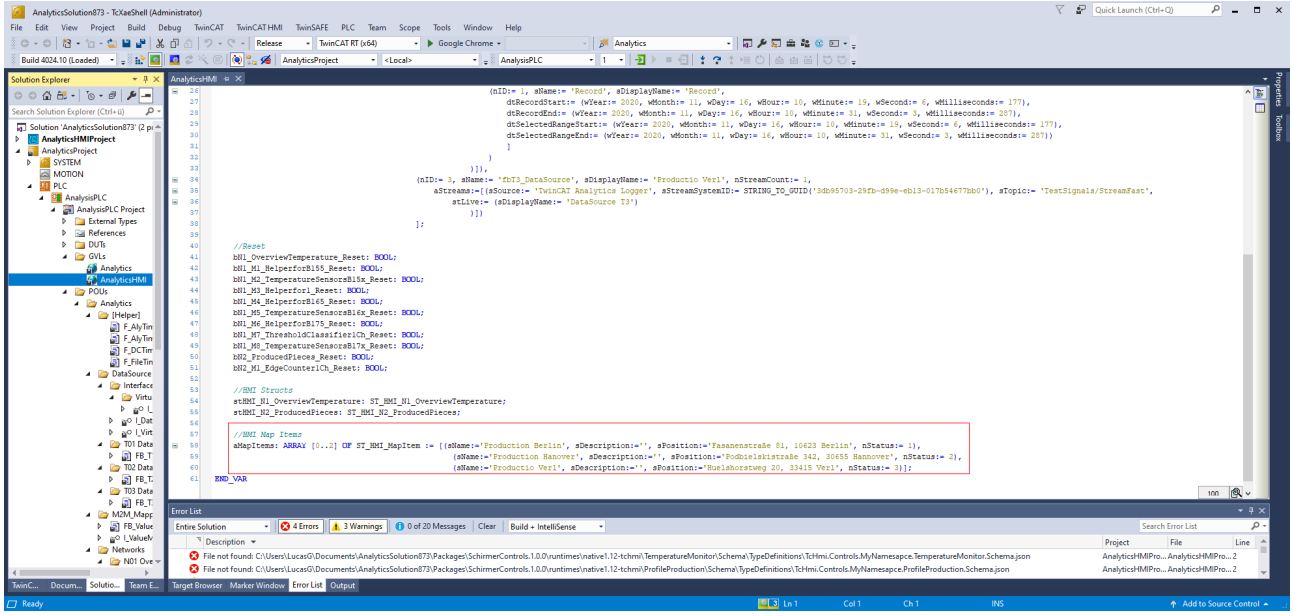


Deploy Wizard

Im Runtime-Deployment [▶ 365] können unter **Input Source** beliebig viele Konfigurationen erstellt werden, die in dem HMI umschaltbar sind. Damit ist es in der Analyse HMI möglich, live sowie historische Daten von ggf. unterschiedlichen Maschinen zu analysieren und umzuschalten. Jede der aufgelisteten Konfigurationen können parallel analysiert werden. Eine Konfiguration besitzt immer so viele Virtual Input Sources, wie in der Analytics Workbench konfiguriert sind.

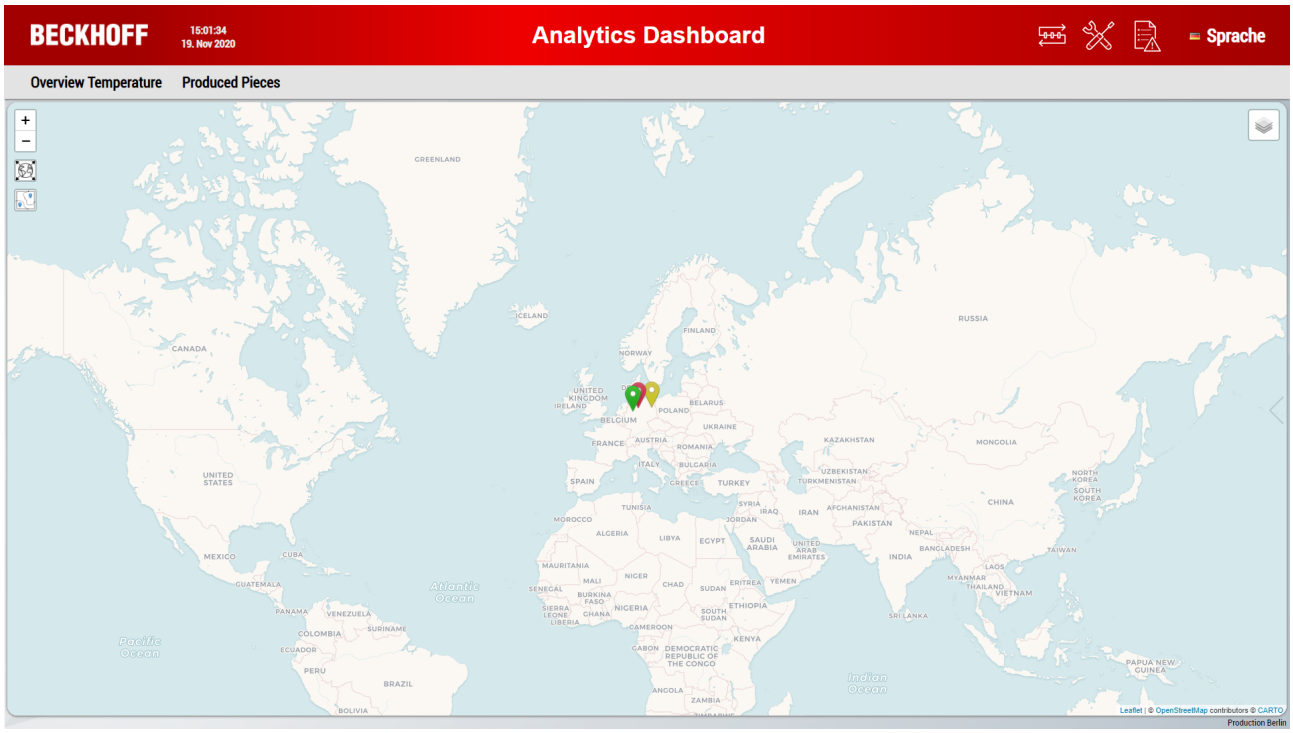


Nach einer erfolgreichen HMI und SPS Generierung sind die Namen, Positionen, und Beschreibungen der Input Sourcen in die globale Variablenliste der SPS eingetragen. Falls etwas nicht stimmig ist oder Sie einen Wert nachträglich ändern möchten, können Sie dies direkt in der SPS tun.

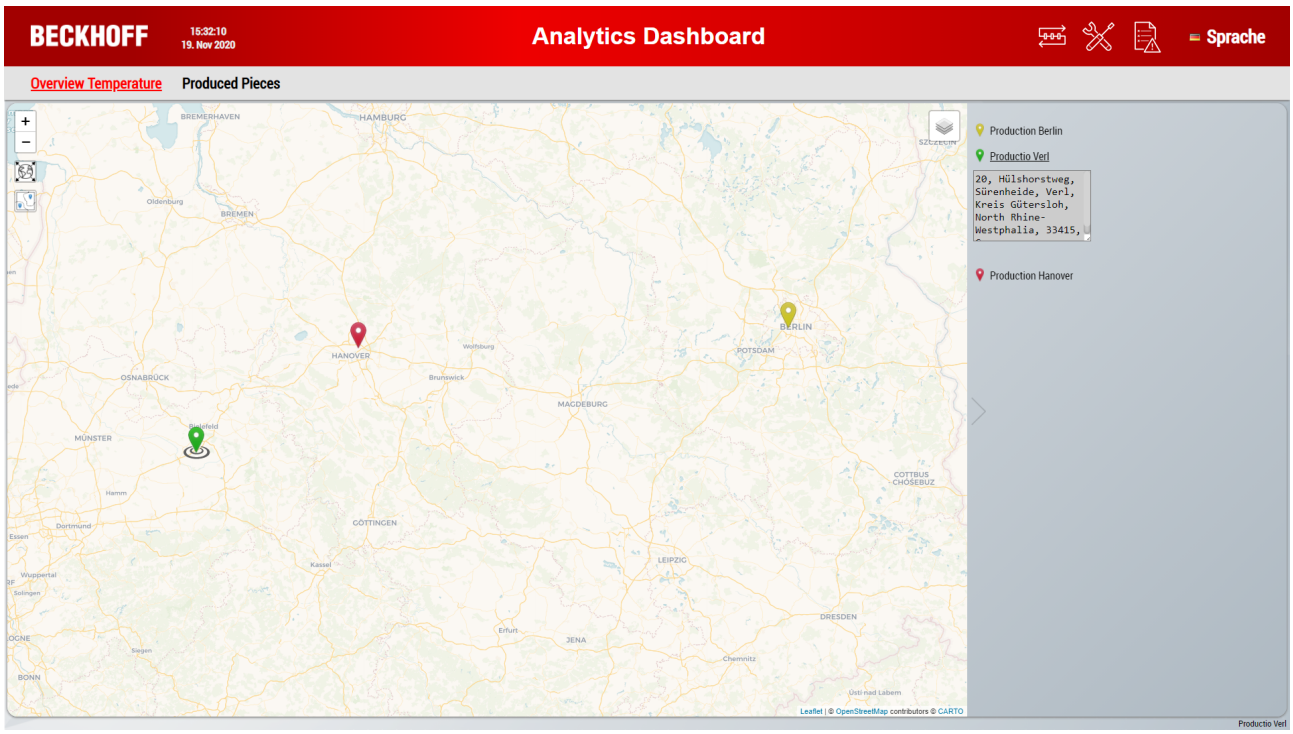


Karte mit Maschinen

Wenn Sie die Erstellung einer Startseite ausgewählt haben, wird das *Map Control* auf der Homepage Ihres HMI Dashboards erzeugt. Die Karte zeigt alle Maschinenstandorte an und listet die Namen der Maschinen (System Alias) in der Legende auf. Die Legende kann über den Pfeil rechts geöffnet und geschlossen werden. Die Farbe des Icons steht für den aktuellen Maschinenstatus: Grün = OK, Gelb = Warnung, Rot = Alarm.

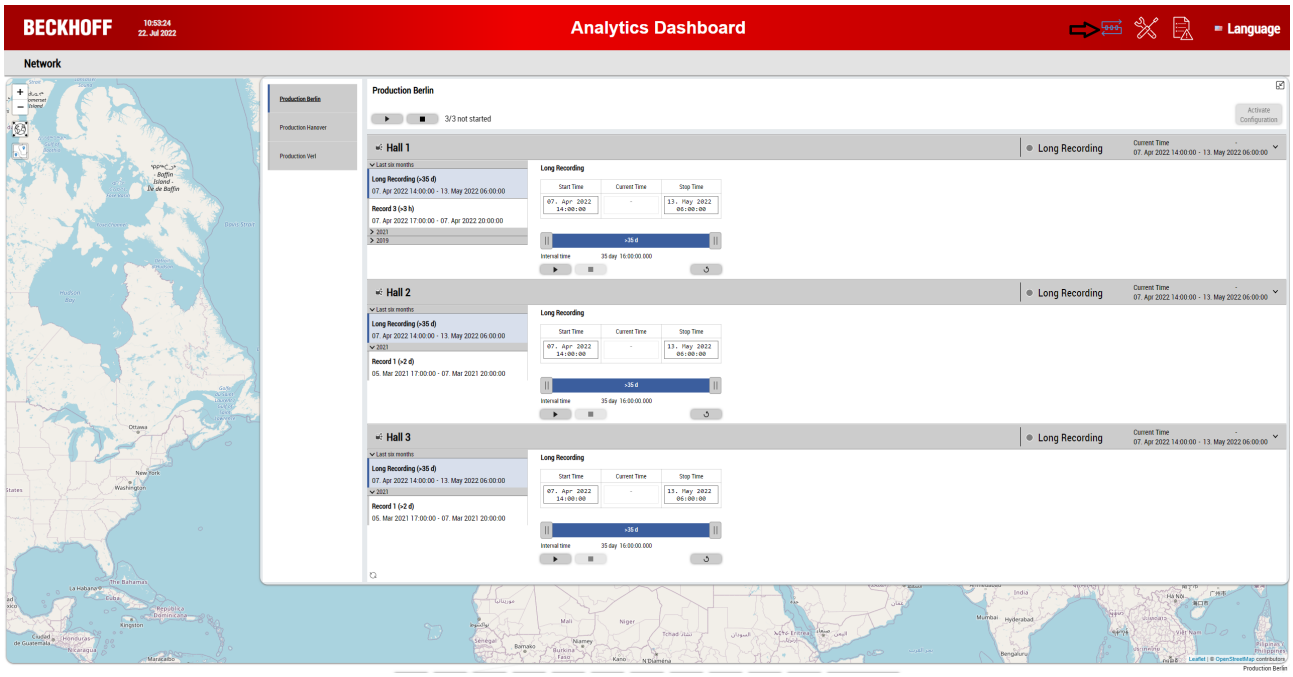


Sie können ein einzelnes Icon durch Doppelklick heranzoomen und mit den Icons auf der linken Seite auf eine Übersicht aller Maschinen zoomen. Zudem besteht die Möglichkeit, auf ein Icon in der Karte zu klicken. Der entsprechende Legendeneintrag wird automatisch hervorgehoben. Dies funktioniert auch andersherum. Wenn Sie auf einen Legendeneintrag klicken, wird das entsprechende Icon automatisch hervorgehoben.



Umschaltung von Maschinendaten

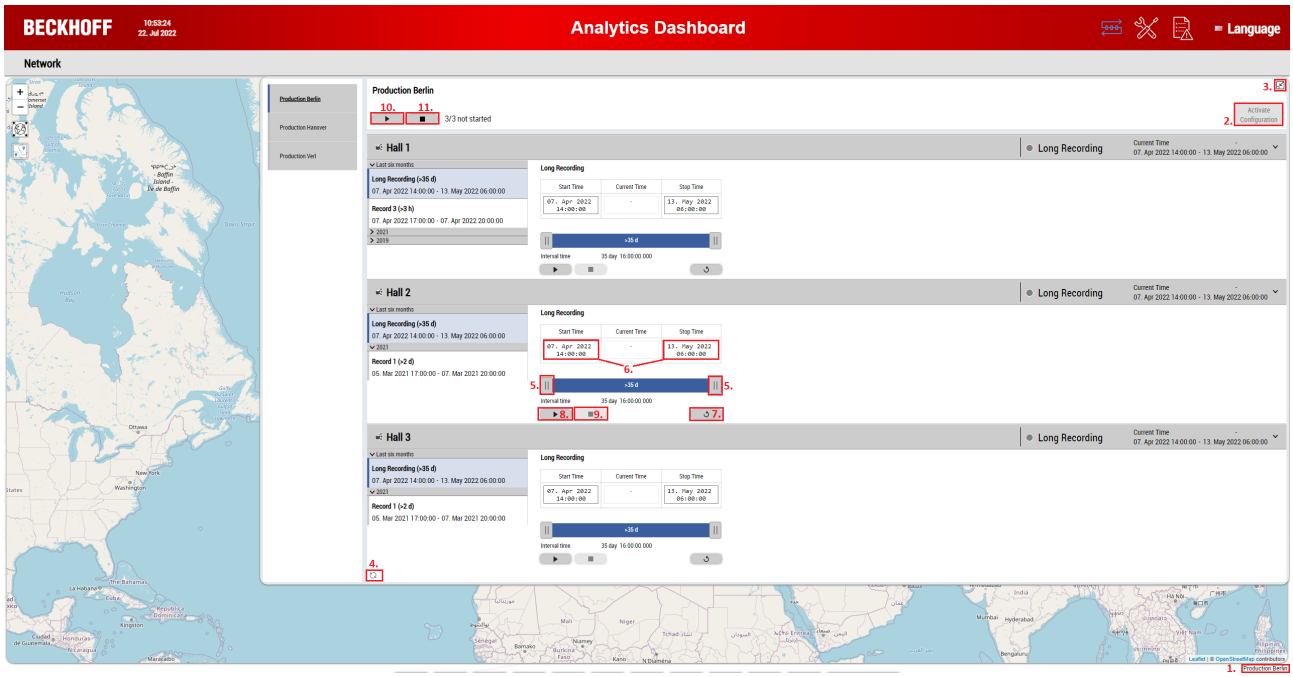
Neben der Karte mit den einzelnen Maschinenkonfigurationen lassen sich die Analysen zu den einzelnen Konfigurationen umschalten. In der SPS laufen alle Analysen in separaten Tasks, wodurch alle Analysedaten gleichzeitig zur Verfügung stehen. In dem generierten HMI können damit die verschiedenen Maschinen, in einem dafür speziell entwickelten Control, umgeschaltet werden. Das Control lässt sich über das im nachfolgenden Bild markierte Icon (schwarzer Pfeil) öffnen und schließen.



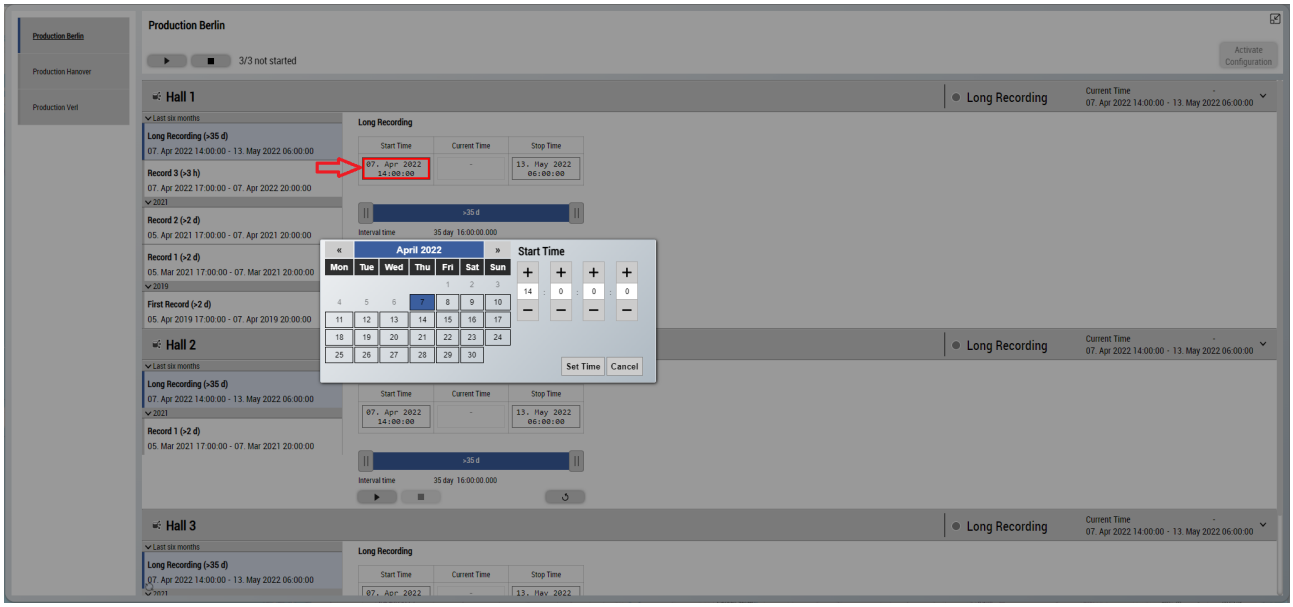
Die Konfigurationen, die zuvor auf der Karte mit den Standorten abgebildet waren, sind in dem Control auswählbar. Eine Konfiguration kann aus live und historischen Daten bestehen, wobei für die **Production Berlin** nur historische Daten konfiguriert sind. Historische Daten können äquivalent wie im Arbeit mit historischen Daten [▶ 363] in der HMI analysiert werden.

Beim Klicken auf eine andere Maschine wird diese Konfiguration nur im Control angezeigt. Klicken Sie zum Aktivieren auf **Activate Configuration**.
Im Folgenden wird das Control erklärt.

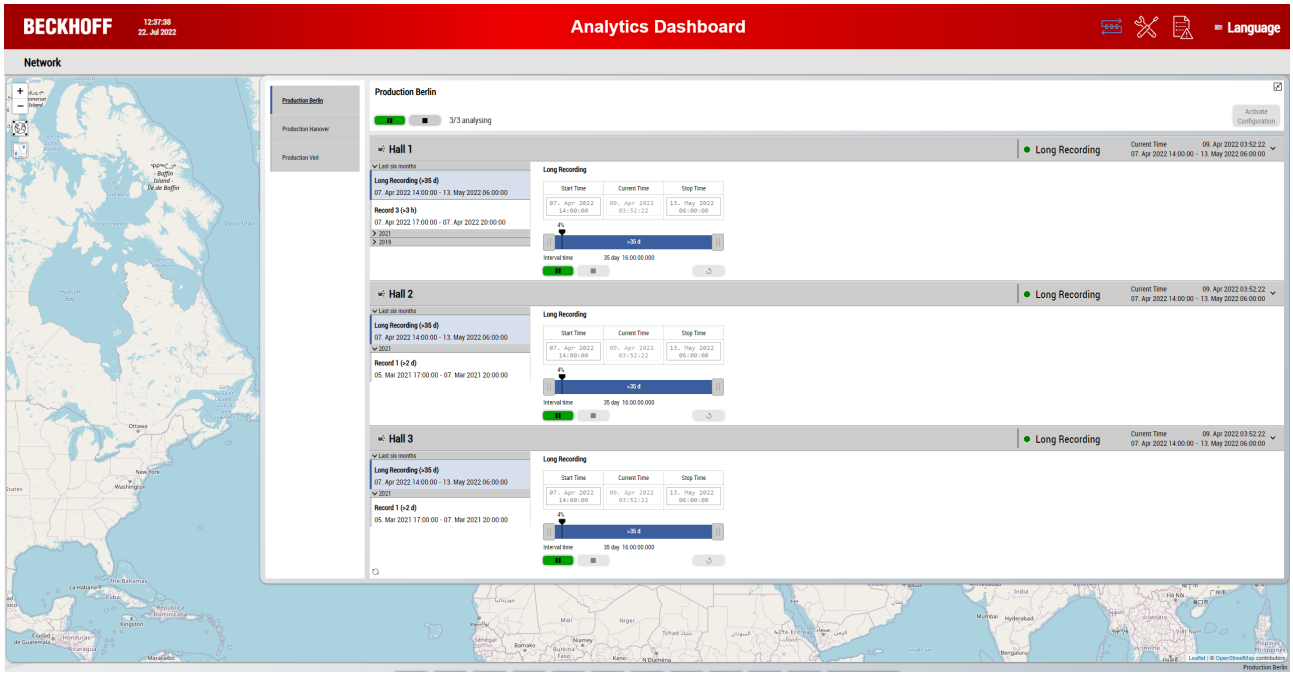
1. Die derzeit aktive Konfiguration.
2. Button, um die selektierte Konfiguration zu aktivieren.
3. Auswahlfenster zum Minimieren (und automatischer Aktivierung der selektierten Konfiguration).
4. Neuladen der Record-Liste. Damit können auch im Nachhinein aufgenommene Daten im Dashboard dynamisch geladen und analysiert werden.
5. Auswahl des Start- und Endzeitpunks über einen Slider.
6. Auswahl des Start- und Enddatums über ein Auswahlfenster mit Kalender.
7. Zurücksetzen des Start- und Endzeitpunks.
8. Starten der Analyse der selektierten **Input Source**.
9. Abbrechen der Analyse der selektierten **Input Source**.
10. Starten der gesamten Analyse
11. Abbrechen der gesamten Analyse



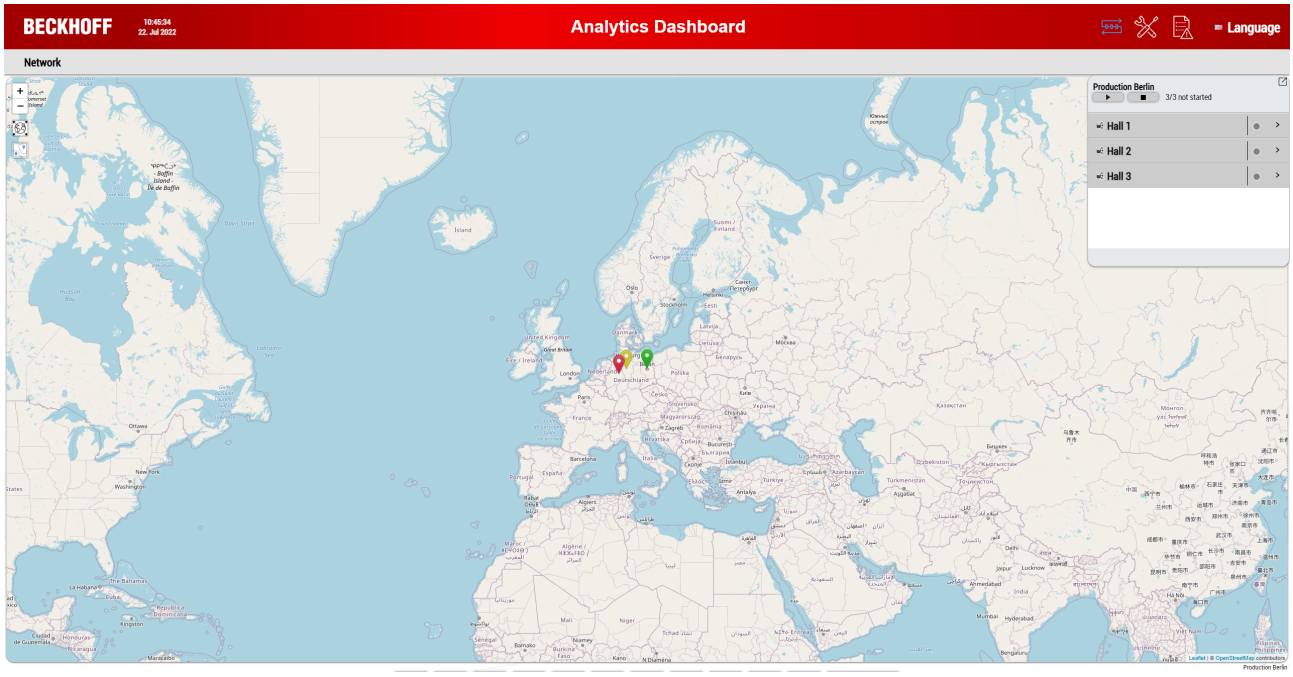
Für schnelle Tests und grobe Einstellung der Start- und Endzeit ist der Slider optimal. Klicken Sie für eine genaue Einstellung bis auf die Millisekunde auf das **Textfeld**. Damit öffnet sich ein Auswahlfenster mit Kalender zur Selektierung des Datums und zur Einstellung der genauen Uhrzeit.

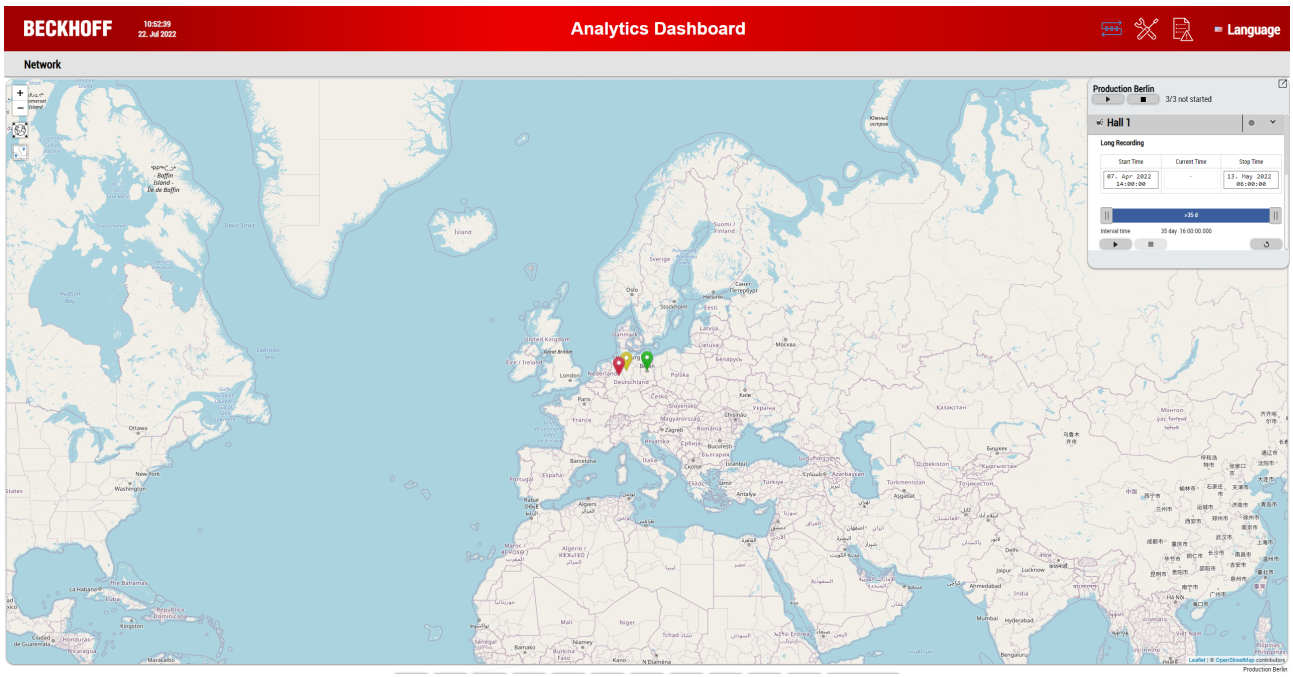


Sobald Sie den Button **Start** klicken, wird die Analyse mit allen Input Sourcen in der SPS angetriggert. Erstmals erscheint ein Ladesymbol, bis die Analyse anläuft. Dieser wird grün, sobald die ersten Daten erscheinen. Ebenfalls zeigt ein Marker den aktuellen Zeitpunkt der historischen Analyse im Slider an. Je nach Bandbreite, Anzahl der Daten und aufgenommene Zykluszeit kann der Prozess schneller oder langsamer abgearbeitet werden. Es ist möglich mehrere historische Aufnahmen gleichzeitig zu analysieren, indem einfach ein Maschinenwechsel vollzogen wird. Intern laufen alle Analysen parallel weiter. Im Dashboard kann damit einfach zwischen den Analysen umgeschaltet werden.



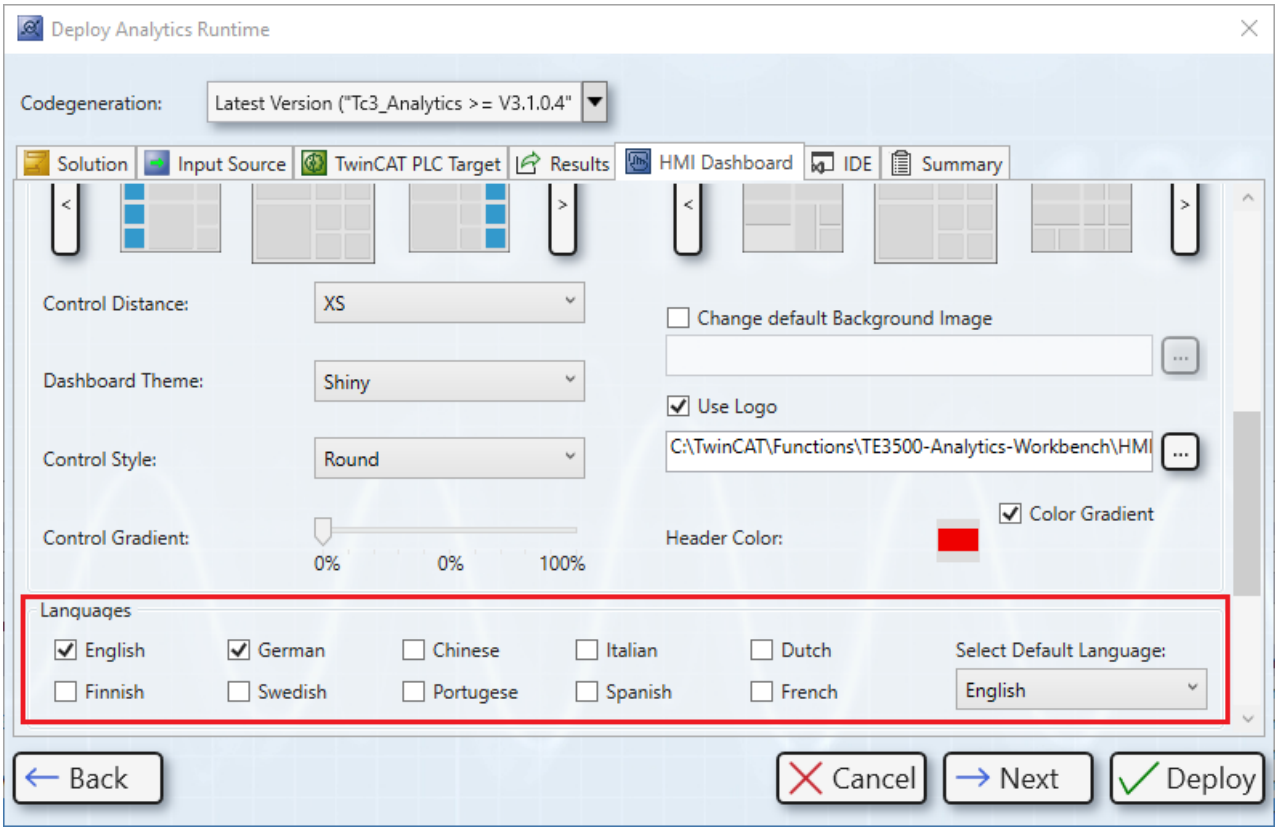
Über den Button **Minimieren** oben rechts in der Ecke wird die Auswahl der Konfigurationen und der Records ausgeblendet. Damit können Sie mit den Records von den Input Sourcen arbeiten und gleichzeitig die Daten betrachten. Sie können das Fenster ganz verstecken, indem Sie oben auf das blaue Icon klicken. Lediglich der Konfigurationsname steht immer rechts unten in der Ecke des Dashboards.





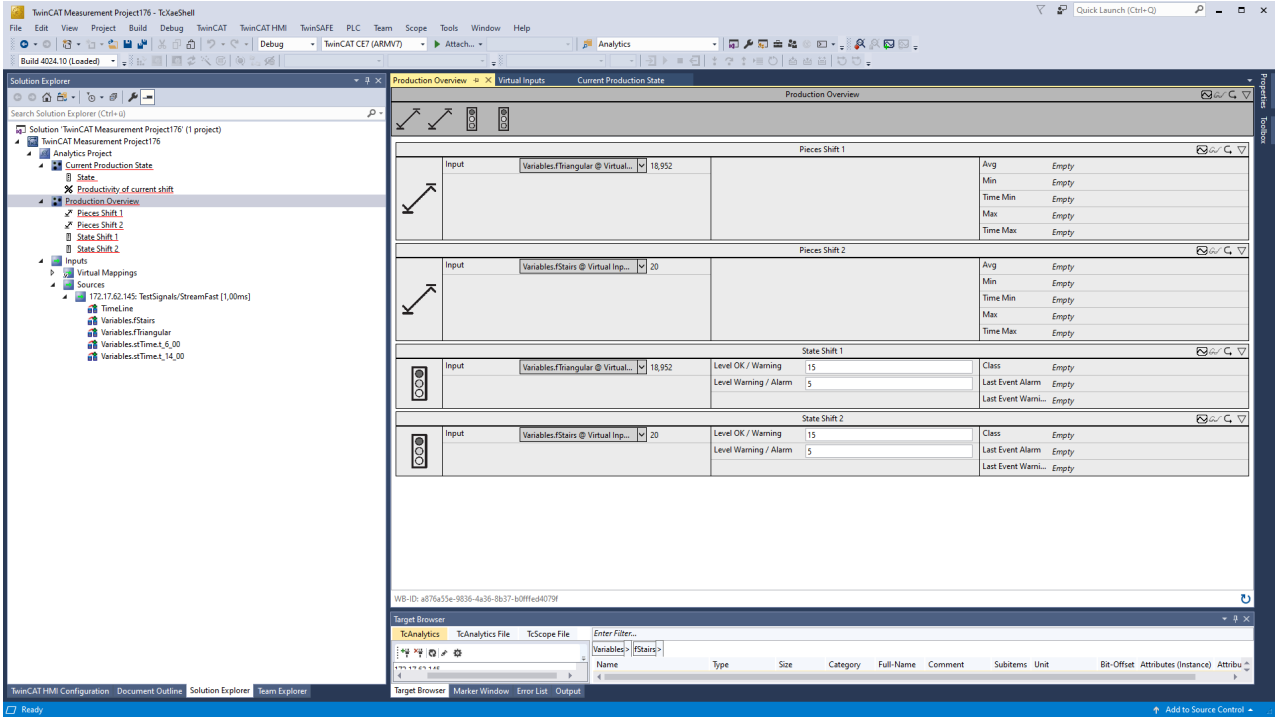
6.8.9 Integration einer Sprachumschaltung

In dem Konfigurationsfenster des Deploy Wizards können im Bereich der HMI bis zu 8 Sprachen für eine Sprachumschaltung ausgewählt werden.

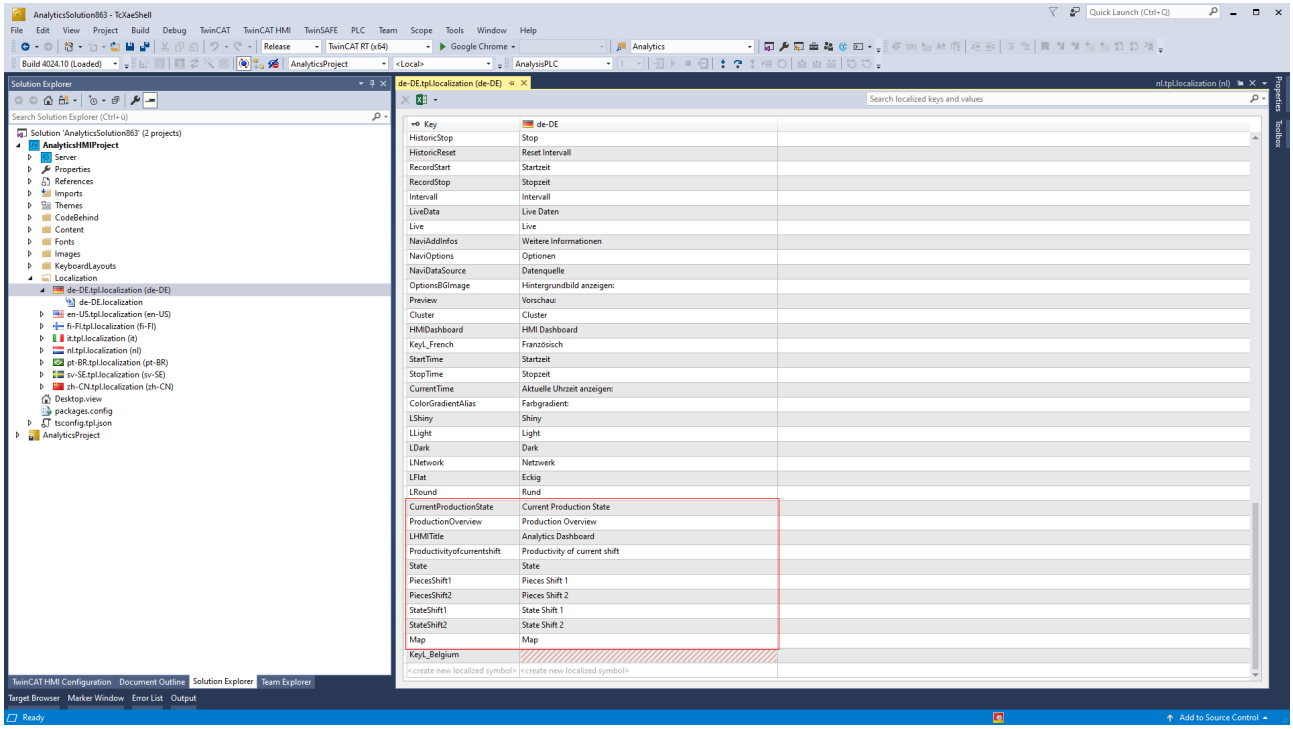


Wenn Sie die Benutzerverwaltung verwenden sind die Sprachen der angelegten Benutzer bereits ausgewählt. Diese können nicht abgewählt werden.

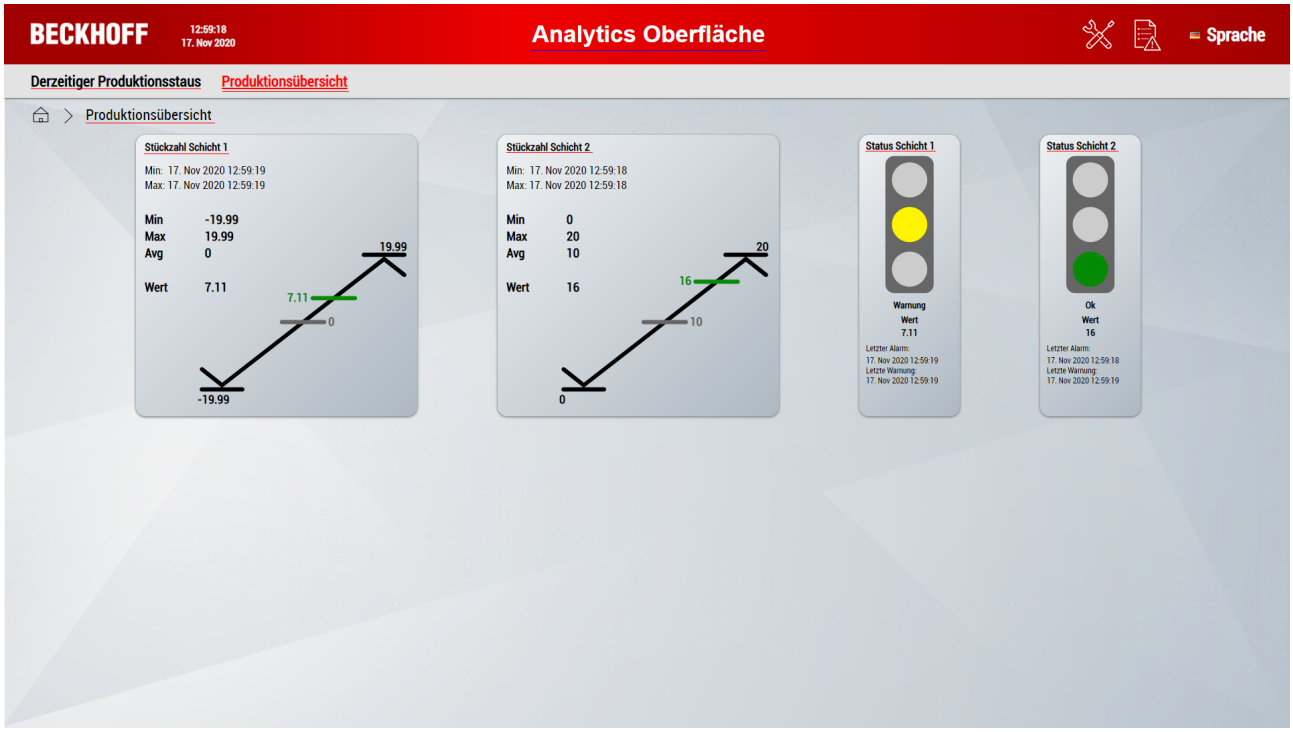
Die Texte in den mitgelieferten Analytics Controls sowie alle anderen Texte können automatisch umgeschaltet werden. Lediglich für die Benennung Ihrer Netzwerke und Module müssen Sie die Texte übersetzen, soweit Sie sich auch dafür eine Sprachumschaltung wünschen. In dem folgenden Screenshot sind die wesentlich betroffenen Namen im **Solution Explorer** gekennzeichnet.



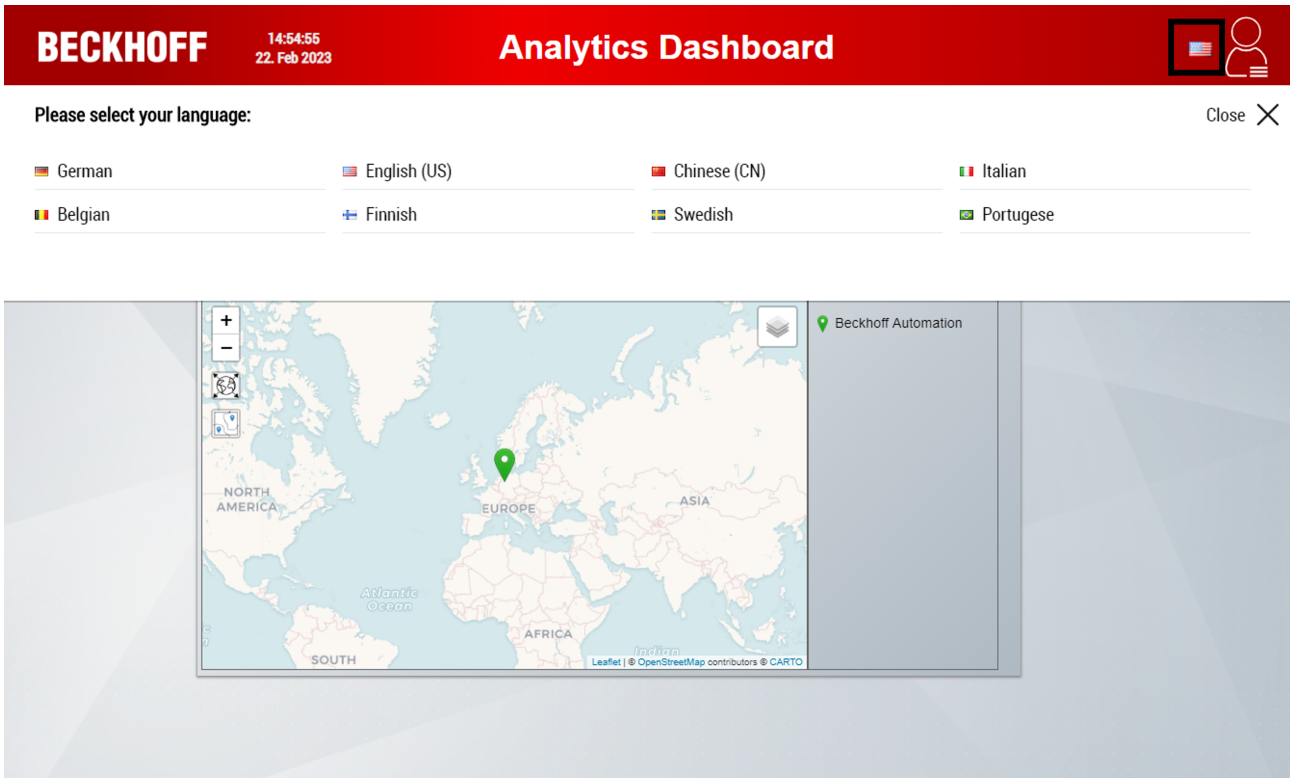
In dem generierten TwinCAT HMI Engineering Projekt ist dies auf einem einfachen Weg möglich, da die Übersetzungseinträge schon vorbereitet sind. In den **Localization** Dateien werden die Texte für die jeweiligen Sprachen gespeichert. Die Namen Ihrer Netzwerke und Module werden automatisch dort eingetragen, es muss lediglich die Übersetzung vorgenommen werden. Dafür gehen Sie in alle Dateien der Sprachen, für die Sie eine Übersetzung benötigen. In dem folgenden Screenshot sind die Einträge aufgelistet, die für die deutsche Sprache übersetzt werden müssen. Die Anzahl der zu bearbeitenden Einträge variiert je nach Komplexität des Analytics Projekts.



Danach können Sie das Dashboard über den Button **Google Chrome** (bzw. der Name ihres Standardbrowsers) öffnen. Ein reines Neuladen im Browser hilft nicht, da intern das Projekt neu gebaut werden muss. Im folgenden Bild sind die Texte markiert, die nun zusätzlich mit der Sprachumschaltung umgeschaltet werden.



Über die Flagge in der oberen rechten Ecke kann die Sprache gewechselt werden.



Die Sprache kann ab der Dashboard-Version 2.0 benutzerspezifisch festgelegt und gespeichert werden.

6.8.10 Verwenden der Interactive Funktionen in Standard TwinCAT HMI-Projekten

Sowohl das Interactive Dashboard als auch die Interactive Chart-Funktion können losgelöst vom One Click Dashboard in jedes TwinCAT HMI-Projekt integriert werden. Hierfür muss auf dem jeweiligen Zielsystem eine gültige TwinCAT Analytics Runtime oder Runtime Base-Lizenz (TF3550 oder TF3551) vorhanden sein.

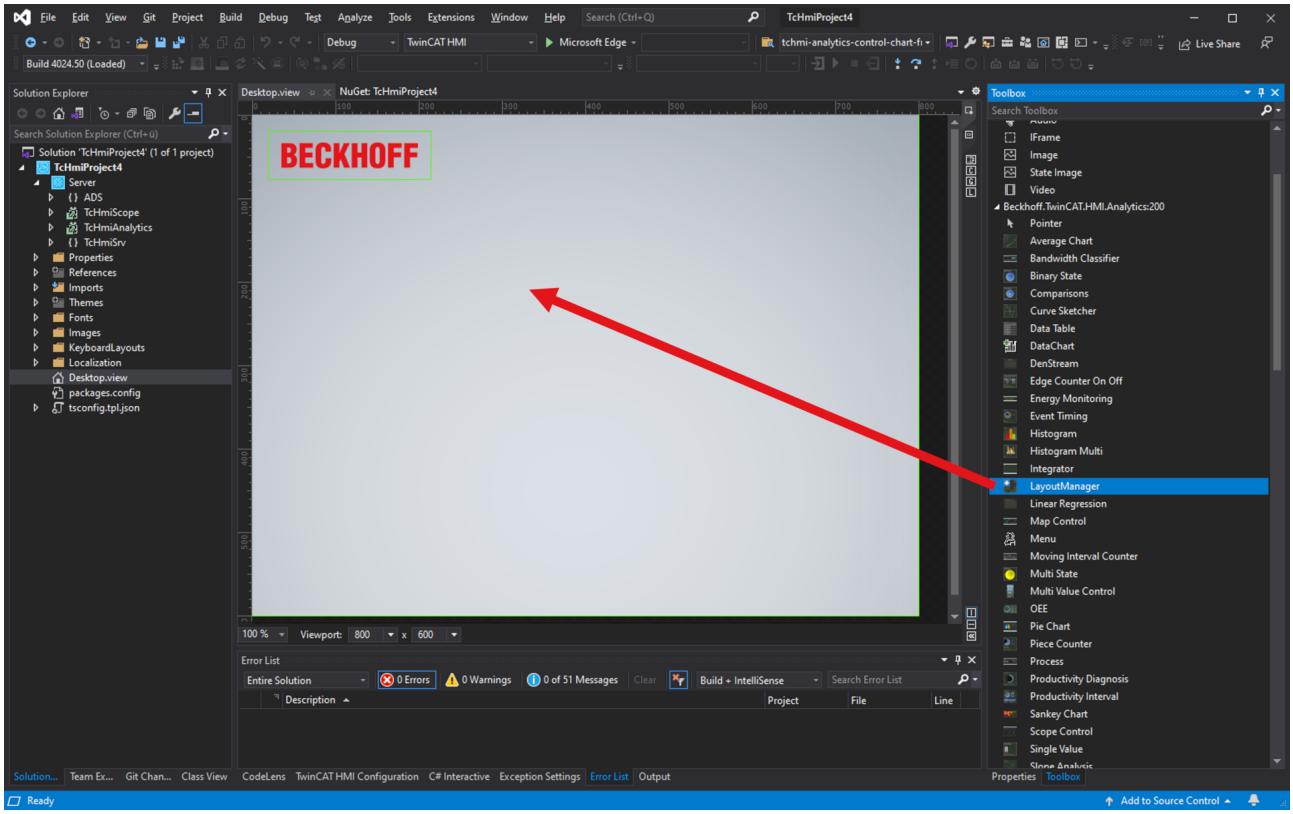
Installation der NuGet-Packages

Die benötigten Komponenten (Analytics-Framework-Controls und die Analytics-Server-Extension) können über NuGet-Packages in das HMI-Projekt integriert werden. Hierfür müssen im NuGet-Package-Manager die Packages „Beckhoff.TwinCAT.HMI.Analytics.ControlsExtension“ und „Beckhoff.TwinCAT.HMI.Analytics.Controls“ installiert werden.

Nach der Installation befindet sich im HMI-Projekt unter dem Server-Knoten die „TcHmiAnalytics“-Server-Extension. Sofern diese nicht automatisch startet, muss diese manuell über einen Rechtsklick auf die Extension gestartet werden.

Hinzufügen des Layout-Manager-Controls

Mittels des Layout-Manager-Controls können dynamische und benutzerspezifische Layouts erzeugt werden. Das Layout-Manager-Control kann über die Toolbox in das HMI-Projekt per Drag-and-Drop eingefügt werden. Es befindet sich im Bereich „Beckhoff.TwinCAT.HMI.Analytics“. Der Layout Manager ist ähnlich zum Container-Control. Der Unterschied liegt darin, dass die Controls in diesem im Browser umpositioniert werden können.



Setzen der Eigenschaften des Layout-Manager-Controls

Damit das Layout-Manager-Control optimal funktioniert, sollten folgende Eigenschaften gesetzt werden:

Name	Beschreibung
Identifier	Nach dem Abspeichern von Layouts mit der Server-Extension sollte der Identifier nicht mehr verändert werden. Andernfalls können gespeicherte Layouts nicht mehr zugeordnet werden.
WidthMode	Auswahl des Werts „Value“ sollte nicht verändert werden.
HeightMode	Auswahl des Werts „Value“ sollte nicht verändert werden.

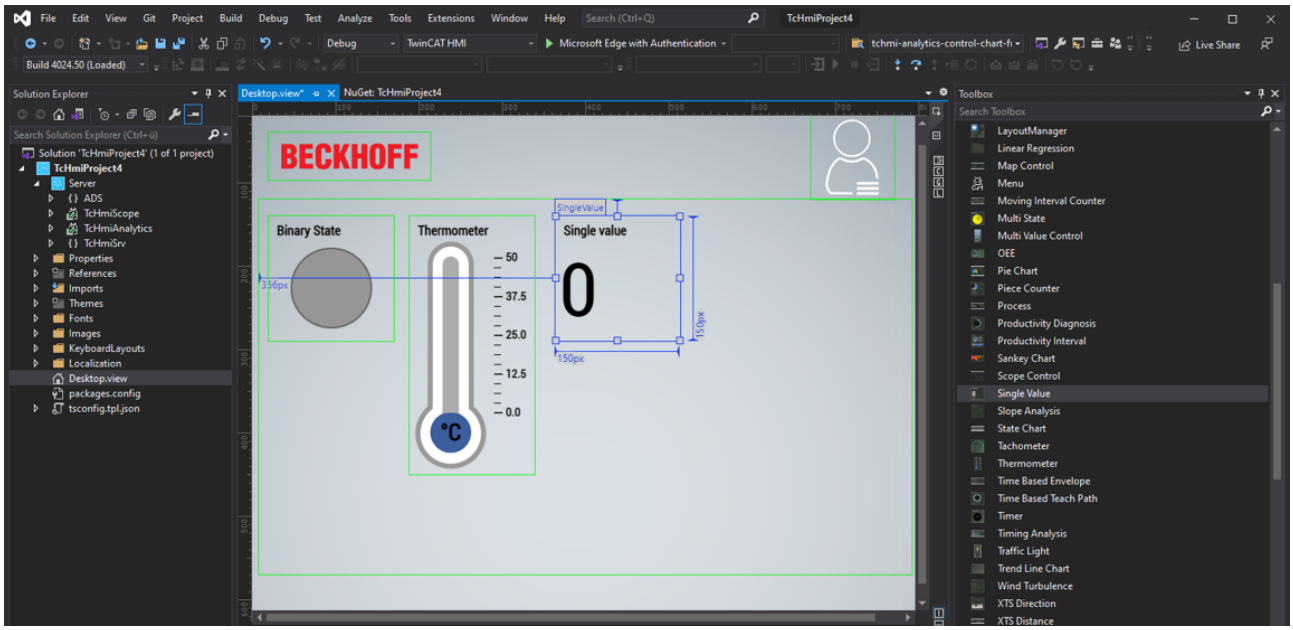
Folgende Parameter können optional gesetzt werden:

Name	Beschreibung
DarkMode	Auswahl, ob das Control im Dark-Mode dargestellt werden soll.
MobileViewWidth	Pixelbreite, ab der auf eine mobile Ansicht umgeschaltet wird.
Number of columns	Anzahl der Spalten, die das Layout in der normalen Ansicht umfassen soll.
Grid Gap	Prozentualer Abstand zwischen den Controls, welche auf dem Layout-Manager positioniert sind.
UseCustomRowHeight	Mit diesem Parameter kann die feste Zeilenhöhe, welche über den Parameter RowHeight festlegbar ist, aktiviert werden. Andernfalls wird eine dynamisch skalierte Höhe verwendet.
RowHeight	Mit diesem Parameter kann die Zeilenhöhe in Pixeln angegeben werden.
LockControls	Mit diesem Parameter kann verhindert werden, dass im Bearbeitungsmodus mit den im Layout existierenden Controls interagiert werden kann.
IsAnalyticsUse	Diese Eigenschaft sollte nur aktiviert werden, wenn das Layout-Manager-Control in einem Analytics-One-Click-Dashboard verwendet wird. So wird das Menü des Layout-Manager-Controls passend auf dem Layout positioniert.
MenuControlSymbol	Hier kann optional ein Control vom Typ „Menü“ (ebenfalls aus dem Bereich „Beckhoff.TwinCAT.HMI.Analytics“) verlinkt werden. Dieses sollte jederzeit auf dem Dashboard sichtbar sein und auf der „Desktop.view“ platziert sein. Über das Menü-Control kann ebenfalls der Bearbeitungsmodus geöffnet und geschlossen werden.

Controls hinzufügen

Anschließend können beliebige Controls auf das Layout-Manager-Control hinzugefügt werden. Diese können im HMI-Engineering beliebig verschoben werden.

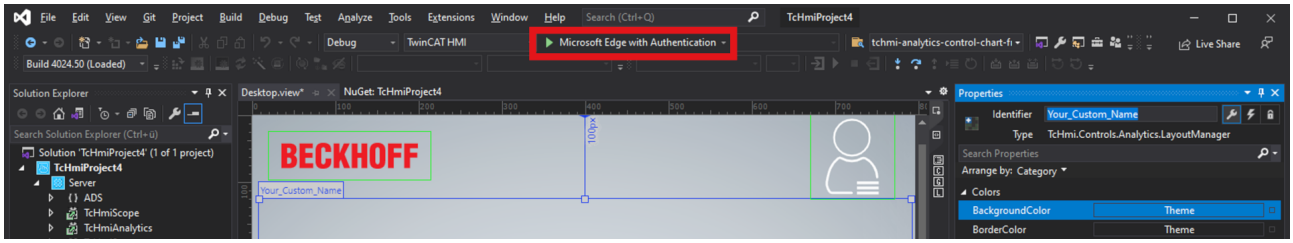
> Wurde für einen Benutzer ein Layout über die Interactive-Funktion gespeichert, werden die Positionen aus dem gespeicherten Layout geladen. Werden zu einem späteren Zeitpunkt weitere Controls hinzugefügt, werden diese an einen freien Platz innerhalb des Layouts einsortiert. Entfernte Controls verschwinden hingegen direkt aus dem Layout.



Interactive-Chart-Controls können auch im Webbrowser hinzufügt werden.

Projekt bauen

Um die Funktionen des Interactive Dashboards und Interactive Charts verwenden zu können, muss das Projekt mit Benutzerauthentifizierung gebaut werden.



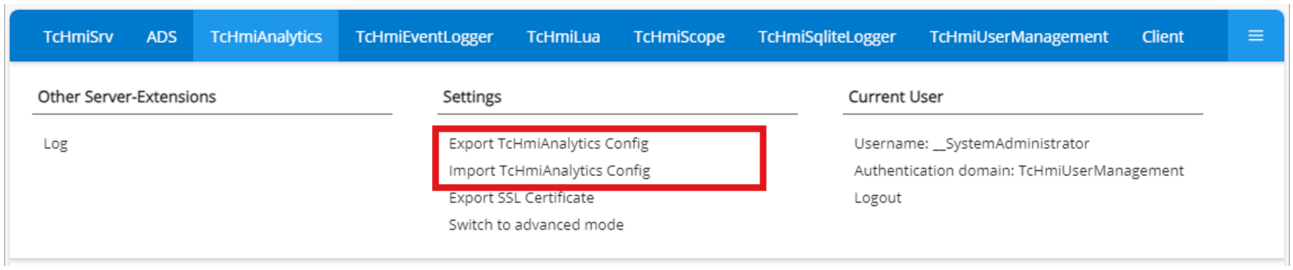
Benutzerspezifische Änderungen vornehmen

Anschließend lassen sich die Interactive Funktionen im Webbrowser verwenden. Auf folgenden Seiten können Informationen zu der Verwendung des Interactive Dashboards [▶ 476] und des Interactive Charts [▶ 482] eingesehen werden.

Layout- und Diagramm-Konfigurationen exportieren und importieren

Die in der „TcHmiAnalytics“-Server-Extension gespeicherten Konfigurationen von Layouts und Diagrammen können exportiert und importiert werden. Dies bietet sich an, wenn die Konfigurationen von einem remote laufenden Projekt (bspw. auf einem HMI-Server laufend) in ein lokales Projekt übernommen werden soll. Alternativ kann auch die Konfiguration eines Remote-Projekts aktualisiert werden.

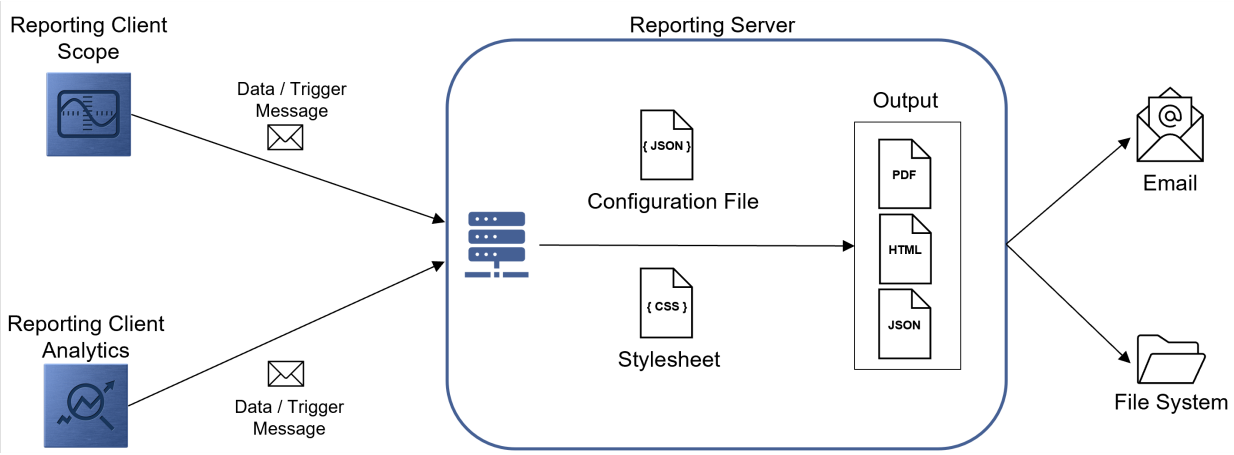
Der Export und Import der Konfigurationen in die Server-Extension lassen sich über die Config-Seite des jeweiligen HMI-Projekts vornehmen.



6.9 Reporting Integration

6.9.1 Grundkonzept

Die folgende Abbildung zeigt das Grundkonzept vom TwinCAT Reporting von den Reporting Clients bis zum Verteilen der generierten Reports.



Reporting Clients

Aktuell ist ein Reporting Client in dem TwinCAT Scope View als auch in der TwinCAT Analytics Workbench bzw. in dem TwinCAT Analytics Service Tool integriert. Diese Tools bieten die Möglichkeit, ein 24/7-Reporting und ein On-Demand-Reporting zu implementieren.

- Scope Reporting
- Analytics Reporting [▶ 520]

Auch über die SPS ist das 24/7-Reporting möglich. Die Funktionsbausteine haben die gleiche Funktionalität wie die Algorithmen aus TwinCAT Analytics.

- PLC Reporting

Die Reporting Clients kommunizieren mit dem TwinCAT Reporting Server. Die Reporting Clients können zwei Arten von Nachrichten verschicken. Zum einen können Daten-Nachrichten mit den dazugehörigen Daten versendet werden. Zum anderen können sogenannte Trigger-Nachrichten kommuniziert werden. Diese Art der Nachricht löst die Generierung eines Reports aus.

TwinCAT Reporting Server

Der TwinCAT Reporting Server [▶ 528] hat eine Vielzahl von Aufgaben. Der Server empfängt die Nachrichten der Reporting Clients und speichert die Daten in einem DataStore. Ein Report kann in einer dazugehörigen Konfigurationsdatei definiert werden. Die Zuordnung eines Datenobjektes zu einem Report wird über einen Report Name und einen Data Key hergestellt. Das Design des Reports lässt sich in einem Stylesheet individualisieren. Sobald eine Trigger-Nachricht empfangen wird, wird ein Report mit den dazugehörigen Daten in den konfigurierten Formaten generiert und entsprechend verteilt.

Glossar

Begriff	Erklärung
Report	Ein Report ist ein Dokument, welches in den Formaten PDF, HTML und JSON zur Verfügung steht. Es enthält die Informationen und Bilder.
Konfigurationsdatei	Die Konfigurationsdatei beschreibt den Aufbau eines Reports. Ebenfalls bietet es die Möglichkeit statische Informationen standardmäßig in einen Report zu integrieren. Des Weiteren können Einstellungen im Reporting Server für den entsprechenden Report gemacht werden.
Reporting Server	Der Reporting Server verwaltet die Konfigurationsdateien, empfängt Informationen und erstellt aus diesen Informationen individuelle Reports.
Reporting Client	Über einen Reporting Client können Daten an den Reporting-Server verschickt werden oder die Erstellung eines Reports ausgelöst werden.
Daten-Nachricht	Ein Reporting Client kann Informationen über eine Daten-Nachricht an den Reporting Server senden. Die enthaltenen Daten werden im Reporting Server zwischengehalten.
Trigger-Nachricht	Ein Reporting Client kann eine Trigger-Nachricht an den Reporting Server senden und löst damit die Generierung eines Reports aus. Der Reporting Server nutzt dazu alle Daten, die bis zu diesem Zeitpunkt empfangen wurden und zu dem entsprechenden Report gehören. Nach der Generierung werden die Daten gelöscht.
Report Name	Der Report Name dient der Identifikation eines Reports. Der Report Name ergibt sich aus dem Namen der Konfigurationsdatei ohne die Extension (z. B.: Konfigurationsdatei: Beckhoff Report Template.json; Report Name: Beckhoff Report Template)
Data Key	Ein Data Key identifiziert ein Datenobjekt eines Reports. Es muss innerhalb eines Reports und dessen Konfigurationsdatei eindeutig sein. Dadurch kann die Zuordnung und Sortierung der Datenobjekte im Report sichergestellt werden.
24/7-Reporting	Das 24/7-Reporting bietet die Möglichkeit, Daten zu sammeln und automatisch Reports zu generieren.
On-Demand-Reporting	Das On-Demand-Reporting bietet die Möglichkeit, individuelle Reports bei Bedarf zu generieren.

6.9.2 Analytics Reporting

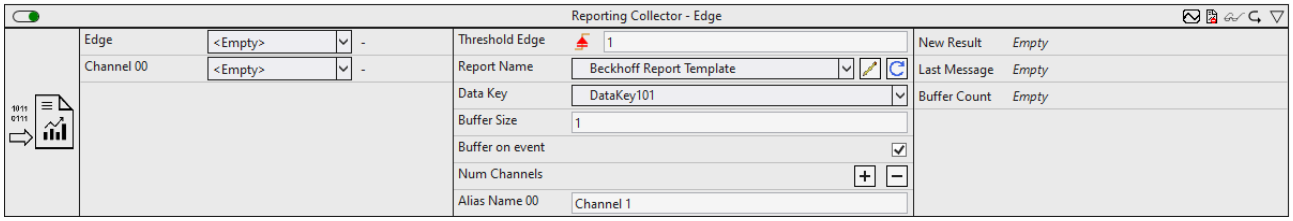
6.9.2.1 24/7-Reporting

Das 24/7-Reporting kann in TwinCAT Analytics mit den Algorithmen der Kategorie *Reporting* umgesetzt werden. Die Reporting Collectoren sammeln die Daten und versenden diese an den Reporting-Server. Die Reporting Trigger lösen die Erstellung eines Reports aus.

6.9.2.1.1 Reporting Collector

Die Reporting Collectoren sammeln Daten und versenden diese nach einem Event in einer Daten-Nachricht an den Reporting Server.

6.9.2.1.1.1 Reporting Collector Edge



Der Reporting Collector Edge sammelt die Daten der Eingangskanäle und sendet die Daten je nach Konfiguration nach einem Ereignis oder nach dem Befüllen des Buffers an den Reporting Server. Ein Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einer bestimmten Schwelle passiert.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

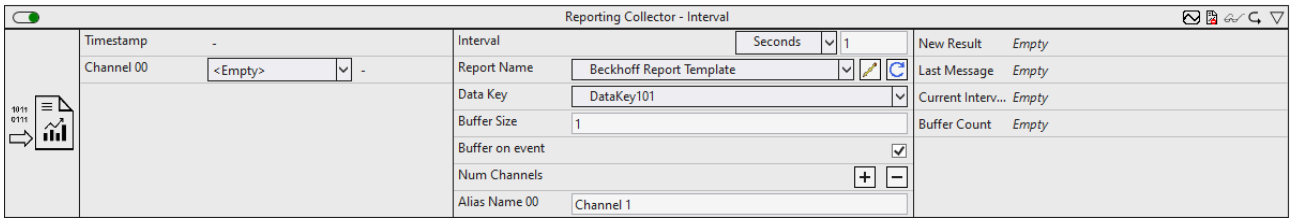
Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Ereignis ausgelöst.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.
- **Data Key:** Der Data Key sollte innerhalb eines Reports eindeutig sein. Über den Data Key lässt sich das Datenobjekt eindeutig in den Report einordnen.
- **Buffer Size:** Gibt die Größe der gepufferten Daten an. Ist die Buffer Size gleich eins, wird eine Key-Value-Struktur aufgebaut. Ist die Buffer Size größer als eins, werden die Daten in einer Tabelle dargestellt.
- **Buffer on event:** Gibt an, wie die Daten gesammelt und gepuffert werden sollen.
  Wenn der Parameter True ist, wird bei jedem Edge-Signal am Eingang die Daten der Inputs im Buffer zwischengespeichert. Sobald der Ausgang Buffer Count den gleichen Wert wie der Parameter Buffer Size hat, werden die Daten an den TcReportingServer versendet.
  Wenn der Parameter False ist, werden bei jedem Zyklus die Daten der Inputs im Buffer zwischengespeichert. Sobald die Buffer Size erreicht ist, ersetzen die neuen Daten die vorherigen Daten. Bei einem Edge-Signal am Eingang werden die Daten an den TcReportingServer versendet.
- **Num Channels:** Die Anzahl der Eingangskanäle, von denen die Daten gesammelt werden. Und die Anzahl der Parameter, um eine Beschreibung der Eingangskanäle zu ermöglichen.
- **Alias Name:** Dient als Beschreibung des Eingangskanals. BufferSize = 1: Der Alias Name dient als Key in der Key-Value-Struktur. BufferSize > 1: Der Alias Name dient als Überschrift der Tabellenspalte.
- **Include Timestamps (optional):** Fügt eine Spalte mit den Zeitstempeln ein.

Ausgangswerte

- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Buffer Count:** Gibt die Anzahl an Elementen im Buffer an.

6.9.2.1.1.2 Reporting Collector Interval



Der Reporting Collector Interval sammelt die Daten der Eingangskanäle und sendet die Daten je nach Konfiguration nach einem Ereignis oder nach dem Befüllen des Buffers an den Reporting Server. Ein Ereignis wird ausgelöst, wenn die Zeitspanne des konfigurierten Intervalls abgelaufen ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

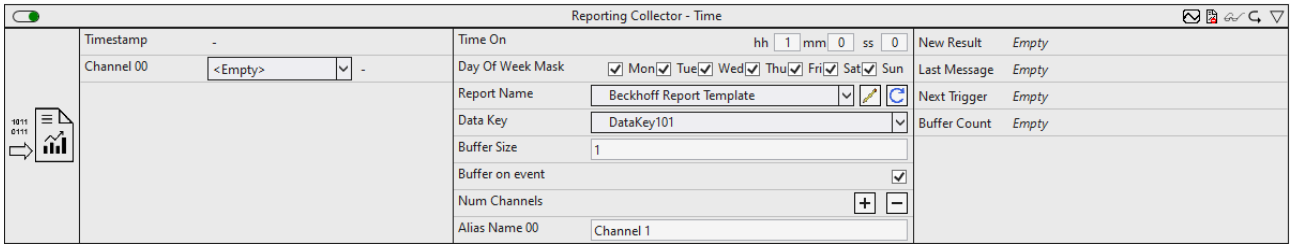
- **Interval:** Zeitintervall, in dem die Daten an den Reporting Server gesendet werden sollen.

- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.
- **Data Key:** Der Data Key sollte innerhalb eines Reports eindeutig sein. Über den Data Key lässt sich das Datenobjekt eindeutig in den Report einordnen.
- **Buffer Size:** Gibt die Größe der gepufferten Daten an. Ist die Buffer Size gleich eins, wird eine Key-Value-Struktur aufgebaut. Ist die Buffer Size größer als eins, werden die Daten in einer Tabelle dargestellt.
- **Buffer on event:** Gibt an, wie die Daten gesammelt und gepuffert werden sollen.
  Wenn der Parameter True ist, wird bei jedem Edge-Signal am Eingang die Daten der Inputs im Buffer zwischengespeichert. Sobald der Ausgang Buffer Count den gleichen Wert wie der Parameter Buffer Size hat, werden die Daten an den TcReportingServer versendet.
  Wenn der Parameter False ist, werden bei jedem Zyklus die Daten der Inputs im Buffer zwischengespeichert. Sobald die Buffer Size erreicht ist, ersetzen die neuen Daten die vorherigen Daten. Bei einem Edge-Signal am Eingang werden die Daten an den TcReportingServer versendet.
- **Num Channels:** Die Anzahl der Eingangskanäle, von denen die Daten gesammelt werden. Und die Anzahl der Parameter, um eine Beschreibung der Eingangskanäle zu ermöglichen.
- **Alias Name:** Dient als Beschreibung des Eingangskanals. BufferSize = 1: Der Alias Name dient als Key in der Key-Value-Struktur. BufferSize > 1: Der Alias Name dient als Überschrift der Tabellenspalte.
- **Include Timestamps (optional):** Fügt eine Spalte mit den Zeitstempeln ein.

Ausgangswerte

- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Current Interval Time:** Zeigt die Zeitspanne an, die vom aktuellen Intervall bereits verstrichen ist.
- **Buffer Count:** Gibt die Anzahl an Elementen im Buffer an.

6.9.2.1.1.3 Reporting Collector Time



Der Reporting Collector Time sammelt die Daten der Eingangskanäle und sendet die Daten je nach Konfiguration nach einem Ereignis oder nach dem Befüllen des Buffers an den Reporting Server. Ein Ereignis wird ausgelöst, wenn die konfigurierte Einschaltzeit erreicht wird. Die Einschaltzeit und die Tage der Wochen lassen sich konfigurieren.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Time On:** Einschaltzeit.
- **Day of Week Mask:** Wochentage, an denen die Zeitschaltung aktiv sein soll.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.
- **Data Key:** Der Data Key sollte innerhalb eines Reports eindeutig sein. Über den Data Key lässt sich das Datenobjekt eindeutig in den Report einordnen.
- **Buffer Size:** Gibt die Größe der gepufferten Daten an. Ist die Buffer Size gleich eins, wird eine Key-Value-Struktur aufgebaut. Ist die Buffer Size größer als eins, werden die Daten in einer Tabelle dargestellt.

- **Buffer on event:** Gibt an, wie die Daten gesammelt und gepuffert werden sollen.
  Wenn der Parameter True ist, wird bei jedem Edge-Signal am Eingang die Daten der Inputs im Buffer zwischengespeichert. Sobald der Ausgang Buffer Count den gleichen Wert wie der Parameter Buffer Size hat, werden die Daten an den TcReportingServer versendet.
  Wenn der Parameter False ist, werden bei jedem Zyklus die Daten der Inputs im Buffer zwischengespeichert. Sobald die Buffer Size erreicht ist, ersetzen die neuen Daten die vorherigen Daten. Bei einem Edge-Signal am Eingang werden die Daten an den TcReportingServer versendet.
- **Num Channels:** Die Anzahl der Eingangskanäle, von denen die Daten gesammelt werden. Und die Anzahl der Parameter, um eine Beschreibung der Eingangskanäle zu ermöglichen.
- **Alias Name:** Dient als Beschreibung des Eingangskanals. BufferSize = 1: Der Alias Name dient als Key in der Key-Value-Struktur. BufferSize > 1: Der Alias Name dient als Überschrift der Tabellenspalte.
- **Include Timestamps (optional):** Fügt eine Spalte mit den Zeitstempeln ein.

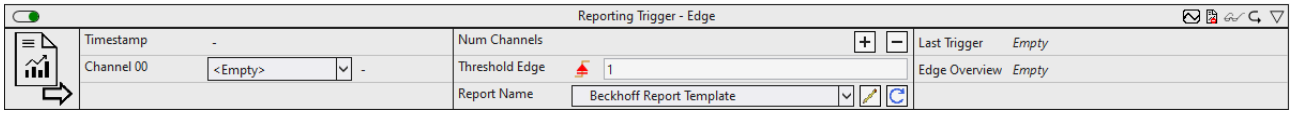
Ausgangswerte

- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Next Message:** Zeigt die verbleibende Zeit bis zur nächsten Nachricht an.
- **Buffer Count:** Gibt die Anzahl an Elementen im Buffer an.

6.9.2.1.2 Reporting Trigger

Die Reporting Trigger versenden nach einem Event eine Trigger-Nachricht an den Reporting Server und lösen damit die Erstellung eines Reports aus.

6.9.2.1.2.1 Reporting Trigger Edge



Der Reporting Trigger Edge löst die Erstellung eines Reports aus, nachdem ein Ereignis ausgelöst wurde. Ein Ereignis wird ausgelöst, wenn das Signal des Eingangskanals die konfigurierte Flanke bei einem bestimmten Schwellenwert überschreitet. Intern bleiben die Eingänge, die einmal True waren auf True. Die Eingänge werden lediglich auf False zurückgesetzt, sobald alle Eingänge mindestens einmal True waren. Dadurch kann der Ausgang bNewResult von mehreren Reporting Collectoren als ein Eingang genutzt werden und sobald alle Reporting Collectoren eine Daten-Nachricht gesendet haben, wird eine Trigger-Nachricht versendet.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

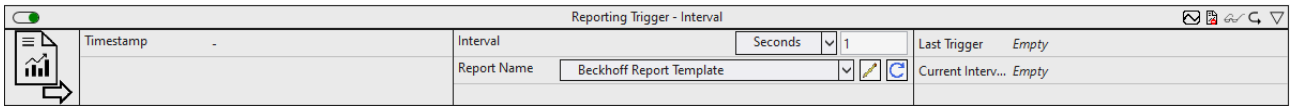
Konfigurationsoptionen

- **Type of the edge:** Gibt an, ob der Flankenzähler auf eine steigende oder eine fallende Flanke reagieren soll.
- **Threshold:** Schwelle des Signals an der jeweiligen Flanke. Wenn das Signal diese Schwelle passiert, wird das Ereignis ausgelöst.
- **Tolerance (optional):** Toleranzwert für die Vergleiche Equal / NotEqual.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.

Ausgangswerte

- **Last Trigger:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Edge Overview:** Zeigt an, welche Eingangskanäle mindestens einmal True waren.

6.9.2.1.2.2 Reporting Trigger Interval



Der Reporting Trigger Interval löst die Erstellung eines Reports aus, nachdem ein Ereignis ausgelöst wurde. Ein Ereignis wird ausgelöst, wenn die Zeitspanne des konfigurierten Intervalls abgelaufen ist.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

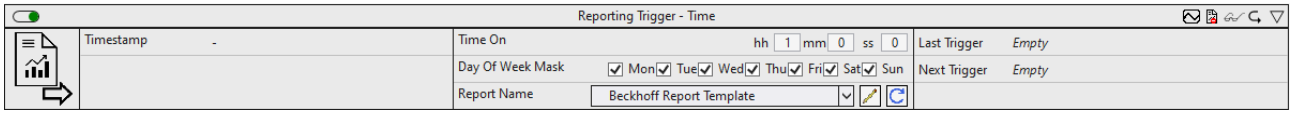
Konfigurationsoptionen

- **Interval:** Zeitintervall, in dem die Daten an den Reporting Server gesendet werden sollen.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.

Ausgangswerte

- **Last Trigger:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Current Interval Time:** Zeigt die Zeitspanne an, die vom aktuellen Intervall bereits verstrichen ist.

6.9.2.1.2.3 Reporting Trigger Time



Der Reporting Trigger Time löst die Erstellung eines Reports aus, nachdem ein Ereignis ausgelöst wurde. Ein Ereignis wird ausgelöst, wenn die konfigurierte Einschaltzeit erreicht wird. Die Einschaltzeit und die Tage der Wochen lassen sich konfigurieren.

Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Time On:** Einschaltzeit.
- **Day of Week Mask:** Wochentage, an denen die Zeitschaltung aktiv sein soll.
- **Report Name:** Gibt den Namen des Reports an. Der Name muss dem Namen der Konfigurationsdatei beim Reporting Server entsprechen. Über das Drop-Down-Menü werden alle Report Namen angezeigt, die im Configuration-Ordner vom Reporting Server liegen. Über den Edit-Button kann die ausgewählte Konfigurationsdatei bearbeitet werden.

Ausgangswerte

- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Next Trigger:** Zeigt die verbleibende Zeit bis zur nächsten Nachricht an.

6.9.2.2 On-Demand-Reporting

Das On-Demand-Reporting bietet die Möglichkeit, die Daten der Algorithmen und zusätzliche Daten manuell in einen Report zu integrieren.

Jeder Algorithmus hat dieselben fünf Icons in der oberen rechten Ecke. Das zweite Icon von links heißt **Include to Report** Icon. Über dieses Icon lässt sich der Algorithmus zu einem manuellen Report hinzufügen.

Dieses Icon symbolisiert, dass der Algorithmus zu einem manuellen Report hinzugefügt wird:

Dieses Icon symbolisiert, dass der Algorithmus in einem manuellen Report nicht enthalten ist:

Ein Netzwerk enthält ebenfalls diese Icons. Über diese werden alle enthaltenden Netzwerke und Algorithmen zum manuellen Report hinzugefügt beziehungsweise ignoriert.

Die Funktionalität des **Include to Report** Icons lässt sich auch über die Properties abbilden.

Bei einem Netzwerk sehen die Properties folgendermaßen aus:

Reporting	
Data Key	**MyNetwork**
Include in Report	**False**
Include SubMember in Report	**True**
Reporting visualization	**Reporting Table Default**

Geschlossene Netzwerke werden bei dem On-Demand-Reporting wie Algorithmen behandelt.

Reporting

Data Key	Über den Data Key lässt sich das Datenobjekt im Report eindeutig identifizieren. Der Data Key wird standardmäßig mit dem Netzwerk-Namen vorbelegt.
Include in Report	Diese Option gibt an, ob das Netzwerk zu einem manuellen Report hinzugefügt wird. Als darzustellende Inhalte werden die Netzwerk-Ein- und Ausgänge beziehungsweise die Netzwerk-Parameter verwendet.
Include SubMember in Report	Diese Option gibt an, ob die in dem Netzwerk enthaltenden Analytics Elemente in den Report integriert werden sollen.
Reporting visualization	Diese Option gibt an, wie die Daten des Netzwerks dargestellt werden sollen.

Bei einem Algorithmus sehen die Properties folgendermaßen aus:

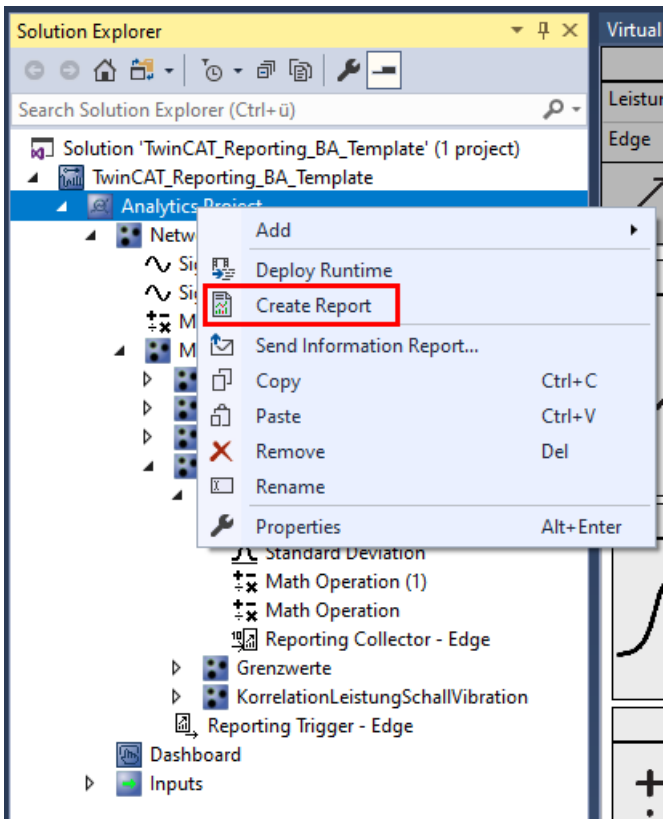
Reporting	
Data Key	**MyAlgorithm**
Include in Report	**True**
Reporting visualization	**Reporting Table Horizontal**

Reporting

Data Key	Über den Data Key lässt sich das Datenobjekt im Report eindeutig identifizieren. Der Data Key wird standardmäßig mit dem Netzwerk-Namen und dem Namen des Algorithmus vorbelegt.
Include in Report	Diese Option gibt an, ob der Algorithmus zu einem manuellen Report hinzugefügt wird.
Reporting visualization	Diese Option gibt an, wie die Daten des Algorithmus dargestellt werden sollen.

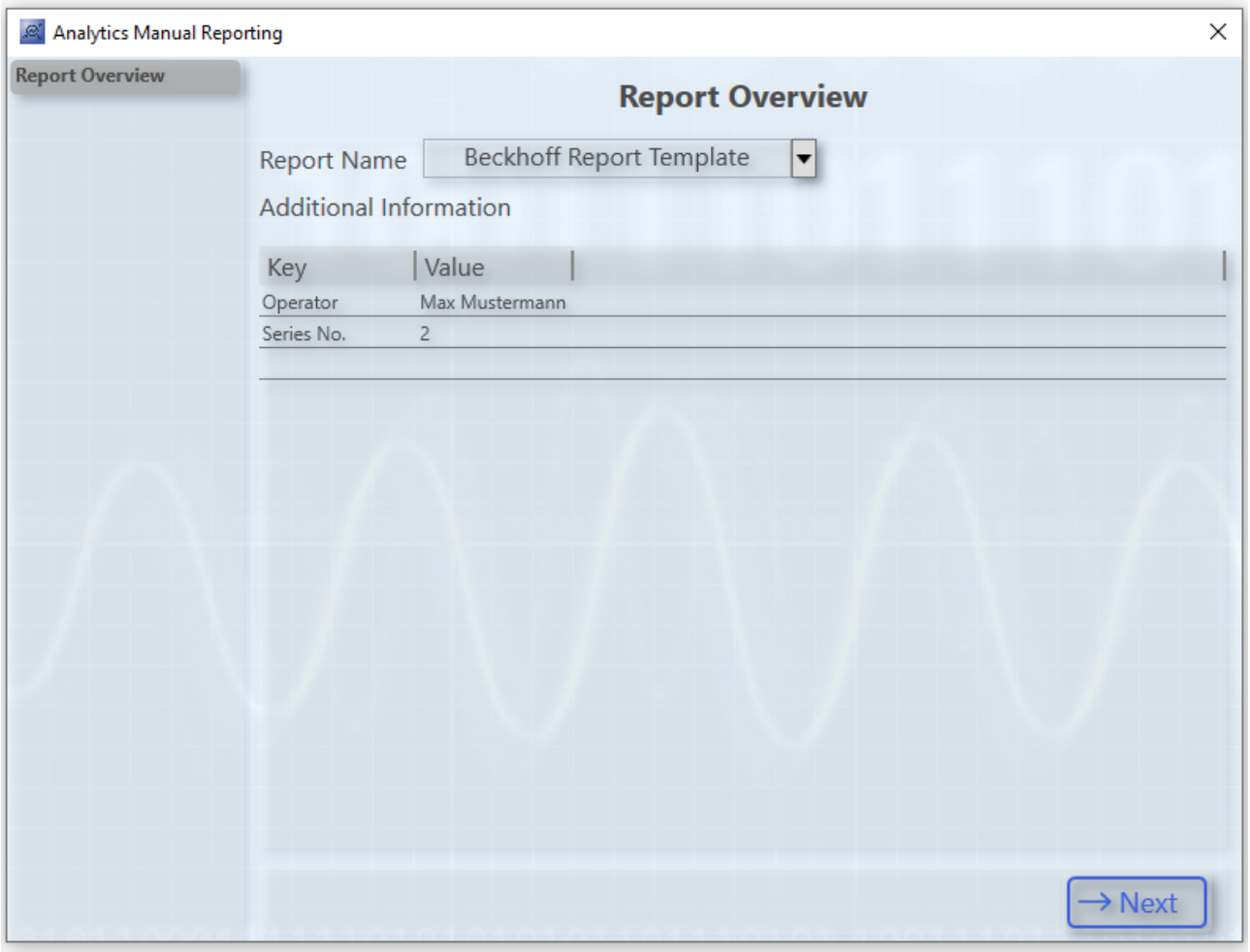
Das On-Demand-Reporting kann über zwei Möglichkeiten gestartet werden. Zum einen lässt sich das Reporting über den Reporting Button in der Toolbar öffnen:

Zum anderen kann es über das Kontextmenü auf dem Analytics Projekt gestartet werden:



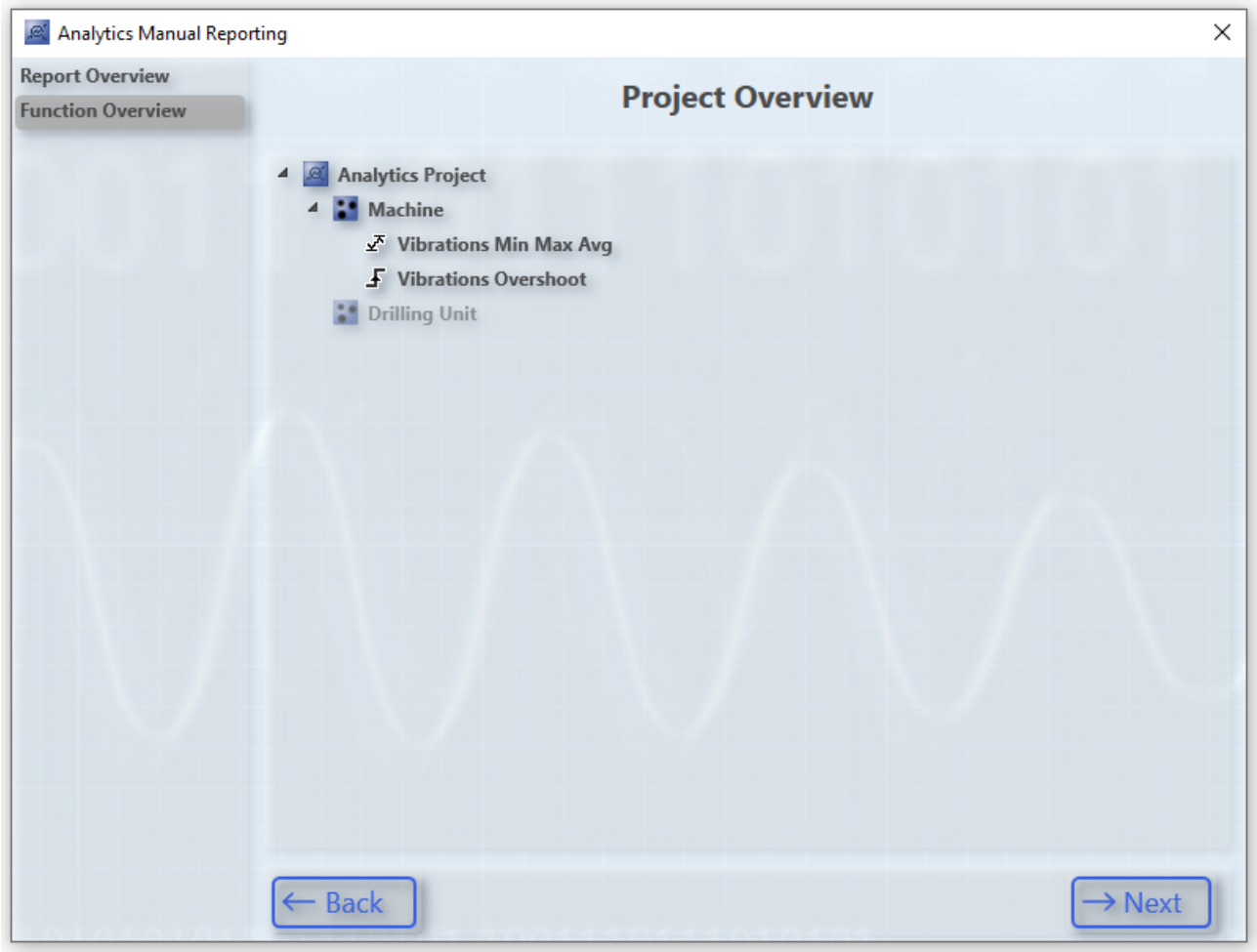
Anschließend öffnet sich der Wizard für das On-Demand-Reporting.

Auf der ersten Seite lassen sich zusätzliche Informationen zum Report hinzufügen.

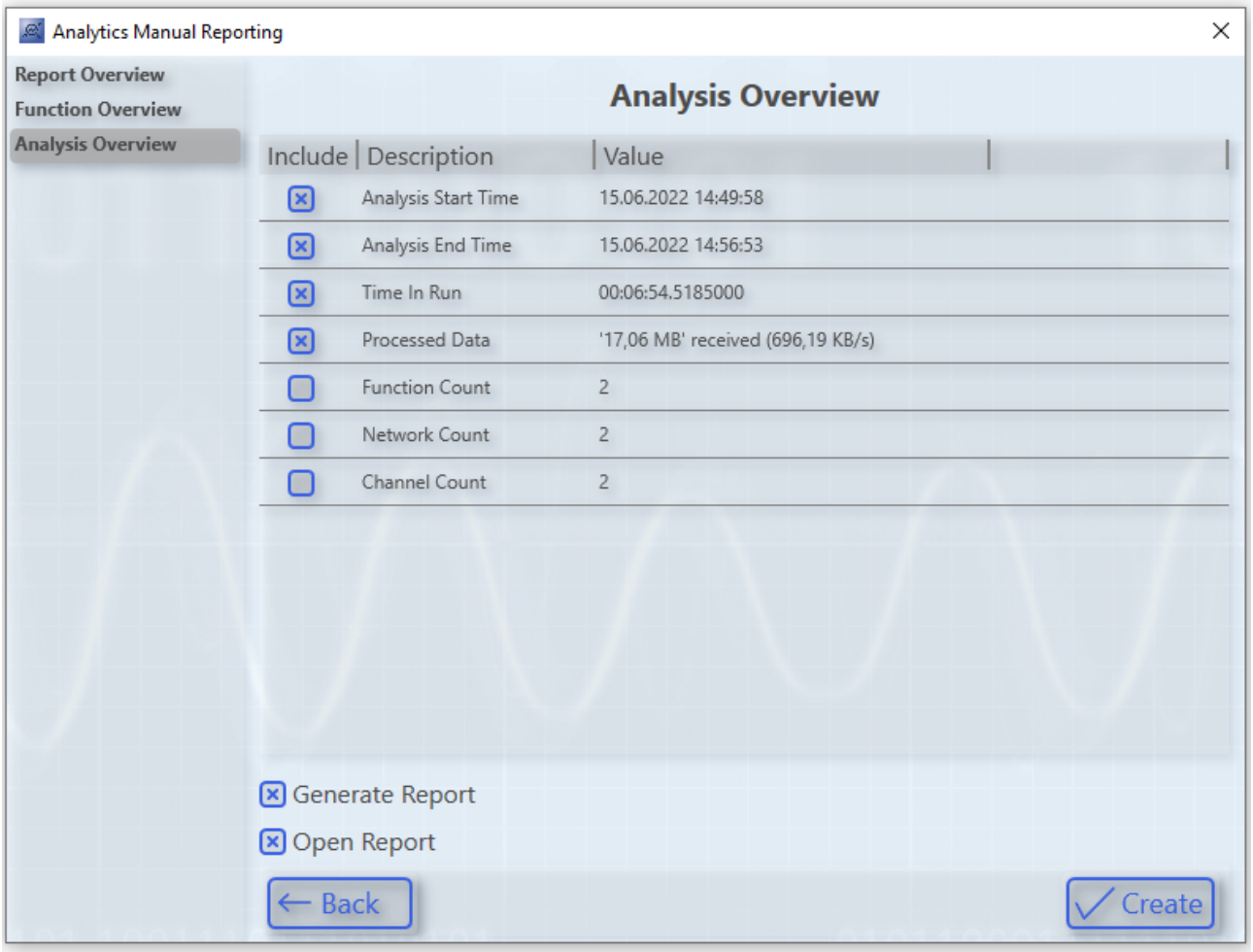


Die zusätzlichen Informationen sind in einer Key-Value-Struktur aufgebaut. Über einen Klick auf die leere Zeile lässt sich eine weitere Zeile hinzufügen. Eine ausgewählte Zeile kann über einen Druck auf die Entfernen-Taste gelöscht werden.

Auf der zweiten Seite wird eine Übersicht des Projektes dargestellt. Die ausgegrauten Algorithmen und Netzwerke werden in dem Report nicht berücksichtigt.



Auf der dritten Seite können Informationen zur Analyse zum Report hinzugefügt werden.



In der Tabelle steht eine Vielzahl von Information für die Integration in einem Report bereit. Über die Checkbox in der Spalte Include können einzelne Informationen aus- beziehungsweise abgewählt werden.

Die Option Generate Report legt fest, ob der Report mit den konfigurierten Daten generiert werden soll. Ist diese Option nicht ausgewählt, werden lediglich die Daten an den Reporting-Server versenden, der Report wird jedoch noch nicht generiert. Diese Möglichkeit ist sinnvoll, wenn aus mehreren Produkten Daten manuell zu einem einzigen Report hinzugefügt werden sollen.

Die Option Open Report legt fest, ob der Report nach der Generierung geöffnet werden soll. Um diese Option nutzen zu können, muss eine FilePublishLocation mit den Formaten PDF beziehungsweise HTML in der Konfigurationsdatei definiert sein. Wird der Report im PDF- und HTML-Format generiert, wird der PDF-Report standardmäßig geöffnet.

6.9.3 TwinCAT Reporting Server

6.9.3.1 Design

6.9.3.1.1 Konfigurationsdatei

Die Konfigurationsdatei bietet die Möglichkeit, den Report zu organisieren, zu sortieren und individuell zu gestalten. Die Konfigurationsdateien müssen in dem Ordner *.\TwinCAT\Functions\Reporting-Server\Configuration\ConfigFiles* abgelegt werden.

Folgende Optionen werden in dem Reporting Server unterstützt.

Key	Beschreibung	Beispiel	Ebene
General			
Store Type	Gibt an, welcher Store-Modus verwendet werden soll. Aktuell wird nur „Ram“ unterstützt.	`"StoreType": "Ram"`	1

Header	Bietet die Möglichkeit, drei untereinander angeordnete Überschriften darzustellen. Wenn die zweite und / oder dritte Überschrift nicht verwendet werden soll, kann ein leerer String angegeben werden.	`"Header": [`
		`"Header 1",`
		`"Header 2",`
		`"Header 3",`
		`]`
			1

Footer	Der Footer einer PDF-Seite teilt sich in drei Bereich auf. Alle drei Bereiche lassen sich durch einen Freitext individualisieren. Um das Datum und die Uhrzeit der Erstellung dazustellen, muss der Platzhalter „{date} {time}“ angegeben werden. Um die Seitenzahl dazustellen, muss der Platzhalter „{page} of {total-pages}“ angegeben werden. Wenn ein Bereich frei bleiben soll, kann ein leerer String angegeben werden.	`"Footer": [`
		`"Footer Left",`
		`"Footer Center",`
		`"Footer Right",`
		`]`
			1

Logo	Das Logo wird auf der ersten Seite oben rechts dargestellt. Das Logo kann als relativer Pfad zu dem Ordner *Reporting-Server\Configuration\ConfigFiles* angegeben werden. Alternativ kann das Logo als Base64String mit dem voranstehenden Encoding angegeben werden.	`"Logo": ".//<Bildname.png>"`
		oder
		`"Logo": "data:image/<Encoding>base64,<Base64String>"`
			1

StyleSheet	Bietet die Möglichkeit, einen Report designtechnisch zu individualisieren. Um ein anwenderspezifisches Stylesheet zu verwenden, muss dessen Name angegeben werden (siehe Stylesheet [▶ 532]).	`"StyleSheet": "StyleBeckhoff.css"`	1
SignatureSettingsFile	Bietet die Möglichkeit, einen Report zu signieren (siehe Signierung [▶ 535]).	`"SignatureSettingsFile": "BeckhoffSignatures.json",`	
Email Text	Dieser Text wird in jede E-Mail eingefügt, sofern kein spezifischer Text angegeben ist.	`"EmailText": "Custom of E-Mails send to all recipients with no or empty E-Mail-Text. Special. Ones"`	1
Required Keys	Ist eine Sammlung aus Data Key. Sofern die Datenobjekte dieser Data Keys nicht im Reporting-Server vorliegen, wird der Report nicht generiert.	`"RequiredKeys": ["Data101"]`	1
Trigger On Required Key Received	Ist eine weitere Möglichkeit, um die Generierung eines Reports auszulösen. Wenn bei der Option true ausgewählt ist und alle Datenobjekten mit den erforderlichen Data Keys vorhanden sind, wird der Report generiert. Dadurch ist eine Trigger-Nachricht von einem Reporting Client nicht nötig.	`"TriggerOnRequiredKeyReceived" : false`	1
Static Default Data			
Default Data	In diesem Objekt können Datenobjekte definiert werden, die in jedem Report dargestellt werden sollen.	`"DefaultData": {}`	1
Text Fields	Dieses Element ist ein Array und kann mehrere Textfeld-Objekte mit den folgenden Elementen besitzen.	`"TextFields": []`	2
Key	Wird als Überschrift für das Textfeld verwendet.	`"Key": "My Text Field"`	3
Comment	Dient als Kommentar für das Datenobjekt. Der Kommentar wird im Report nicht dargestellt.	`"Comment": "My Comment"`	3

Key	Beschreibung	Beispiel	Ebene
Sort Priority	Gibt die Position des Datenobjektes im Report an.	`"SortPriority": 42`	3
Value	Enthält einen individuellen Wert und muss als STRING angegeben werden.	`"Value": "253-5553-9421"`	3
Tables	Dieses Element ist ein Array und kann mehrere Tabellen-Objekte mit den folgenden Elementen besitzen.	`"Tables": []`	2
Key	Wird als Überschrift für die Tabelle verwendet.	`"Key": "My Table"`	3
Comment	Dient als Kommentar für das Datenobjekt. Der Kommentar wird im Report nicht dargestellt.	`"Comment": "My Comment"`	3
Sort Priority	Gibt die Position des Datenobjektes im Report an.	`"SortPriority": 42`	3
Value	Dieses Element ist ein Objekt und enthält die folgenden Elemente.	`"Value": {}`	3
Header	Dieses Element ist ein Array und erhält die Spaltenüberschriften der Tabelle. Die Anzahl der Spaltenüberschriften muss gleich der Spalten der Tabelle sein.	`"Header": ["Table Header 1", "Table Header 2"]`	4
DataTypes	Dieses Element ist ein Array und erhält die Datentypen der Tabellenspalte. Folgende Datentypen sind möglich: BOOL, BYTE, SBYTE, CHAR, DECIMAL, DOUBLE, FLOAT, INT, UINT, LONG, ULONG, SHORT, USHORT, STRING. Die Anzahl der Datentypen muss gleich der Spalten der Tabelle sein.	`"DataTypes": ["string", "double"]`	4
Values	Dieses Element ist ein 2D-Array und erhält die Wert der Tabelle. Ein Array in dem Array repräsentiert eine Tabellenspalte. Die Tabellenspalte muss zu dem angegebenen Datentyp passen. Die Anzahl der Arrays muss gleich der Spalten der Tabelle sein. Die Arrays sollten gleich groß sein.	`"Values": [[],[]]`	4
Images	Dieses Element ist ein Array und kann mehrere Bild-Objekte mit den folgenden Elementen besitzen.	`"Images": []`	2
Key	Wird als Überschrift für das Bild verwendet.	`"Key": "My Image"`	3
Comment	Dient als Kommentar für das Datenobjekt. Der Kommentar wird im Report nicht dargestellt.	`"Comment": "My Comment"`	3
Sort Priority	Gibt die Position des Datenobjektes im Report an.	`"SortPriority": 42`	3
Value	Enthält den Base64String mit PNG-Encoding.	`"Value": "<Base64String PNG-Encoding>"`	3
Value Pairs	Dieses Element ist ein Array und kann mehrere Key-Value-Paar-Objekte mit den folgenden Elementen besitzen.	`"ValuePairs": []`	2
Key	Wird als Überschrift für die Sammlung aus Key-Value-Paaren genutzt.	`"Key": "My Value Pair"`	3
Comment	Dient als Kommentar für das Datenobjekt. Der Kommentar wird im Report nicht dargestellt.	`"Comment": "My Comment"`	3
Sort Priority	Gibt die Position des Datenobjektes im Report an.	`"SortPriority": 42`	3
Value Pair Collection	Dient als Sammlung von Key-Value-Paaren. Dieses Element ist ein Array und kann mehrere Objekte mit den folgenden Elementen besitzen.	`"ValuePair_Collection": []`	3
Key	Enthält einen individuellen Wert und wird auf der linken Seite dargestellt.	`"Key": "Serial number"`	4
Data Type	Beschreibt den Datentyp des Elements „Value“.	`"DataType": "string"`	4

Key	Beschreibung	Beispiel	Ebene
Value	Enthält einen individuellen Wert und wird auf der rechten Seite dargestellt. Der Value muss als STRING angegeben werden.	`"Value": "253-5553-9421"`	4
Data Configurations			
Data Configurations	Über die Data Configurations können die Daten, die dynamisch von den Reporting Clients versendet werden, sortiert und individualisiert werden. Dieses Element ist ein Array und kann mehrere Objekte mit den folgenden Elementen besitzen.	`"DataConfigurations": []`	1
Key	Ist dem Data Key des Datenobjektes entsprechend. Dadurch wird sichergestellt, dass diese Datenkonfiguration auf ein eindeutiges Datenobjekt angewendet wird. (Required)	`"Key": "DataTable101"`	2
Title	Bietet die Option einen besseren Namen als Überschrift. (Optional)	`"Title": "Rotations and vibrations"`	2
SortPriority	Gibt die Position des Datenobjektes im Report an. (Required)	`"SortPriority": 42`	2
IsRequired	Gibt an, ob dieses Datenobjekt für die Generierung des Reports erforderlich ist. (siehe: TriggerOnRequiredKey)	`"IsRequired": false`	2
StoreMode	Bietet die Option, einzelne Datenobjekte auf bestimmte Art zu speichern und dementsprechend darzustellen. Aktuell werden vier unterschiedliche Store Modi unterstützt: • **Blocking**: Das erste Datenobjekt wird gespeichert und alle nachfolgenden Datenobjekte mit dem gleichen Data Key werden verworfen. • **Overwriting**: Das neu empfangene Datenobjekt ersetzt das letzte Datenobjekt mit dem gleichen Data Key. • **Appending**: Das neu empfangene Datenobjekt wird gespeichert. Die zuvor empfangenen Datenobjekte werden ebenfalls gehalten und im Report dargestellt. Bei einer Tabelle werden die Daten zusammengeführt. • **Appending_Series**: Dieser Store Modus entspricht dem Store Modus „Appending“. Jedoch werden die Datenobjekte nummeriert und in einzelne Tabellen dargestellt.	`"StoreMode": "Overwriting"`	2
RoundTo	Bietet die Option, Gleitkommazahlen auf eine bestimmte Nachkommastelle zu runden.	`"SortPriority": 4`	2
TimestampFormat	Bietet die Option, Timestamp in ein bestimmtes Format zu konvertieren. Standardmäßig ist folgende Formatierung ausgewählt: "yyyy-MM-dd HH:mm:ss.ffff" (Siehe auch: Custom date and time format strings)	`"TimestampFormat": "HH:mm:ss.ffff"`	2
IsAsc	Bietet die Option, die Sortierung einer Datentabelle festzulegen. Ist der Wert auf true, wird der erste Datensatz in der ersten Zeile dargestellt.	`"IsAsc": true`	2

Key	Beschreibung	Beispiel	Ebene
StyleClass	Bietet die Option eine benutzerdefinierte CSS-Style-Klasse zu verwenden. Die CSS-Style-Klasse muss in der CSS-Datei definiert sein und die CSS-Datei muss unter der Option StyleSheet referenziert werden.	`"StyleClass": "CustomClass"`	2
Publish Locations			
Publish Locations		`"PublishLocations": []`	1
Email Publish Location	Wird verwendet, um den Report als E-Mail weiterzuleiten. Dieses Element ist ein Array und kann mehrere Objekte mit den folgenden Elementen besitzen.	`"EmailPublishLocations": []`	2
Type	Gibt den Typ der Verteilung an.	`"Type": "File"`	3
Format	Bietet die Option, nur eine Auswahl an Formaten per E-Mail zu versenden.	`"Format": ["PDF", "HTML", "JSON"]`	3
Address	Bietet die Möglichkeit, den Report an mehrere E-Mail-Adressen zu verschicken.	`"Address": [ "m.mustermann@tester.com "]`	3
Use Bcc	Bietet die Option, die E-Mail-Adressen in BCC zu setzen.	`"UseBcc": false`	3
Text	Bietet die Option, einen spezifischen E-Mail-Text anzugeben. Dieser Text ersetzt den Standard-E-Mail-Text.	`"Text": "Hello, \n insert email text here.\n Your Reporting Service.`	3
File Publish Location	Wird verwendet, um den Report in einen bestimmten Ablageort zu kopieren. Dieses Element ist ein Array und kann mehrere Objekte mit den folgenden Elementen besitzen.	`"FilePublishLocations": []`	2
Type	Gibt den Typ der Verteilung an.	`"Type": "File"`	3
Format	Bietet die Option ,nur eine Auswahl an Formaten per E-Mail zu versenden.	`"Format": ["PDF", "HTML", "JSON"]`	3
Path	Bietet die Möglichkeit, einen Zielpfad für den Report anzugeben. Der Zielpfad darf keine Administrator-Rechte erfordern.	`"Path": "C:\\tmp\\CpyFolder"`	3

6.9.3.1.2 StyleSheet

Die Stylesheets bieten die Möglichkeit, den Report designtechnisch individuell zu gestalten. Als Vorlage dient das Stylesheet *StyleBeckhoff.css.* Dieses kann kopiert, umbenannt und angepasst werden. Der Name des Stylesheets ist anschließend in der entsprechenden Konfigurationsdatei anzugeben. Die Style Sheets müssen in dem Ordner *.\TwinCAT\Functions\Reporting-Server\Configuration\ReportStyleSheets* abgelegt werden.
Grundsätzlich lassen sich alle in einem HTML-Report enthaltenden HTML-Objekte in der benutzerdefinierten Stylesheet individualisieren.
Folgende CSS-Objekte werden häufig verwendet und bieten sich zur Individualisierung an:

CSS-Element	Beschreibung
html	Das HTML-Element bezieht sich auf den gesamten Report.
h1	Hauptüberschrift des Headers im Report
h3	Unterüberschrift des Headers im Report; Überschrift eines Datenobjektes

Folgende CSS-Klassen werden zum Design von Datenobjekten verwendet:

CSS-Klasse	Beschreibung
reporting-header	Klasse für den Header des Reports.
reaporting-header img	Klasse für das Icon oben rechts im Header.
reporting-header table	Der Header ist tabellarisch aufgebaut und lässt sich über diese Klasse anpassen.
reporting-body	Klasse für den Inhalt des Reports.
reporting-body caption	Die Überschriften der Datenobjekte befinden sich in einem Caption-Element.
reporting-body valuetable	Klasse für eine Datentabelle im Report.
reporting-body valuepair	Klasse für eine Sammlung aus Key-Value-Paaren im Report.
reporting-body infotable	Klasse für die zusätzlichen Information und Analyse- bzw. Aufnahmeinformationen aus dem manuellen Report.
reporting-body figure	Klasse für Bilder, wie zum Beispiel Charts aus dem Scope im Report.
reporting-body img	Diese Klasse enthält das eigentliche Bild. Es befindet sich in dem figure-Element.

6.9.3.2 Report-Format

Die Reports werden in den folgenden Formaten generiert.

1. PDF
2. HTML
3. JSON

6.9.3.3 Netzwerk Konfiguration

Ein möglicher Anwendungsfall könnte sein, dass der Reporting Server und der Reporting Client auf unterschiedlichen Systemen ausgeführt werden. Der Reporting Server und der Reporting Client müssen sich im selben Netzwerk befinden. Um diesen Anwendungsfall abbilden zu können, muss eine JSON-Datei erweitert werden. Die Datei sollte unter folgenden Pfad liegen und den entsprechenden Namen haben: *.\TwinCAT\Functions\Reporting-Server\Configuration\Network\ReportingNetworkConfiguration.json*
Eine Netzwerk-Konfiguration sieht folgendermaßen aus:

Name	Beschreibung	Datentyp
ConfigIds	Report Namen der entsprechenden Reports	List<string>
AmsNetId	AmsNetId des Zielsystem. Auf dem Zielsystem wird der Reporting Server ausgeführt.	string

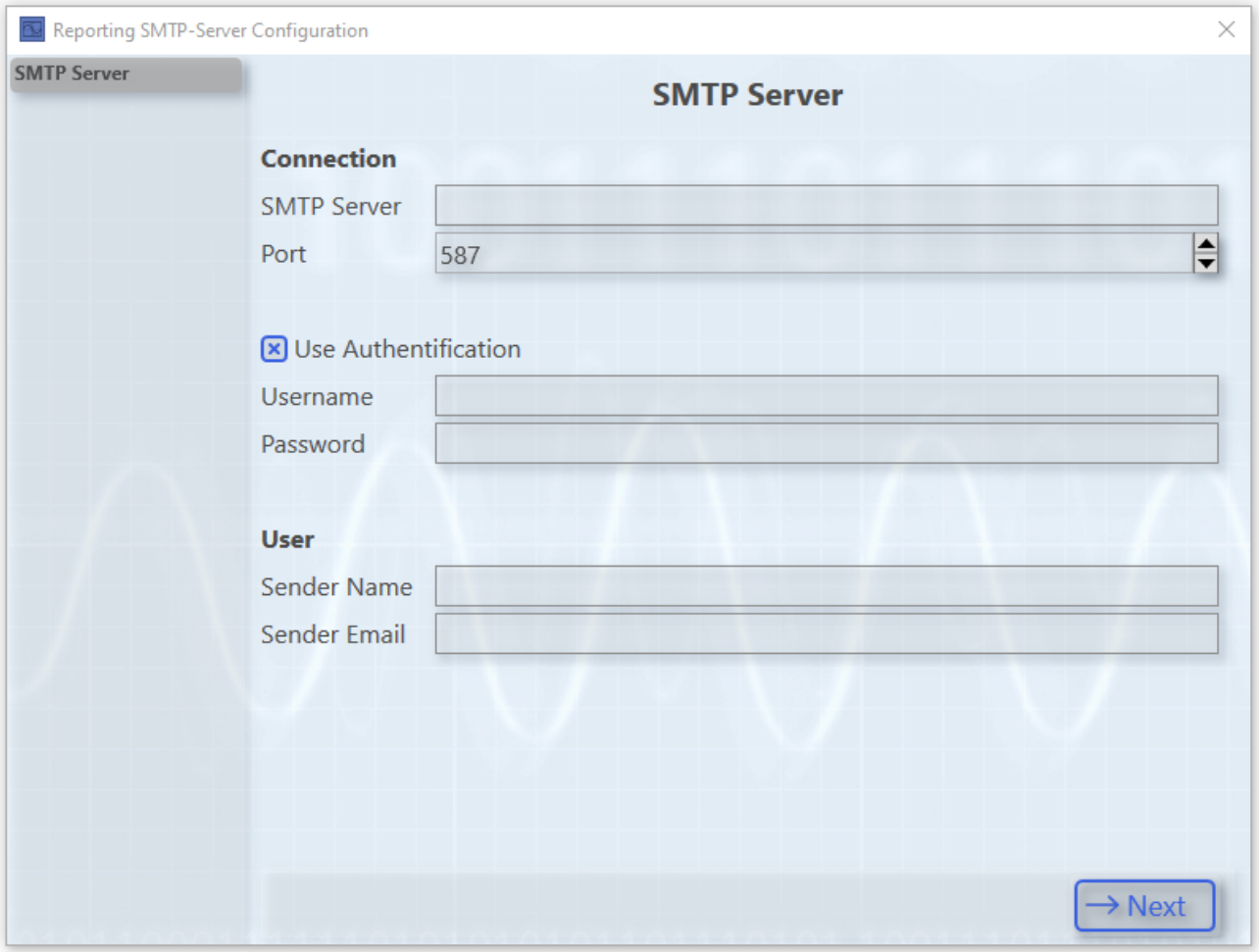
6.9.3.4 E-Mail-Client-Konfiguration

Ein TwinCAT Reporting Server kann mit einem SMTP-Server verbunden werden. Der SMTP-Server ist von dem Anwender aufzusetzen und ist kein Teil des TwinCAT Reportings. Der TwinCAT Reporting Server sendet die konfigurierte E-Mail mit dem Report an den SMTP-Server. Dieser leitet die E-Mail an den Empfänger weiter.

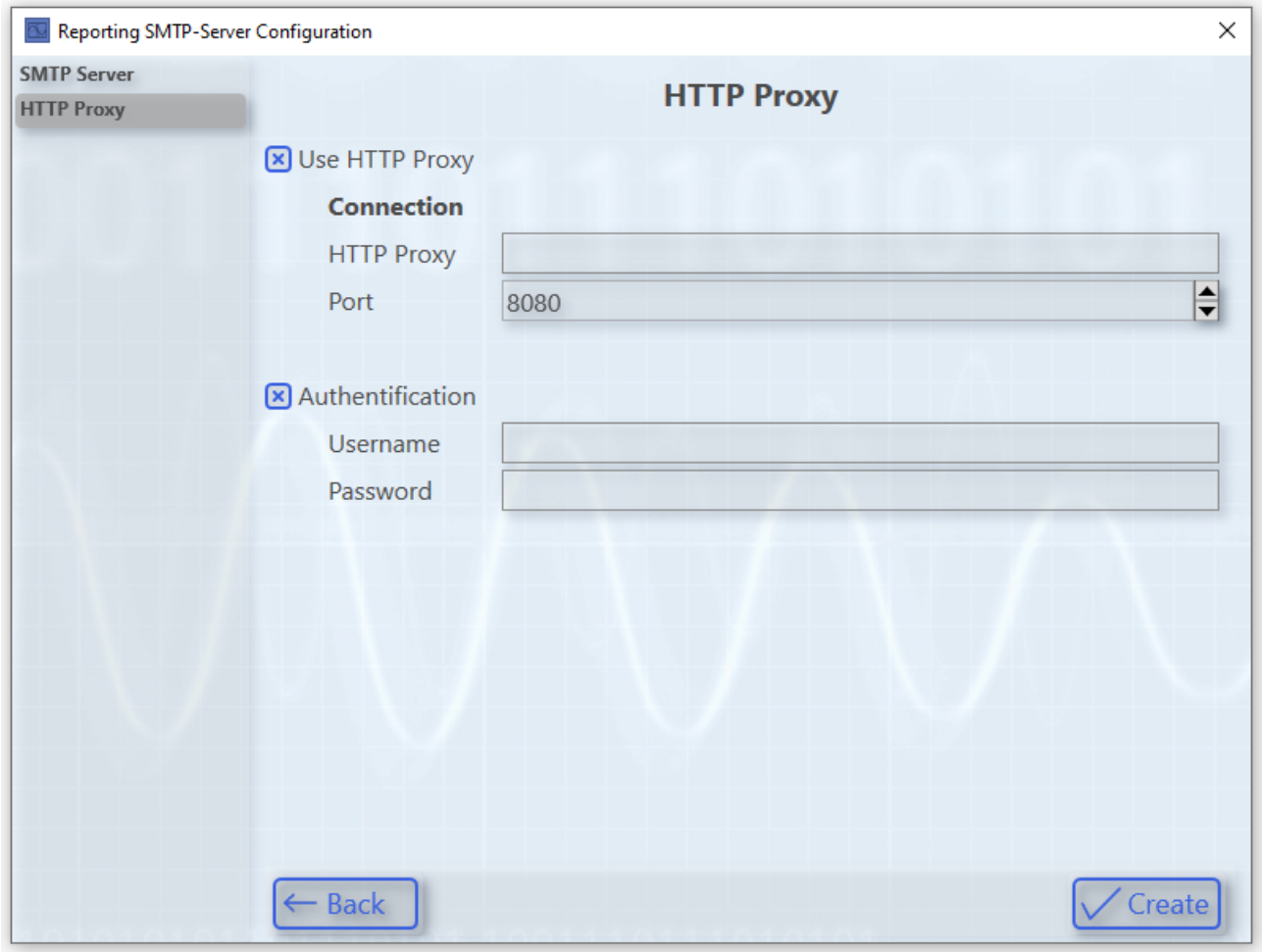
Zum Konfigurieren der Verbindung zu einem SMTP-Server steht ein Wizard zur Verfügung. Der Wizard lässt sich über zwei Möglichkeiten öffnen.

1. **TwinCAT > Analytics > Reporting Email**
2. **Scope > Reporting Email**

Im ersten Dialog können Sie eine Verbindung zum SMPT-Server konfigurieren.



Der zweite Dialog bietet die Möglichkeit, eine Verbindung zu einem HTTP-Proxy zu konfigurieren. Diese Option kann aktiviert werden, falls die Architektur des Anwenders dies voraussetzt.



Nach dem Klick auf den Button **Create** wird die Konfiguration in einer entsprechenden Datei gespeichert.

6.9.3.5 Signierung

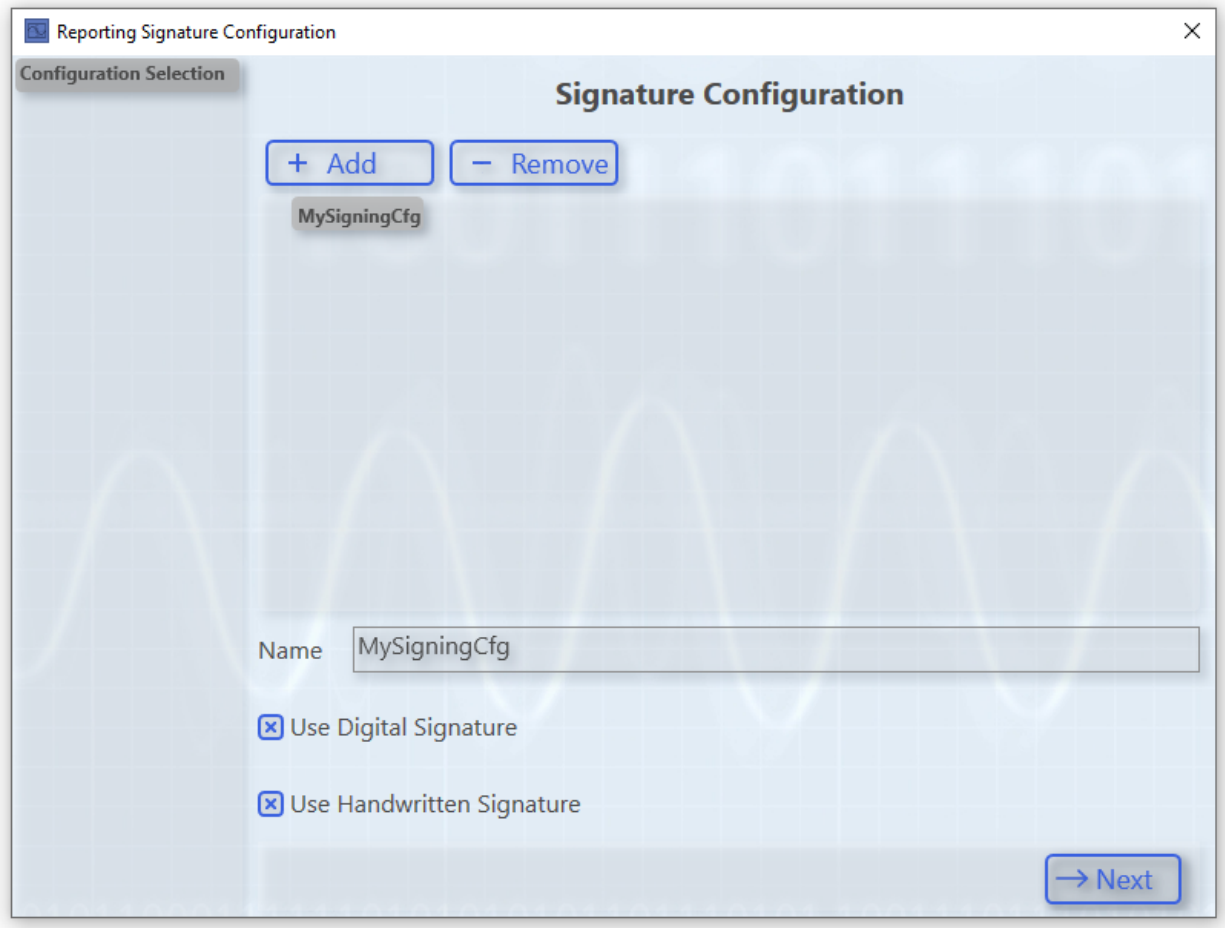
Um Reports zu verifizieren, können in dem TwinCAT Reporting PDF-Reports signiert werden. Es werden digitale und handschriftliche Signaturen unterstützt. Der Wizard zur Konfiguration der Signaturen kann über folgende Wege geöffnet werden.

- **TwinCAT > Analytics > Reporting > Reporting Signatures**
- **Scope > Reporting > Reporting Signatures**

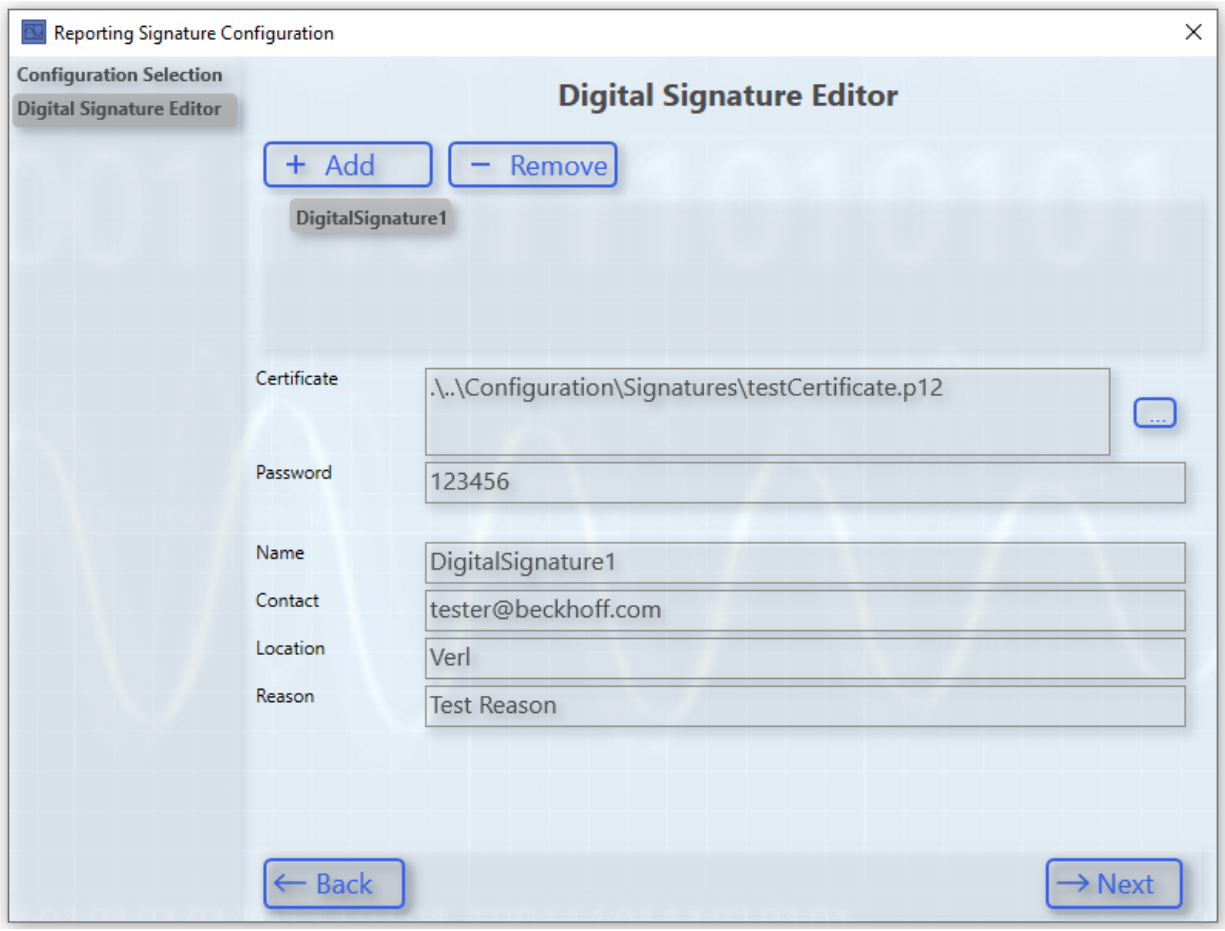
✓ Der Wizard bietet die Möglichkeit mehrere Signatur-Konfigurationen zu erstellen, die anschließend in mehreren Report-Konfigurationen wiederverwendet werden können.

1. Der Name dieser Konfiguration muss in der Konfigurationsdatei über den Key „SignatureSettingsFile“ referenziert werden (siehe Konfigurationsdatei [▶ 528]).
   Über die beiden Checkboxen können Sie auswählen, ob die konfigurierten Signaturen, die auf den

nächsten Seiten angezeigt werden, verwendet werden sollen.
Mit einem Klick auf den Button **Next** können Sie die selektierte Konfiguration bearbeiten.

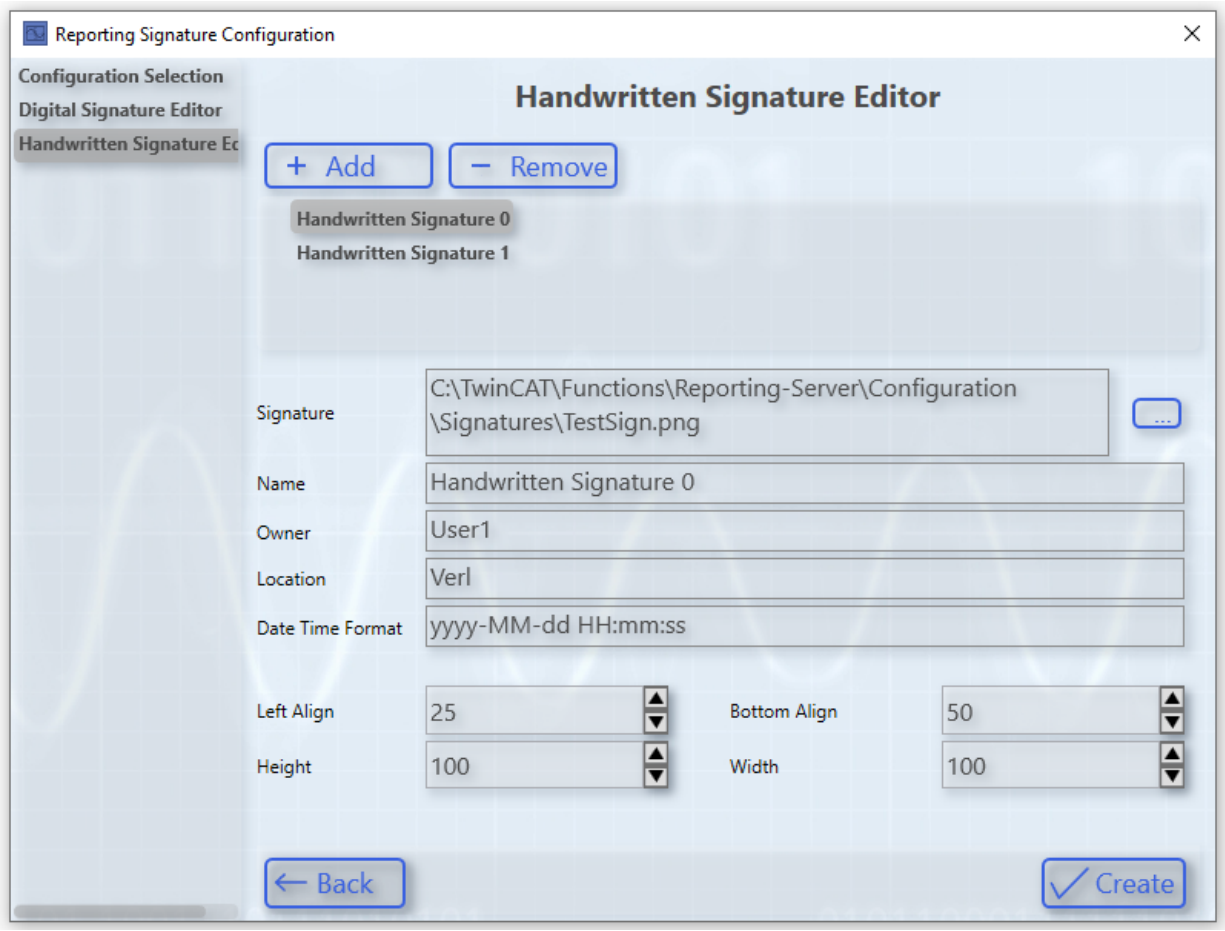


2. Hier können Sie eine digitale Signatur definieren.



⇨ Pfx- und p12-Zertifikate werden vom TwinCAT Reporting unterstützt.

3. Hier können Sie eine handschriftliche Signatur definieren.



⇨ Wenn Sie handschriftliche Signaturen verwenden, wird eine zusätzliche Signaturseite in den Report eingefügt.

4. Über das Left- und Bottom-Alignment können die Signaturen positioniert werden. Der Ursprung ist unten links auf der Signatur-Seite.
   Auf der folgenden Abbildung ist eine exemplarische handschriftliche Signatur zu sehen:

⇨ Die Information der Signatur-Konfiguration wird in dem Reporting-Server Ordner im Klartext in einer Datei gespeichert.

6.9.3.6 Lizenzierung

Das TwinCAT Reporting ist als Vollversion und als 7-Tage-Testversion lizenzierbar.
Die folgenden Produkte enthalten die TwinCAT Reporting Lizenz.

Produktnummer	Produktname
TE1300	TC3 Scope View Professional
TF3300	TC3 Scope Server
TE3500	TC3 Analytics Workbench
TE3520	TC3 Analytics Service Tool

Folgende Features sind mit der Vollversion bzw. mit der 7-Tage-Testversion im TwinCAT Reporting Server verfügbar.

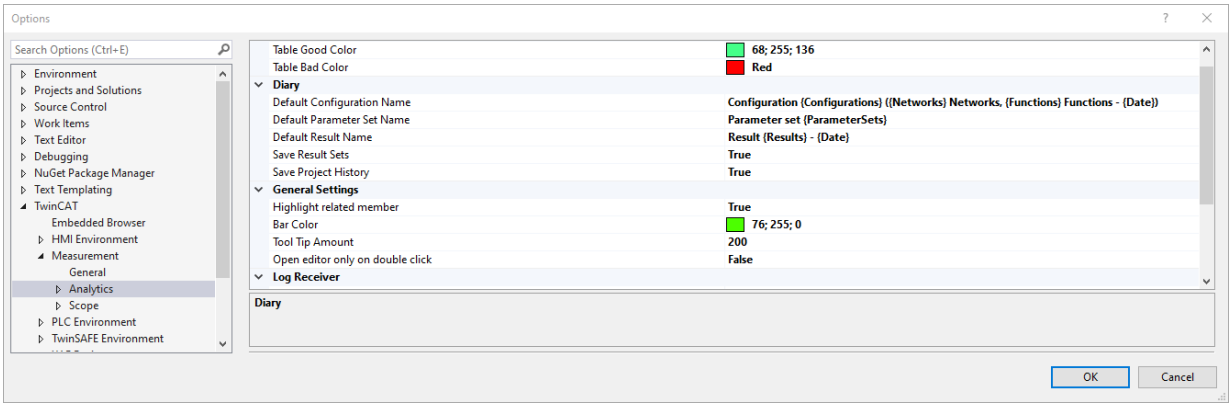
Features	TwinCAT Reporting Server	
	7-Days-Trial	Full License
General:		
Reports	✔ (Max. 1)	✔ (unlimited)
Digitale Signaturen	✘	✔
Handschriftliche Signaturen	✘	✔

Einschränkungen der Testversion bei den Reporting Clients entnehmen Sie bitte der Produktübersichtsseite.

6.10 Vergleich von Analyseergebnissen

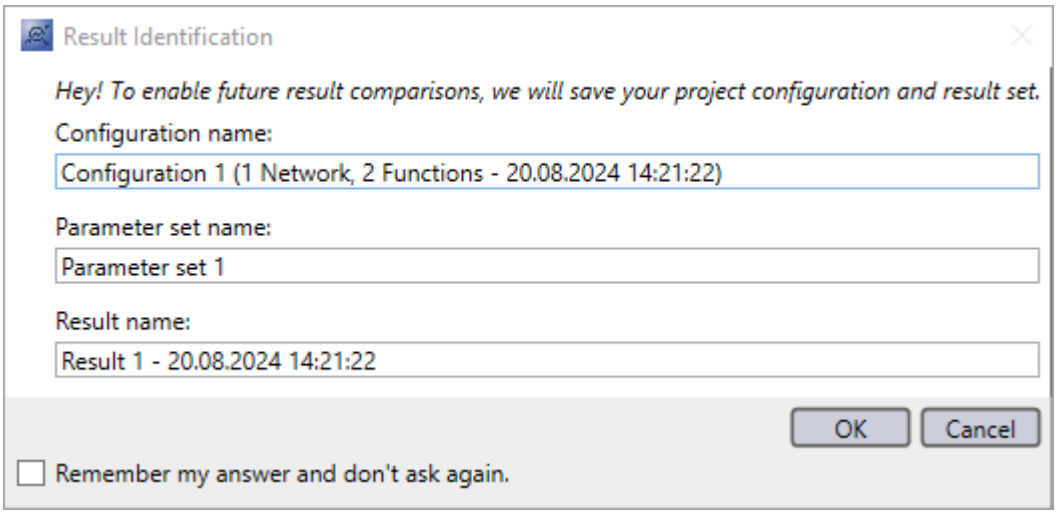
Verschiedene Analysedurchläufe können mit dem Analytics Diary verglichen werden. Das Analytics Diary bietet einige Vorteile in der explorativen Datenanalyse und der Identifizierung eines geeigneten Analyseverfahrens inklusive der Parameter von Algorithmen.

1. Standardmäßig werden alle Analyseergebnisse inklusive der Projekt-Konfiguration mit jedem Analysedurchlauf gespeichert. Diese Einstellung können Sie ändern unter **Tools > Options > TwinCAT > Measurement > Analytics > Diary** .

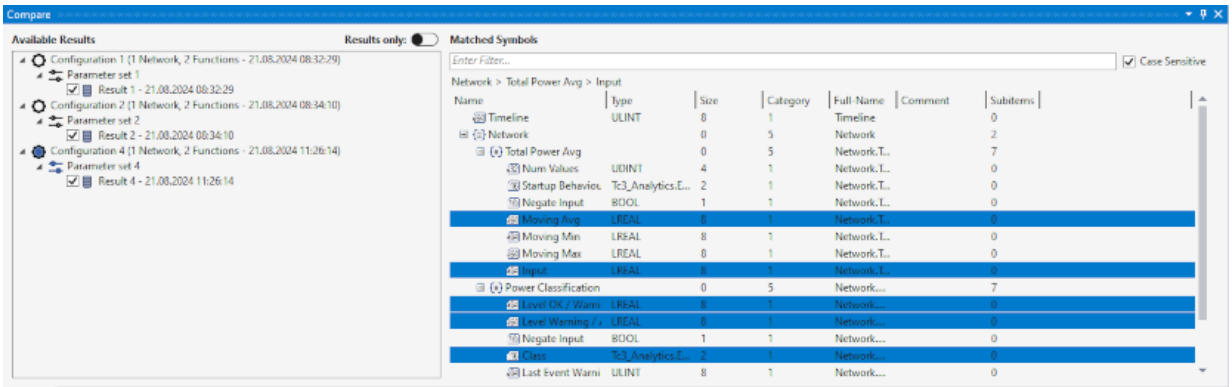


2. Mit dem Starten einer Analyse erscheint ein Fenster vom Diary. In diesem können Sie die Namen der zugrundeliegenden Analysekonfiguration, der Parameterkonfiguration und der Analyseergebnisse einstellen. Soweit sich nichts am Projekt geändert hat, werden nur die Daten gespeichert. Ebenfalls wird

keine Analysekonfiguration erzeugt, wenn sich lediglich die Konfiguration an Parametern ändert (siehe dazu auch Globale Parameter [▶ 79]).

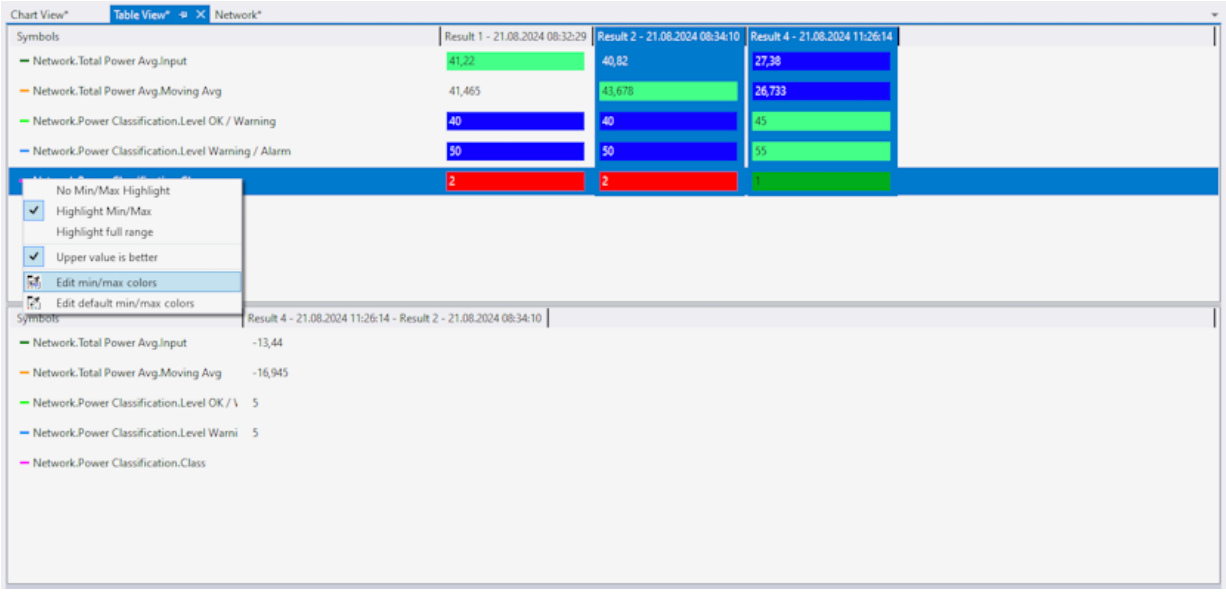


3. Diese Einträge werden anschließend im Fenster **Compare** in einer Baumstruktur erscheinen, welches Sie über einen Klick aus dem **Solution Explorer** öffnen. Dies bildet die Historie an Analysedurchläufen ab. Sie können mehrere Ergebnisdatensätze sowie übereinstimmende Symbole auswählen, die anschließend in den Fenstern **Table View** und **Chart View** betrachtet werden können. Die Auswahl der Datensätze und Symbole ist also mit den anderen beiden Fenstern synchronisiert. Standardmäßig wird immer der letzte Datensatz mit ausgewählt.

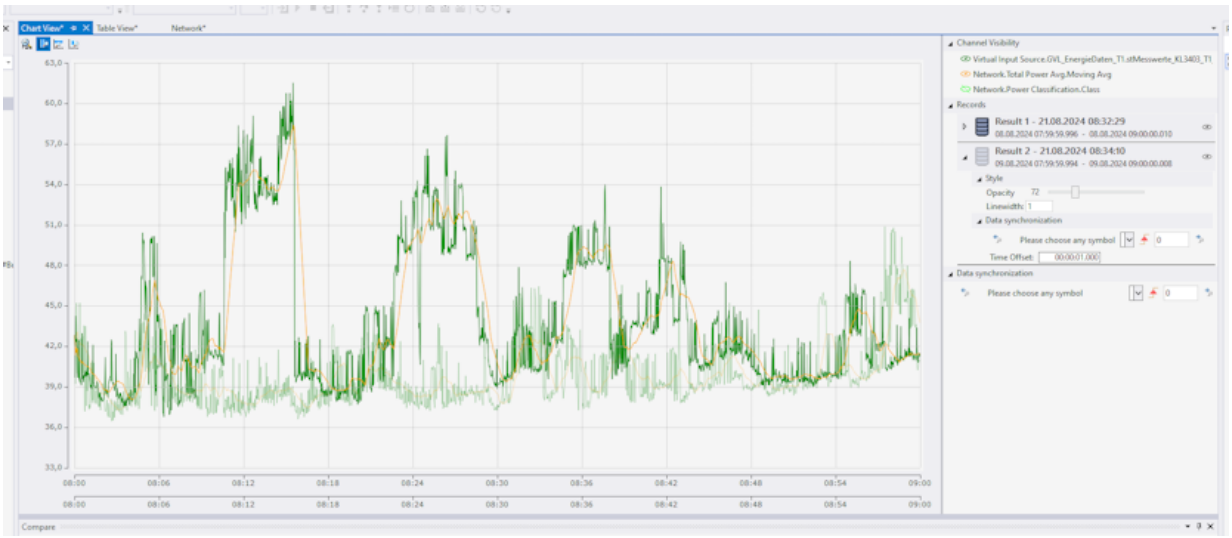


4. Die Fenster **Table View** und **Chart View** öffnen Sie über einen Klick aus dem **Solution Explorer**. In der **Table View** sind zwei Tabellen abgebildet. In der ersten wird das letzte Ergebnis von allen ausgewählten Datensätzen und Symbolen angezeigt. Dabei werden die Datensätze spaltenweise und die Symbole zeilenweise aufgelistet. Über eine optionale farbliche Darstellung können die Unterschiede eines Symbols visuell hervorgehoben werden. Des Weiteren können die Differenzen der Symbole in einer

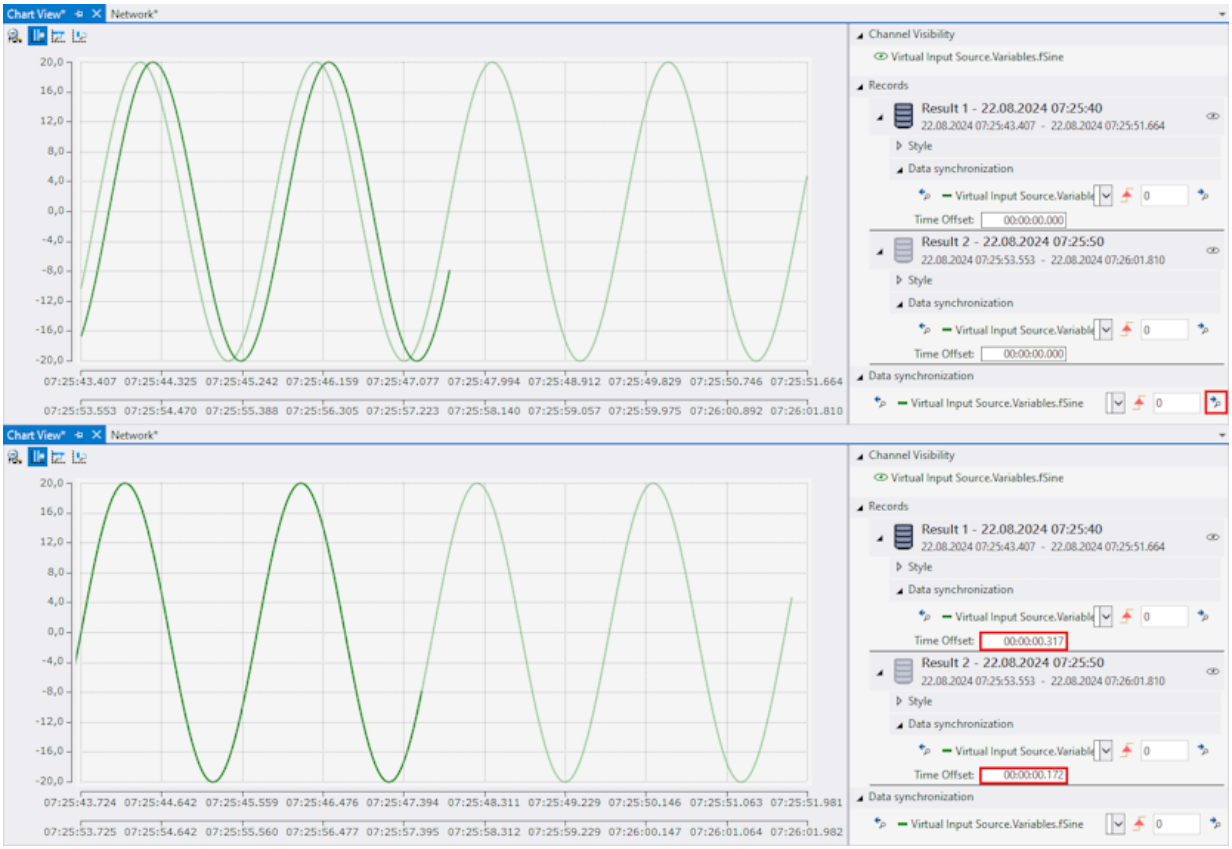
zweiten Tabelle aufgezeigt werden. Dafür müssen mindestens zwei Spalten bzw. Datensätze aus der ersten Tabelle selektiert sein.



5. Das *Chart View* dient dazu, Ergebnisdatensätze als Zeitreihen gegenüberzustellen. Hierbei werden den Symbolen unterschiedliche Farben zugewiesen. Die Symbole erscheinen entsprechend der Anzahl der ausgewählten Datensätze, wobei die Deckkraft der Linien jeweils abgeschwächt wird. Die Deckkraft und die Liniendicke zur Unterscheidung der Datensätze können Sie manuell anpassen. Für die Betrachtung eines näheren Bereichs können Sie äquivalent zum Scope ein Zoom anwenden.

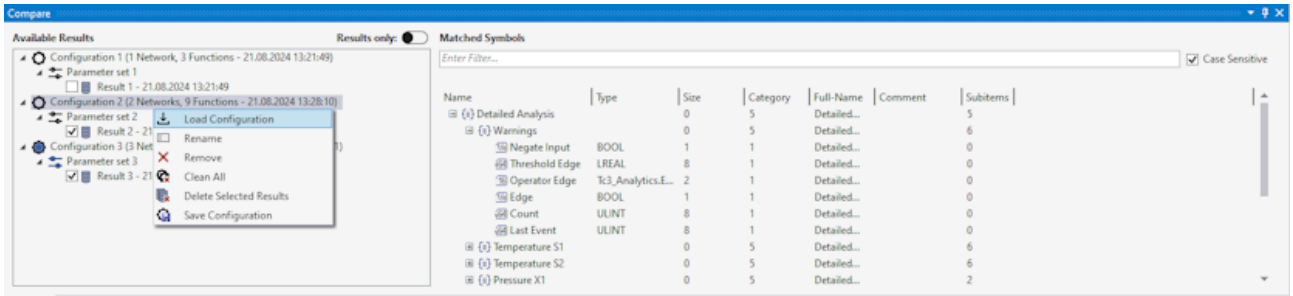


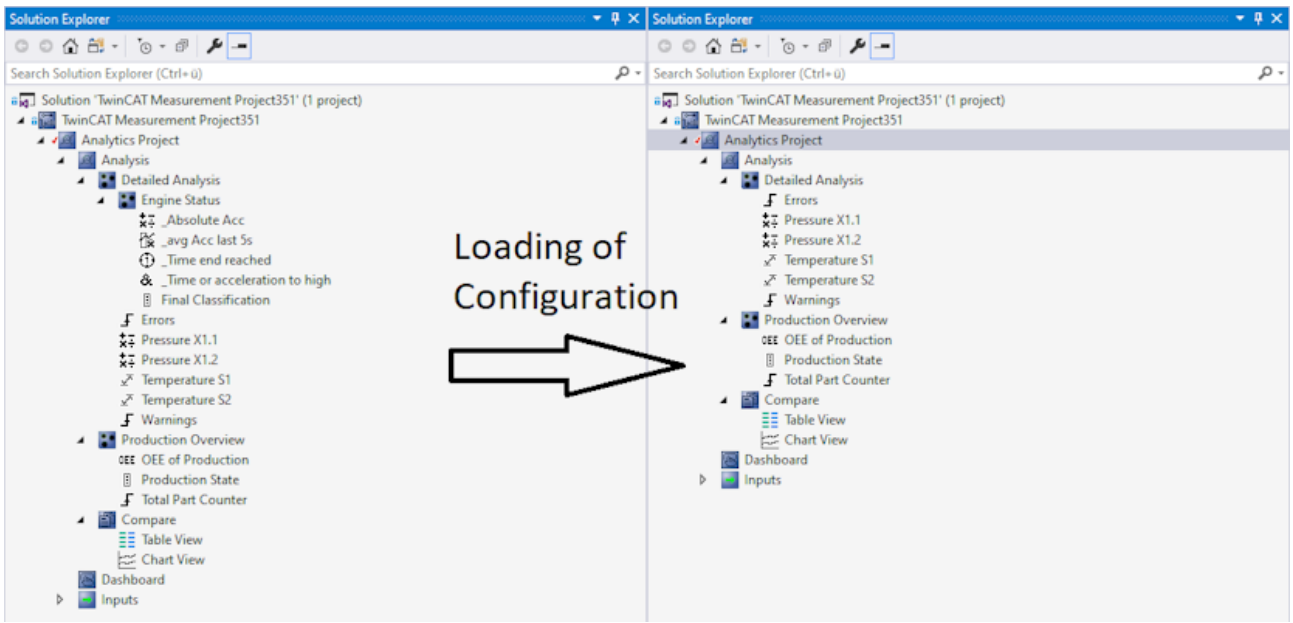
6. Wenn die Startzeitpunkte der Datensätze nicht übereinstimmen, können Sie diese für einen direkten Vergleich anpassen. Dies kann entweder durch manuelle Eingabe des Zeitversatzes oder durch eine automatische Flankensuche erfolgen.



6.10.1 Laden und Speichern von Projektkonfigurationen

Innerhalb des *Compare* Fensters wird die Historie der Analysedurchläufe aufgelistet. Zu einer Konfiguration ist der entsprechende Projektstand gespeichert, der geladen werden kann. Um die Konfiguration zu laden, klicken Sie mit der rechten Maustaste auf die gewünschte Konfiguration (bzw. Parametersatz) und wählen Sie im Kontextmenü die Option *Load Configuration*. Damit wird der alte Projektstand wiederhergestellt. Anschließend wird das Symbol der Konfiguration blau, was auf die zuletzt geladene Konfiguration hinweist.





Neben der automatischen Speicherung einer Konfiguration beim Starten der Analyse kann diese auch manuell gespeichert werden. Dafür muss lediglich im Kontextmenü der Eintrag *Save Configuration* ausgewählt werden.

7 API

Mit der nachfolgend dokumentierten TwinCAT Analytics Data Exchange API können Daten in den Analytics Workflow eingespeist bzw aus diesem herausgeholt werden. Somit können Sie externe Quellen anbinden oder alternativ die Daten aus Analytics in anderen Softwarekomponenten weiterverwenden.

7.1 Data Exchange API

In allen unterstützten Programmier- bzw. Skript-Sprachen lassen sich mit der Analytics Data Exchange API eigene Analytics Logger oder Analytics Runtimes nachbauen.

Ein Logger ermöglicht es, Daten aus Fremdquellen in Analytics zu speichern oder auszuwerten. Eine Runtime wird programmiert, um die Analytics Daten in anderen Systemen zu verwenden. Je nach Richtung wird die entsprechende Lizenz fällig. Beim Logger ist es die TF3500 Analytics Logger Lizenz. Bei einer Runtime-Umsetzung ist es TF3551 Analytics Runtime Base oder alternativ die TF3550 Analytics Runtime. Sollten diese Produkte bereits auf den Systemen eingesetzt werden, ist keine weitere Lizenz für eine eigene programmatische Umsetzung erforderlich.

7.1.1 .NET

Installation

Die folgenden zwei Komponenten müssen für die Verwendung der Analytics Data Exchange API installiert sein:

1. Microsoft Visual C++ Redistributable 2019 (x64/ x86) oder höher. Fehlt dieses Paket treten Fehler bei der Verarbeitung von Symbolen auf.
2. TwinCAT 3 ADS

NuGet-Pakete

Die Analytics Data Exchange API wird als NuGet-Paket ausgeliefert. Es stehen drei verschiedene Pakete zur Verfügung:

1. Beckhoff.TwinCAT.Analytics.DataExchange.Core (.Net-Core 3.1 oder .Net-5 und höher)
2. Beckhoff.TwinCAT.Analytics.DataExchange.Framework (.Net-Framework 4.5.2 oder höher)
3. Beckhoff.TwinCAT.Analytics.DataExchange (Multi-Target Projekte)

7.1.1.1 Abbildung der Logger Funktion

Diese Dokumentation beschreibt, wie Daten mithilfe der Analytics Data Exchange API im Analytics Format versendet oder gespeichert werden können.

Lizenz

Für das Schreiben von Daten im TwinCAT-Analytics-Format wird eine Lizenz für den TwinCAT 3 Analytics Logger benötigt. Die Data Exchange API kann mit einer 7-Tage Lizenz getestet werden.

Erzeugen und Konfigurieren einer Datenablage:

Zum Versenden oder Speichern von Daten im Analytics Format wird im ersten Schritt ein Builder erzeugt, um die Datenablage zu konfigurieren:

```
ISinkFactory factory = new SinkFactory();

IFileSinkBuilder fileBuilder = factory.ToFile("C:\Data\Demo");

IMqttSinkBuilder mqttBuilder = factory.ToMessageBroker("127.0.0.1", 1883);
```

Es stehen zurzeit zwei verschiedene Builder zur Verfügung, um entweder in einer Datei zu schreiben oder MQTT-Daten zu verschicken. Die Builder werden über eine Verkettung von Methodenaufrufen konfiguriert:

```
mqttBuilder.WithSymbol("Symbol1", TypeCode.Double).WithCompressionWidth(8)
```

Folgende Methoden stehen zur Konfiguration der Symbole zur Verfügung:

AllgemeineParameter	Beschreibung
WithCompression(…)	Legt die Komprimierungsmethode für die Datensenke fest.
WithCompressionWidth(..)	Die Vergrößerung der CompressionWidth führt zu einer gegebenenfalls geringeren Komprimierung aber schnelleren Verarbeitung. Werte unter 3 werden nicht empfohlen. Standard: 8
WithMaxSamplesPerChunk(…)	Maximale Anzahl an Daten in einer MQTT-Message oder in einer Datei. Standard: 32 bei MQTT, 500 bei Datei
WithMaxChunkSize(…)	Maximale Datengröße einer MQTT-Message oder einer Datei. Standard: Nicht begrenzt
WithCycleTime(…)	Zykluszeit der Daten: Standard 10ms
Symbole:	
WithSymbol(…)	Fügt ein Symbol hinzu. Optional kann eine Arraylänge angegeben werden.
WithSymbols(Action)	Ermöglicht es mehrere Symbole gleichzeitig hinzuzufügen.
WithSymbol<T>	Fügt ein Symbol des generischen Datentyps hinzu. Information, welche Datentypen erlaubt sind, finden Sie im Abschnitt Generische Datentypen [▶ 545].
Nur MQTT:	
ToTopic(…)	Topic, über welches die Daten kommuniziert werden.
Nur Datei:	
WithIndexing(…)	Abstand zwischen zwei indexierten Datenpunkten bei Verwendung von Komprimierung. Indexierung erhöht die Performance beim Auslesen von Datenpunkten, aber vergrößert die Speichernutzung. 0 deaktiviert Indexierung und ist Standard. Zur Aktivierung sollte der Werte mindestens auf 100 gestellt werden.

Spezielle Parameter für die MQTT-Kommunikation werden über folgende Methoden konfiguriert:

WithTLS(…)	Konfiguriert die Zertifikate zur Verwendung von TLS
WithCredentials(…)	Benutzername und Passwort für den Zugriff auf den Message-Broker
WithTcpBufferSize(…)	Stellt die Größe des Puffers für die TCP-Verbindung zum Message Broker ein. Standard ist 32kB. Wenn sehr große Messages versendet werden, kann dieser Wert erhöht werden.
WithKeepAlivePeriod(…)	Einstellen der KeepAlive Zeit für die MQTT-Verbindung. (Standard 60s)
WithCommunicationTimeout(…)	Timeout für die TCP-Kommunikation zum Message Broker. (Standard 3s)
ToTopic(…) / FromTopic(…)	Topic über das die Daten kommuniziert werden.

Die Konfiguration einer Datensenke wird über den Aufruf der Build-Methode abgeschlossen, welche die konfigurierte Datensenke zurückgibt. Die Datensenke stellt folgende Methoden zur Verfügung. Wird die Datensenke nicht weiterverwendet, ist es notwendig, die Dispose-Methode aufzurufen, um alle Ressourcen wieder freizugeben.

InitWrite()	Initialisiert die MQTT-Verbindung oder öffnet die zu schreibende Datei. Wird auch automatisch beim ersten Schreiben von Werten aufgerufen.
Write(…)	Fügt einen neuen Wert zum Puffer hinzu.
Flush()	Sendet die im Puffer gespeicherten Daten. Wird automatisch von Write aufgerufen. Wenn MaxSamples oder MaxSize überschritten ist.
CompleteWrite()	Sendet die im Puffer gespeicherten Daten und schließt alle offenen MQTT-Verbindungen oder Dateizugriffe.
Dispose()	Schließt alle offenen MQTT-Verbindungen oder Dateizugriffe, ohne den Puffer zu versenden und gibt alle Ressourcen frei. Die Senke kann danach nicht mehr verwendet werden.

Generische Datentypen:

Die Analytics Data-Exchange-API unterstützt nur „unmanaged structs“ als generische Datentypen. Dies bedeutet folgende Einschränkungen

1. Innerhalb der Strukturen dürfen nur Strukturen oder Basis-Datentypen verwendet werden.
2. Arrays innerhalb der Strukturen sind nur mit „unsafe“ Code möglich. (Fixed-Array-Size)
3. Es können nur Arrays von Basis-Datentypen verwendet werden. Arrays von Strukturen sind nicht möglich.

Für die Strukturen kann außerdem nur ein sequentielles oder explizites Layout verwendet werden (siehe https://docs.microsoft.com/de-de/dotnet/api/system.runtime.interopservices.structlayoutattribute?view=net-6.0)

1. Beim Schreiben von Daten sollte ein explizites Layout verwendet werden.
2. Beim Schreiben von Daten darf das Layout kein Padding enthalten (Pack = 1)
3. Beim Lesen von Daten muss das Padding der Datenquelle entsprechen (https://infosys.beckhoff.com/index.php?content=../content/1031/tc3_plc_intro/3539428491.html&id=)

Fehlermeldungen

Treten beim Verwenden der Data Exchange API Fehler auf, werden diese per Ausnahme an den Benutzer weitergegeben. Folgende Ausnahmen werden verwendet:

Klassenname	Oberklasse	Beschreibung
DataExchangeException	Exception	Oberklasse für alle weiteren Ausnahmen innerhalb der API.
InvalidSymbolDefinitionException	DataExchangeException	Die konfigurierten Symbole sind nicht korrekt.
SymbolNotFoundException	InvalidSymbolDefinitionException	Ein konfiguriertes Symbol konnte im Datenstrom nicht gefunden werden.
InvalidSymbolNameException	InvalidSymbolDefinitionException	Der eingegebene Symbolname ist nicht gültig.
InvalidSymbolOffsetException	InvalidSymbolDefinitionException	Der eingegebene Symboloffset ist kleiner als 0.
InvalidSymbolByteSizeException	InvalidSymbolDefinitionException	Die eingegebene Größe in Bytes ist kleiner oder gleich 0.
InvalidSymbolTypeCodeException	InvalidSymbolDefinitionException	Der eingegebene TypeCode für ein Symbol ist nicht gültig.
InvalidSymbolRegexPatternException	InvalidSymbolDefinitionException	Das eingegebene Regex-Pattern für eine Symbol ist nicht gültig.
InvalidSymbolFilterException	InvalidSymbolDefinitionException	Der eingegebene SymbolFilter für eine Regex-Pattern ist nicht gültig.
DataExchangeLicenseException	DataExchangeException	Es konnte keine gültige Lizenz für die API gefunden werden oder TwinCAT ist nicht im Run-Mode.
DataExchangeTimeoutException	DataExchangeException	Eine Timeout-Zeit wurde überschritten.
QueueFullException	DataExchangeException	Eine Warteschlange für die Abarbeitung von Daten ist voll, da die Abarbeitung zu langsam war.
FileSourceNotFoundException	DataExchangeExeption	Im angegebenen Ordner konnte kein Analytics-File gefunden werden.
ValueExtractException	DataExchangeException	Beim Auslesen der Werte aus dem Datenstrom ist ein Fehler aufgetreten. Genauere Informationen gibt es in der inneren Ausnahme.
DataExchangeMQTTException	DataExchangeException	Ein Fehler ist bei der MQTT-Verbindung aufgetreten.
BrokerConnectException	DataExchangeMQTTException	Eine Verbindung zum Broker konnte nicht aufgebaut werden.
TopicNotFoundException	DataExchangeMQTTException	Das angegebene Topic wurde am Broker nicht gefunden.
InvalidMqttStreamFormat	DataExchangeMQTTException	Das angegebene Stream-Format ist nicht korrekt.

Beispiele:

Beispiele für die Verwendung der Analytics Data Exchange API finden Sie hier:

https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/12970252171.zip

7.1.1.2 Abbildung der Runtime Funktion

Diese Dokumentation beschreibt, wie Daten mithilfe der Analytics Data Exchange API aus dem Analytics-Format ausgelesen werden können.

Lizenz

Für das Lesen von Daten aus dem TwinCAT-Analytics-Format wird eine Lizenz für die „TwinCAT 3 Analytics Runtime“ oder „TwinCAT 3 Analytics Runtime Base“ benötigt. Die Data Exchange API kann mit einer 7-Tage Lizenz getestet werden.

Erzeugen und Konfigurieren einer Datenquelle:

Zum Auslesen von Daten aus dem Analytics Format wird im ersten Schritt ein Builder erzeugt und konfiguriert.

```
ISourceFactory factory = new SourceFactory();

IFileSourceBuilder fileBuilder = factory.FromFile("C:\Data\Demo");

IMqttLiveSourceBuilder mqttBuilder = factory.FromLiveStream("127.0.0.1", 1883);

IStorageProviderSourceBuilder FromStorageProvider("127.0.0.1", 1883);
```

Es stehen zurzeit drei verschiedene Builder zur Verfügung:

1. IFileSourceBuilder: Daten aus einer Datei.
2. IMqttLiveSourceBuilder: Daten von einem Datenstrom eines Analytics Loggers.
3. IStorageProviderSourceBuilder: Aufgezeichnete Daten eines Analytics Storage Providers.

Der Builder wird über eine Verkettung von Methodenaufrufen parametriert. Dabei stehen folgende Methoden zur Verfügung:

Symbole	Beschreibung
WithSymbol(…)	Fügt ein zusätzliches Symbol zum Auslesen hinzu. Zur Verfügung stehende Symbole können über den TwinCAT Target Browser ausgelesen werden.
WithSymbols(…)	Ermöglicht es mehrere Symbole gleichzeitig hinzuzufügen.
WithSymbols(pattern, …)	Fügt mehrere Symbole anhand eines Regex-Ausdrucks hinzu. Zusätzlich können RegexOptions und ein SymbolFilter angegeben werden. SymbolFilter: Include Parent: Verwendet das Obersymbol, auch wenn Subsymbole zutreffen. IncludeChilds: Verwendet die Untersymbole, auch wenn das Obersymbol zutrifft. IncludeChilds oder InlcudeParent müssen gesetzt werden.
WithSymbol<T>()	Fügt ein Symbol des generischen Datentyps hinzu. Information, welche Datentypen erlaubt sind, finden Sie im Abschnitt Generische Datentypen [▶ 550].
Nur MQTT:	
FromTopic(…)	MQTT-Topic von dem die Daten gelesen werden sollen.
WithDataTimeOut(…)	Timeout zwischen zwei MQTT-Nachrichten (Standard 500ms). -1ms bedeutet ein unendliches Timeout.
WithSymbolTimeOut(…)	Timeout zum Empfangen der Symbol-Information vom Message Broker (Standard 1s). -1ms bedeutet ein unendliches Timeout.
WithSortQueueSize(…)	Einstellung der Größe des Puffers für eingehende Nachrichten. Der Puffer wird zusätzlich zum Sortieren der Nachrichten verwendet. (Standard: Min: 5 Max: 25)
WithStreamFormat(…)	Legt fest, welches Format die empfangenen Daten haben. Wenn nichts angegeben ist wird das Format automatisch erkannt. Zur automatischen Erkennung muss der Datenstrom beim Start online sein.
Storage Provider und Datei:	
From(…)	Startzeit ab der die Daten abgerufen werden sollen.
To(…)	Endzeit bis zu der die Daten abgerufen werden sollen.
Nur StorageProvider	
FromStorageProivder(…)	Gibt an, von welchem Storage Provider am verwendeten Broker die Daten abgerufen werden sollen.
FromHistoricalStream(…)	Aus dem angegeben Topic wird die ProviderId und das MainTopic ausgelesen, welche andernfalls über die Methode FromStorageProvider übergeben werden müssen.
FromLayout(…)	Layout des Historischen Datenstroms.
FromOriginSystem(…)	System-Id von dem die Daten ursprünglich gesendet wurden.
FromOriginStreamTopic (…)	Topic über das die Daten ursprünglich versendet wurden.
FromStreamAlias(…)	Alias unter dem der Datenstrom aufgezeichnet und im TargetBrowser angezeigt wird.

Die Methoden FromLayout, FromOriginSystem, FromOriginStreamTopic, FromStreamAlias, werden zur Identifikation des korrekten Datensatzes verwendet. Es nicht zwingend notwendig alle zu parametrieren, aber es wird empfohlen möglichst viele anzugeben, um eine Verwechslung von Datensätzen auszuschließen. Am einfachsten können diese Informationen aus dem Target Browser ausgelesen werden. Ziehen Sie dazu ein Symbol des Datensatzes in eine Texteditor (z. B. Notepad ++). In dem XML finden Sie die notwendigen Informationen im Bereich Target-Info.

Spezielle Parameter für die MQTT-Kommunikation werden über folgende Methoden konfiguriert:

WithTLS(…)	Konfiguriert die Zertifikate zur Verwendung von TLS
WithCredentials(…)	Benutzername und Passwort für den Zugriff auf den Message-Broker
WithTcpBufferSize(…)	Stellt die Größe des Puffers für die TCP-Verbindung zum Message Broker ein. Standard ist 32kB. Wenn sehr große Messages versendet werden, kann dieser Wert erhöht werden.
WithKeepAlivePeriod(…)	Einstellen der KeepAlive Zeit für die MQTT-Verbindung. (Standard 60s)
WithCommunicationTimeout(…)	Timeout für die TCP-Kommunikation zum Message Broker. (Standard 3s)
ToTopic(…) / FromTopic(…)	Topic über das die Daten kommuniziert werden.

Die Konfiguration einer Datenquelle wird über den Aufruf der Build-Methode abgeschlossen, welche die konfigurierte Datenquelle zurückgibt. Die Datenquelle stellt verschiedene Methoden zur Verfügung. Wird eine Datenquelle nicht mehr verwendet, ist es notwendig, die Dispose-Methode aufzurufen, um alle Ressourcen wieder freizugeben.

InitSymbols()	Initialisiert die ReadSymbols-Collection und löst vorhandene Regex-Ausdrücke auf.
ReadSamples(…)	Fragt alle Daten synchron im aufrufenden Kontext ab.
ReadAsync(…)	Fragt alle Daten asynchron ab. Für jedes Datum wird das ReadDelegate aufgerufen.
Dispose()	Schließt alle offenen MQTT-Verbindungen oder Dateizugriffe und gibt alle Ressourcen frei. Die Datenquelle kann danach nicht mehr verwendet werden.

Generische Datentypen:

Die Analytics Data-Exchange-API unterstützt nur „unmanaged structs“ als generische Datentypen. Dies bedeutet folgende Einschränkungen

1. Innerhalb der Strukturen dürfen nur Strukturen oder Basis-Datentypen verwendet werden.
2. Arrays innerhalb der Strukturen sind nur mit „unsafe“ Code möglich. (Fixed-Array-Size)
3. Es können nur Arrays von Basis-Datentypen verwendet werden. Arrays von Strukturen sind nicht möglich.

Für die Strukturen kann außerdem nur ein sequentielles oder explizites Layout verwendet werden (siehe https://docs.microsoft.com/de-de/dotnet/api/system.runtime.interopservices.structlayoutattribute?view=net-6.0)

1. Beim Schreiben von Daten sollte ein explizites Layout verwendet werden.
2. Beim Schreiben von Daten darf das Layout kein Padding enthalten (Pack = 1)
3. Beim Lesen von Daten muss das Padding der Datenquelle entsprechen (https://infosys.beckhoff.com/index.php?content=../content/1031/tc3_plc_intro/3539428491.html&id=)

Strukturierte Datentypen:

Die Analytics Data Exchange API erkennt die Datentypen von Symbolen automatisch, sollte kein generischer Datentyp angegeben sein. Handelt es sich bei dem verwendeten Symbol um einen strukturierten Datentyp, wird ein IStructuredValue zurückgegeben. Vom IStructuredValue können zum einen die Werte der Subsymbole abgefragt werden. Zum anderen kann dieser direkt in einen JSON-Text konvertiert werden. Die letzte Möglichkeit ist die Erzeugung eines IStructedValueReader, um das gesamte Symbol ähnlich eines XMLReaders durchzuparsen.

```
void Read(ISample sample)
{
  foreach(var value in sample.Values)
  {
    if(value is IStructuredValue stValue)
    {
      //Values of structured DataTypes without generic DataType could be proccessed here
      var subValue = value["SubValues1"]; //SubValues1 is a known SubElement of the structured Value

      var jsonString = stValue.ToJsonString(); //Convert the value to JSON

      IStructuredValueReader reader stValue.GetReader(); //Use reader to parse the value iterative
```

```
        }
    }
}
```

Fehlermeldungen

Treten beim Verwenden der Data Exchange API Fehler auf, werden diese per Ausnahme an den Benutzer weitergegeben. Folgende Ausnahmen werden verwendet:

Klassenname	Oberklasse	Beschreibung
DataExchangeException	Exception	Oberklasse für alle weiteren Ausnahmen innerhalb der API.
InvalidSymbolDefinitionException	DataExchangeException	Die konfigurierten Symbole sind nicht korrekt.
SymbolNotFoundException	InvalidSymbolDefinitionException	Ein konfiguriertes Symbol konnte im Datenstrom nicht gefunden werden.
InvalidSymbolNameException	InvalidSymbolDefinitionException	Der eingegebene Symbolname ist nicht gültig.
InvalidSymbolOffsetException	InvalidSymbolDefinitionException	Der eingegebene Symboloffset ist kleiner als 0.
InvalidSymbolByteSizeException	InvalidSymbolDefinitionException	Die eingegebene Größe in Bytes ist kleiner oder gleich 0.
InvalidSymbolTypeCodeException	InvalidSymbolDefinitionException	Der eingegebene TypeCode für ein Symbol ist nicht gültig.
InvalidSymbolRegexPatternException	InvalidSymbolDefinitionException	Das eingegebene Regex-Pattern für eine Symbol ist nicht gültig.
InvalidSymbolFilterException	InvalidSymbolDefinitionException	Der eingegebene SymbolFilter für eine Regex-Pattern ist nicht gültig.
DataExchangeLicenseException	DataExchangeException	Es konnte keine gültige Lizenz für die API gefunden werden oder TwinCAT ist nicht im Run-Mode.
DataExchangeTimeoutException	DataExchangeException	Eine Timeout-Zeit wurde überschritten.
QueueFullException	DataExchangeException	Eine Warteschlange für die Abarbeitung von Daten ist voll, da die Abarbeitung zu langsam war.
FileSourceNotFoundException	DataExchangeExeption	Im angegebenen Ordner konnte kein Analytics-File gefunden werden.
ValueExtractException	DataExchangeException	Beim Auslesen der Werte aus dem Datenstrom ist ein Fehler aufgetreten. Genauere Informationen gibt es in der inneren Ausnahme.
DataExchangeMQTTException	DataExchangeException	Ein Fehler ist bei der MQTT-Verbindung aufgetreten.
BrokerConnectException	DataExchangeMQTTException	Eine Verbindung zum Broker konnte nicht aufgebaut werden.
TopicNotFoundException	DataExchangeMQTTException	Das angegebene Topic wurde am Broker nicht gefunden.
InvalidMqttStreamFormat	DataExchangeMQTTException	Das angegebene Stream-Format ist nicht korrekt.

Beispiele:

Beispiele für die Verwendung der Analytics Data Exchange API finden Sie hier:

https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/12970252171.zip

7.1.2 Python

Installation

Die folgenden zwei Komponenten müssen für die Verwendung der Analytics Data Exchange API installiert sein:

1. Microsoft Visual C++ Redistributable 2019 (x64/ x86) oder höher. Fehlt dieses Paket treten Fehler bei der Verarbeitung von Symbolen auf.
2. TwinCAT 3 ADS

7.1.2.1 Abbildung der Logger Funktion

Diese Dokumentation beschreibt, wie Daten mithilfe der Analytics Data Exchange API im Analytics-Format versendet oder gespeichert werden können.

Lizenz

Für das Schreiben von Daten im TwinCAT-Analytics-Format wird eine Lizenz für den TwinCAT 3 Analytics Logger benötigt. Die Data Exchange API kann mit einer 7-Tage Lizenz getestet werden.

Erzeugen und Konfigurieren einer Datenablage:

Zum Versenden oder Speichern von Daten im Analytics-Format wird im ersten Schritt ein Builder erzeugt, um die Datenablage zu konfigurieren:

```
Factory = SinkFactory()

Builder = Factory.ToFile(FilePath)

Builder = Factory.ToMessageBroker(Broker, 1883)
```

Die verschiedenen Builder stellen unterschiedliche Methoden zur Konfiguration bereit. Symbole können über die Methode „WithSymbol“ hinzugefügt werden. Es können beliebig viele Symbole verwendet werden. Für die meisten Basisdatentypen stehen bereits Symbol Implementierungen bereit. Strukturierte Datentypen und Arrays können zusätzlich implementiert werden.

```
class MyStruct(Struct):

def __init__(self, name : str):
self.b = Bool("b")
self.i64 = Int64("i64")
self.f64 = Real64("f64")
super().__init__(name)

TestValInt = Int32("i32")
TestValFloat = Real32("f32")
TestValArray = Array("arrf64", Real64(), [10])
TestValStruct = MyStruct("myStruct")
Builder.WithSymbol(TestValInt)
Builder.WithSymbol(TestValFloat)
Builder.WithSymbol(TestValArray)
Builder.WithSymbol(TestValStruct)
```

Ist der Builder vollständig konfiguriert wird über die Build Methode eine neues Sink angelegt, welche mit InitWrite initialisiert wird. Anschließend wird über die Write-Methode der aktuelle Wert der Symbole geschrieben. Zwischen mehreren Aufrufen der Write-Methode können die Werte der Symbole aktualisiert werden. Sind alle Daten geschrieben wird der Schreibprozess mit CompleteWrite beendet:

```
SampleCount = 32

TestValInt.Value = 1000
TestValFloat.Value = 0.25
Sink.Write()
for i in range(0, SampleCount):
TestValInt.Value += 1
TestValFloat.Value += 0.25
Sink.Write()
Sink.CompleteWrite()
```

Beispiel:

Das vollständige Beispiel finden Sie hier: https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/17763893899.zip

7.1.2.2 Abbildung der Runtime Funktion

Diese Dokumentation beschreibt, wie Daten mithilfe der Analytics Data Exchange API aus dem Analytics-Format ausgelesen werden können.

Lizenz

Für das Lesen von Daten aus dem TwinCAT-Analytics-Format wird eine Lizenz für die „TwinCAT 3 Analytics Runtime“ oder „TwinCAT 3 Analytics Runtime Base“ benötigt. Die Data Exchange API kann mit einer 7-Tage Lizenz getestet werden.

Erzeugen und Konfigurieren einer Datenquelle:

```
Factory = SourceFactory()

Builder = Factory.FromLiveStream(Broker, 1883)

Builder = Factory.FromFile(FilePath)
```

Die verschiedenen Builder stellen unterschiedliche Methoden zur Konfiguration bereit. Symbole können über die Methoden „WithSymbol“, „WithName“ oder „WithRegex“ hinzugefügt werden:

```
TestValue = UInt64("Variables.nGrowSteady")
Builder.WithSymbol(TestValue)

Builder.WithName("TestValues.nGrowSteady")
Builder.WithRegex("Variables.f", 0, RegexSymbolFilter.IncludeChilds)
```

Ist der Builder vollständig konfiguriert wird über die Build-Methode eine neues Sink angelegt. Die Datenabfrage startet durch Aufrufen der Read-Methode. Als Parameter wird ein Callback-Methode übergeben. Enthält das über geben Sample keine Symbole wurden bereits alle Daten ausgelesen. Gibt die Callback-Methode einen Wert ungleich 0 zurück wird das Auslesen abgebrochen.

```
def Callback(sample : ISample) -> int:
if sample.Symbols() is not None:
print(f'Stamp = {sample.Timestamp()}, {sample.Symbols()[0].SymbolName()} = {sample.Symbols()
[0].Value}')
Values.append(sample.Symbols()[0].Value)
Stamps.append(DateTimeToFileTime(sample.Timestamp()))
if len(Stamps) >= SampleCount:
return False
return True

Source.Read(Callback)
```

Beispiel:

Das vollständige Beispiel finden Sie hier: https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/17763978891.zip

8 Beispiele

8.1 Pick-and-place-Anwendung

Das folgende Beispiel zeigt eine Analyse einer XTS-Anwendung zusammen mit einem Pick-and-place-Roboter. Es bietet Ihnen zwei Möglichkeiten:

1. Sie können die Beispieldaten verwenden und Ihr eigenes TwinCAT Analytics-Projekt aufbauen.
2. Sie können das fertige „ready-to-use“ TwinCAT Analytics-Projekt verwenden, um nachzuvollziehen, wie TwinCAT Analytics funktioniert.

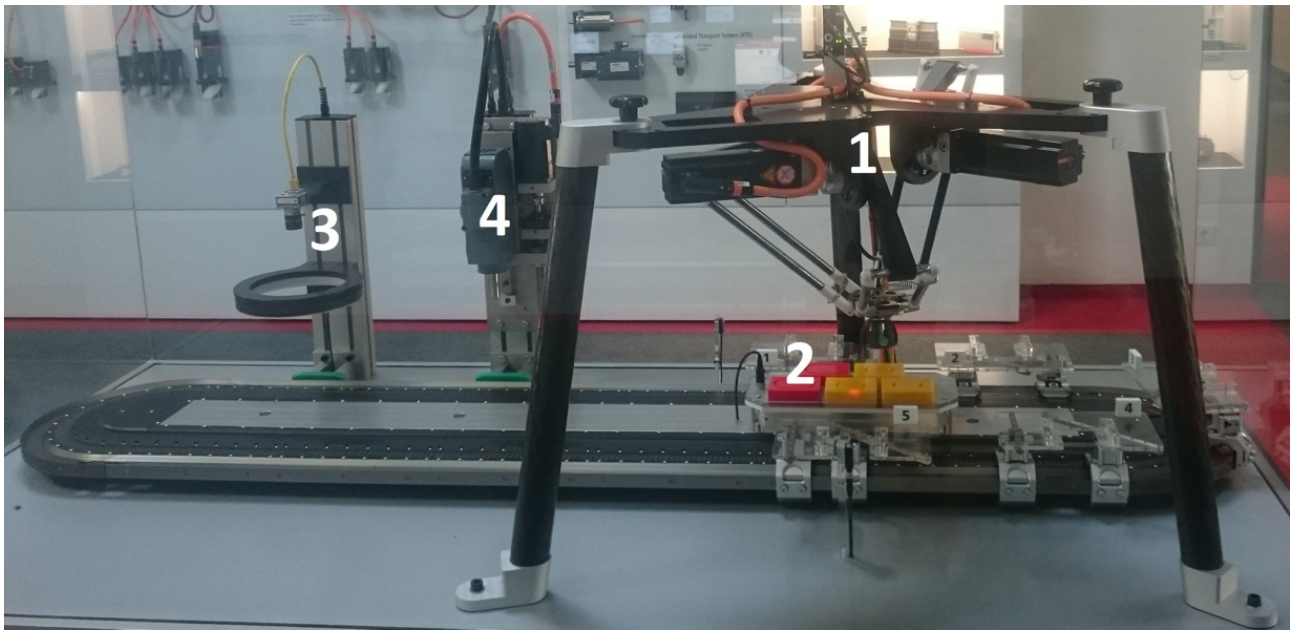
In beiden Fällen ist es sinnvoll, die Maschinenanwendung zu verstehen, die die Daten für die Analyse liefert. Der Ablauf der Anwendung wird im Video gezeigt:

video::https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/6815846411.mp4.

Anwendungsbeschreibung

Dieses Applikationsbeispiel beschreibt eine Simulation zur Bearbeitung von Produktionswaren. Die Anwendung transportiert Werkstücke über ein Beckhoff XTS-System von einem Lager zu einem Bohrer. Je nach Farbe des Werkstücks wird ein anderes Muster in jedes Teil gefräst. Das gezeigte XTS-System hat eine Länge von 4000 mm. Es umfasst zehn Mover, wobei zwei Mover zusammen eine Ablage tragen. Ein Pick-and-place-Roboter (1) ist am Werkstücklager (2) positioniert. Zwei rote und drei gelbe Teile liegen in fünf Lagerpositionen, die der Picker über eine Vakuumpumpe aufnehmen oder ablegen kann. Auf der gegenüberliegenden Seite des XTS befinden sich ein Vision System (3) zur Identifizierung der Werkstückfarbe und ein Fräser (4) für die spezifische Fräskontur.

Die Ablagen laufen zum Werkstücklager. Der Picker nimmt die Teile auf und be- oder entlädt die Ablagen. Anschließend läuft die Ablage zum Scanner. Dort wird die Farbe des Werkstücks gescannt und dem nachfolgenden Fräser mitgeteilt. Je nach Farbe werden die Werkstücke anders bearbeitet, in gelbe Teile wird ein Kreis, in rote Teile eine ovale Form gefräst. Danach läuft das Werkstück zurück zum Lager und wird dort vom Picker eingelagert. Die Ablage wird mit einem anderen Teil beladen und läuft weiter.



Variablenbeschreibung

Die folgenden Variablen werden vom TwinCAT Analytics Logger im angehängten Analytics File Format aufgezeichnet. Sie können die Daten für Ihr eigenes Analytics-Konfiguratorprojekt oder für die vorgegebene Konfiguration in der bereitgestellten tnzip-Datei verwenden. Sinn und Funktion der einzelnen Variablen werden in der folgenden Tabelle dargestellt.

Variablenname	Data Type	Beschreibung
bMillerMilling	BOOL	Informationen zur Aktivität des Bohrers: 1, wenn der Bohrer aktiv ist, und 0, wenn der Bohrer inaktiv ist.
bMillerMovingZ	BOOL	Informationen zur Bewegung des Bohrers in z-Richtung: Ist 1, wenn sich der Bohrer in z-Richtung bewegt, und 0, wenn er sich nicht in z-Richtung bewegt.
bMillerSpindleRotation	BOOL	Informationen zur Aktivität der Bohrnadel: Ist 1, wenn sich die Nadel bewegt und der Bohrer aktiv ist, und 0, wenn sie sich nicht bewegt.
bNewWorkpieceColorScanned	BOOL	Registriert, ob eine neue Farbe gescannt wurde (= 1), und informiert dann den Bohrer. Wenn sich die Farbe ändert, muss das Bohrmuster geändert werden. Wenn die Farbe noch gleich ist (=0), ändert sich das Bohrmuster nicht.
bPickerHoldingWorkpiece	BOOL	Ist 1, wenn der Picker ein Werkstück hält, und 0, wenn nicht.
bPickerInStorage1Position	BOOL	Ist 1, wenn sich der Picker in Lagerposition 1 befindet.
bPickerInStorage2Position	BOOL	Ist 1, wenn sich der Picker in Lagerposition 2 befindet.
bPickerInStorage3Position	BOOL	Ist 1, wenn sich der Picker in Lagerposition 3 befindet.
bPickerInStorage4Position	BOOL	Ist 1, wenn sich der Picker in Lagerposition 4 befindet.
bPickerInStorage5Position	BOOL	Ist 1, wenn sich der Picker in Lagerposition 5 befindet.
bPickerInXtsPosition	BOOL	Ist 1, wenn sich der Picker in XTS-Position befindet, d. h. über dem XTS.
bPickerMoving	BOOL	Information, wenn sich der Picker bewegt (=1) oder inaktiv ist (=0).
fMillerPositionZ	LREAL	Gibt die z-Position des Bohrers an. Bei einem Wert von 224 befindet sich der Bohrer in der Bohrposition, bei 245 ist er inaktiv.
fMillerSpindleRotation	LREAL	Gibt die Anzahl der Umdrehungen pro Minute der Bohrnadel an.
nColorOfLastScannedWorkpiece	INT	Registriert die Farbe des zuletzt gescannten Werkstücks. 5 → gelbes Werkstück 4 → rotes Werkstück
nCurrentProductionTime	ULINT (64 Bit)	Registriert die aktuelle Produktionszeit.
nVacuumPressure	DINT	Der Picker greift die Werkstücke über Druckluft. Diese Variable gibt den aktuellen Druck in mbar an.
stMachineVibrations aMachineVibrations[1...10]	STRUCT with ARRAY of LREAL	Die Struktur stMachine Vibrations gibt die Vibrationen der Maschinenbasis in der Einheit m/s^2 mit Oversampling an.
stPickerPosition aPickerPosition[1...3]	STRUCT with ARRAY of LREAL	Die Struktur stPickerPosition enthält ein Array mit drei Werten. Die Werte geben die x-/y- und z-Position des Pickers an.

Variablenname	Data Type	Beschreibung
stTableInMillingPosition aTableInMillingPosition[1...5]	STRUCT with ARRAY of BOOL	Die Struktur stTableInMillingPosition enthält ein Array mit fünf Werten, von denen jeder für eine Ablage steht: Wert 1 für Ablage 1, Wert 2 für Ablage 2... wenn der Wert 1 ist, befindet sich die Ablage in der Bohrposition.
stTableInStoragePosition aTableInStoragePosition[1...5]	STRUCT with ARRAY of BOOL	Die Struktur stTableInStoragePosition enthält ein Array mit fünf Werten, von denen jeder für eine Ablage steht: Wert 1 für Ablage 1, Wert 2 für Ablage 2... wenn der Wert 1 ist, befindet sich das Ablage im Werkstücklager.
stWorkpieceColors aWorkpieceColors[1...5]	STRUCT with ARRAY of INT	Die Struktur stWorkpieceColors enthält ein Array mit fünf Werten, von denen jeder für ein Werkstück steht und seine Farbe angibt (als Int-Wert): Werkstück 1: gelb Werkstück 2: gelb Werkstück 3: rot Werkstück 4: rot Werkstück 5: gelb
stWorkpiecePositions aWorkpiecePosition[1...5]	STRUCT with ARRAY of INT	Die Struktur stWorkpiecePositions enthält ein Array mit fünf Werten für jedes Werkstück. Der Int-Wert gibt die aktuelle Position des Werkstücks an: 1: Lager 1 2: Lager 2 3: Lager 3 4: Lager 4 5: Lager 5 6: Ablage 1 7: Ablage 2 8: Ablage 3 9: Ablage 4 10: Ablage 5 11: Picker
stXtsMoverAcceleration aXtsMoverAcceleration[1...10]	STRUCT with ARRAY of LREAL	Die Struktur stXtsMoverAcceleration enthält ein Array mit zehn Werten für die Mover 1-10 und gibt die Beschleunigung des jeweiligen Movers an.
stXtsMoverMovingNegative aXtsMoverMovingNegative[1...10]	STRUCT with ARRAY of BOOL	Die Struktur stXtsMoverMovingNegative enthält ein Array mit zehn Werten für die Mover 1-10 und ist 1, wenn sich der jeweilige Mover rückwärts bewegt.
stXtsMoverMovingPositive aXtsMoverMovingPositive[1...10]	STRUCT with ARRAY of BOOL	Die Struktur stXtsMoverMovingPositive enthält ein Array mit zehn Werten für die Mover 1-10 und ist 1, wenn sich der jeweilige Mover vorwärts bewegt.
stXtsMoverPositionsX aXtsMoverPositionsX[1...10]	STRUCT with ARRAY of LREAL	Die Struktur stXtsMoverPositionsX enthält ein Array mit zehn Werten für die Mover 1-10. Der Wert gibt die x-Position des jeweiligen Movers auf dem XTS an.

Variablenname	Data Type	Beschreibung
stXtsMoverVelocity aXtsMoverVelocity[1...10]	STRUCT with ARRAY of LREAL	Die Struktur stXtsMoverVelocity enthält ein Array mit zehn Werten für die Mover 1-10 und gibt die Geschwindigkeit des jeweiligen Movers an.

Wie Sie die Daten erhalten

Im Download-Bereich finden Sie eine 30-minütige Aufzeichnung des XTS-Systems, die für die weitere Analyse verwendet werden kann. Um auf diese Daten zuzugreifen, müssen Sie die Daten mit dem Analytics Storage Provider in einem historischen Stream bereitstellen. Die Verfügbarkeit eines Message-Brokers sollte dafür gegeben sein. Für die vorbereiteten Samples müssen alle Komponenten auf dem lokalen System laufen.

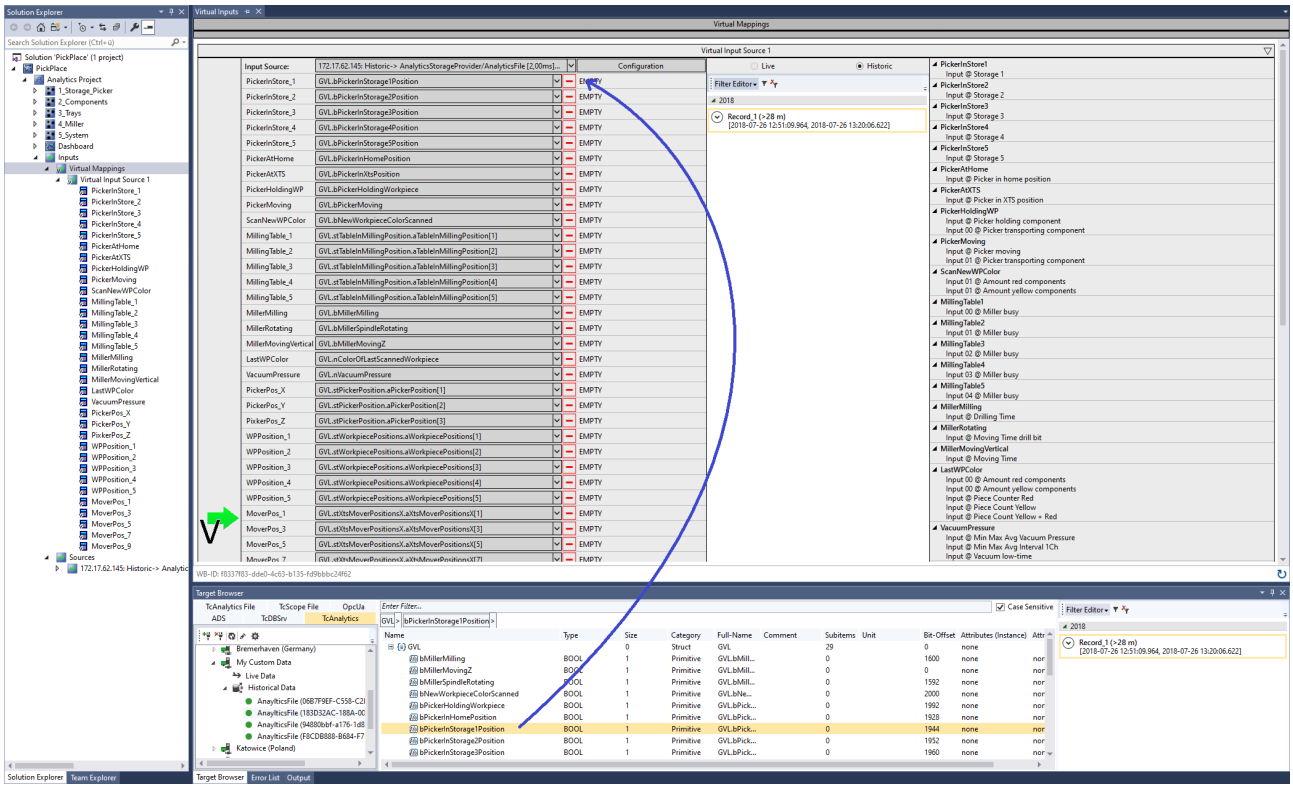
Installieren Sie den Analytics Storage Provider. Folgen Sie den gegebenen Anweisungen.

1. Importieren Sie ihre Analytics File Daten in den Storage Provider (siehe: Importieren von Analytics Files).
2. Nutzen Sie den Analytics Storage Provider Configurator um den Storage Provider Windows Service zu starten.
3. Öffnen Sie den TwinCAT Target Browser in der TcXaeShell oder im Visual Studio®. Wählen Sie die TcAnalytics Registrierkarte aus und fügen Sie Ihren Message-Broker hinzu. Im Baum auf der linken Seite finden Sie nun einen Eintrag für die Historischen Daten, die Ihnen bei Auswahl in der Symbolliste auf der rechten Seite angezeigt werden.
4. Verbinden Sie einzelne Symbole mit den Eingängen ihrer Analytics-Algorithmen.
5. Starten Sie die Analyse.

Analyse mit der Analytics Workbench

Im Folgenden wird die Analyse erläutert, die Gegenstand des TwinCAT Analytics-Projektes aus dem Download-Bereich ist. Laden Sie das fertige Projekt herunter und öffnen Sie es.

Selektieren Sie den Virtual Mappings Knoten und browsen Sie zusätzlich im Target Browser auf das von Ihrem Storage Provider vorgegebene Topic. Wählen Sie die Daten aus und verknüpfen diese mit den entsprechenden Virtuellen Inputs im Editor.



Beschreibung vom Analytics-Projekt

Netzwer-kname	Verwendete Variablen	Verwendete Algorith-men	Beschreibung
1_Storage_Picker	bPickerinStoragePosition, bPickerHoldingWorkpiece, bPickerInXtsPosition, bPickerInHomePosition, bPickerMoving, stPickerPosition	Edge Counter, Math Operation, Edge Counter OnOff, Logic Operation Counter, Numerical Compare, Min Max Avg	Picker: Dieses Netzwerk analysiert die Aktionen des Pickers. Es zählt die Mengen, wenn der Picker ein Werkstück hält, sich in der Ausgangsposition oder in der XTS-Position befindet. Zudem wird analysiert, wie oft der Picker ein Werkstück transportiert, wie oft er einen Mover belädt sowie seine xyz-Position. Wie oft lädt der Picker ein Werkstück? Speicherung: Dieses Netzwerk analysiert die Lagernutzung der fünf Lagerpositionen neben dem Picker. Die Mengen und Anteile der Lagernutzungen werden gezählt. Wird jedes Lager gleichermaßen genutzt?
2_Components	bNewWorkpieceColorScanned, bMillerMilling, nColorOfLastScannedWorkpiece, stWorkpiecePosition	Logic Operation Counter, Threshold String Classificator, Histogram, Trend Line	Dieses Netzwerk analysiert die Werkstücke selbst. Wie viele Werkstücke von jeder Farbe an Scanner und am Fräser verarbeitet werden sowie die aktuelle Position jedes Werkstücks kann erkannt werden. Die Histogramm-Algorithmen zeigen die Verteilung der Werkstückpositionen. Wie viele rote Werkstücke wurden verarbeitet?

Netzwer-kname	Verwendete Variablen	Verwendete Algorith-men	Beschreibung
3_Mover	stTableInMillingPosition, sXtsMoverPosition	XTS Velocity Analysis, XTS Distance Integrator, Min Max Avg Interval, Math Operation, Threshold Classificator, Trend Line, Edge Counter, Lifecycle Analysis, Logic Operation Counter	Distance: Dieses Netzwerk analysiert die Strecke, die jede Ablage zurückgelegt hat, und berechnet die Gesamtstrecke aller Mover. Da die Länge des XTS-Systems bekannt ist (4000 mm), kann die durchschnittliche Anzahl der Runden für jede Ablage abgeleitet werden. Die Threshold-Analyse gibt eine Warnung nach 40 Runden und einen Alarm nach 45 Runden aus. Welche Strecke haben alle Ablagen insgesamt zurückgelegt? Ablageposition: Dieses Netzwerk analysiert, wie oft die Ablagen am Lager oder am Fräser anhalten, und stoppt, wie lange eine Ablage für eine Runde benötigt. Wie oft haben alle Ablagen am Fräser angehalten? Velocity: Dieses Netzwerk analysiert die aktuelle Geschwindigkeit der Ablagen sowie die minimale und maximale Geschwindigkeit in den letzten 20 Sekunden. Die Ablagen können sich rückwärts bewegen. Daher kann die minimale Geschwindigkeit negativ sein. Welche maximale Geschwindigkeit können Sie ablesen?
4_Miller	bMillerMilling, bMillerSpindleRotating, bPickerMoving	Event Timing Analysis, Lifetime Analysis	Dieses Netzwerk führt die Analyse für den Fräser durch. Bohrzeit: Zeit, in der der Fräser aktiv ist. Bewegungszeit der Bohrspitze: Zeit, in der die Bohrspitze aktiv ist. Bewegungszeit des Bohrers: Zeit, in der der Bohrer aktiv ist. Welche der drei Aktionen hat der Fräser am längsten ausgeführt?

Netzwer-kname	Verwendete Variablen	Verwendete Algorith-men	Beschreibung
5_System	bNewWorkpieceColorScanned, bMillerMilling, nVacuumPressure, stMachineVibrations	Min Max Avg, Min Max Avg Interval, Timing Analysis, Math Operation Lifecycle Analysis, Threshold Classificator, Continuous Piece Counter	Productivity: Die Produktivität wird analysiert, indem die verarbeiteten Werkstücke pro 20 Sekunden gezählt und mit den vorangegangenen Intervallen verglichen werden. Ausgehend von der maximalen Anzahl wird die Produktivität der folgenden Intervalle abgeleitet. Wie lautet die maximale Anzahl der Werkstücke in einem Intervall? Technische Bauteile: Dieses Netzwerk analysiert den Vakuumdruck des Pickers und gibt eine Warnung oder Alarm Meldung aus, wenn der Lebenszyklus von der Vakuum-Ansauge unter definierte Schwellwerte fällt. Wie lange dauert es bis zu einer Warnung?

SPS-Codegenerierung und HMI

Nach Abschluss der Analyse in der Analytics Workbench können Sie SPS-Code und ein HMI-Dashboard generieren (siehe auch: 24-stündige Anwendung von Analytics [▶ 34]):

1. Mit einem Doppelklick auf den Projekt Knoten öffnet sich die Übersichtsseite. Dort wählen Sie **deploy Runtime** aus

2. Es öffnet sich der **Deploy Wizard**, der Sie Schritt für Schritt anleitet. Soweit keine individuellen Einstellungen gewünscht sind, können Sie einfach fortfahren.

3. Optional: Wählen Sie in der Registerkarte PLC einen CPU-Kern aus, auf dem die Analyse ausgeführt werden soll. Für die Ausführung in einer virtuellen Maschine muss ein isolierter Kern ausgewählt werden.

4. In der Registerkarte HMI Dashboard aktivieren Sie die Generierung. Passen Sie das Design nach Ihrem Geschmack an.

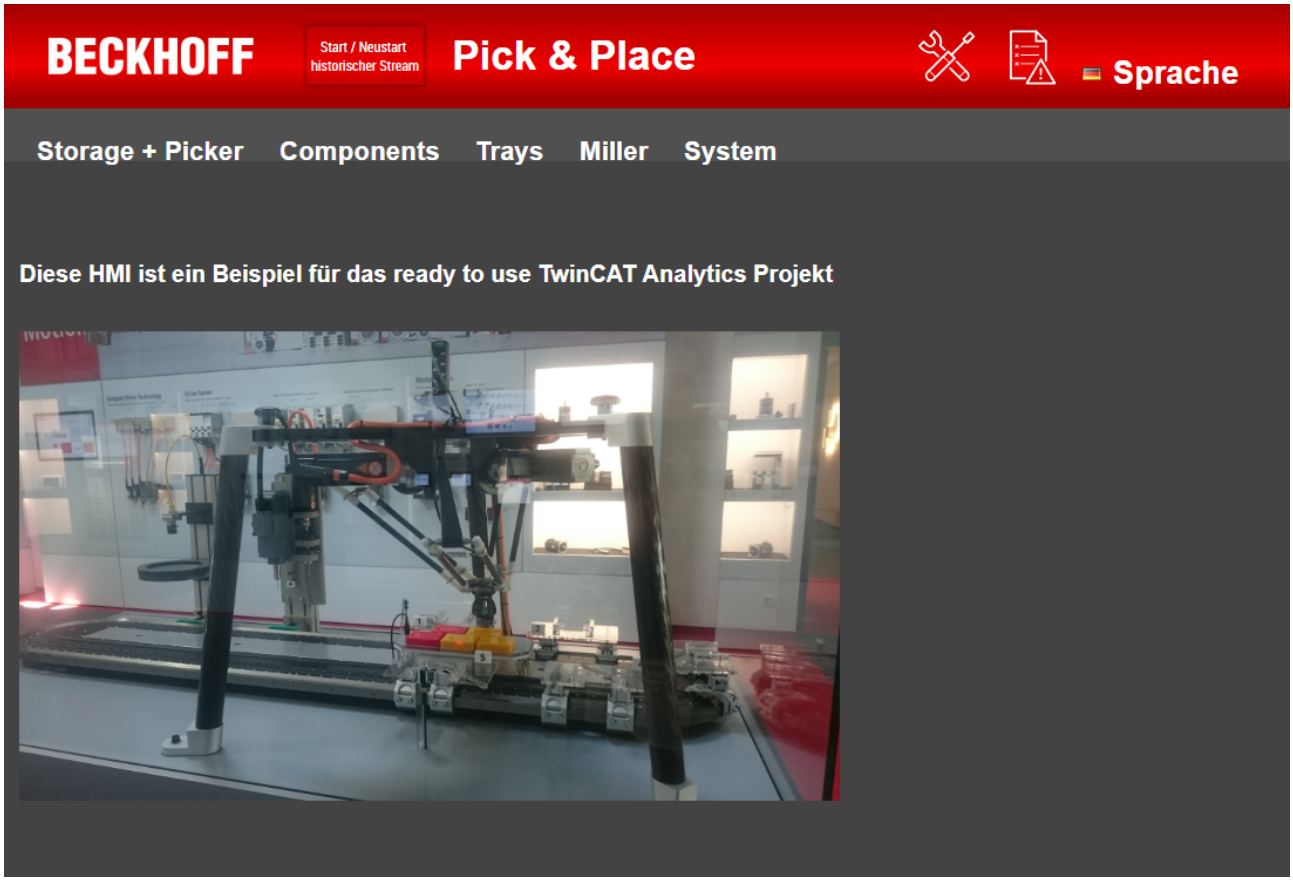
5. Nach abschließender Bestätigung über den **Deploy** Taster öffnet sich eine neue Entwicklungsumgebung. Es werden ein TwinCAT XAE Projekt und ein TwinCAT HMI Projekt automatisch generiert.

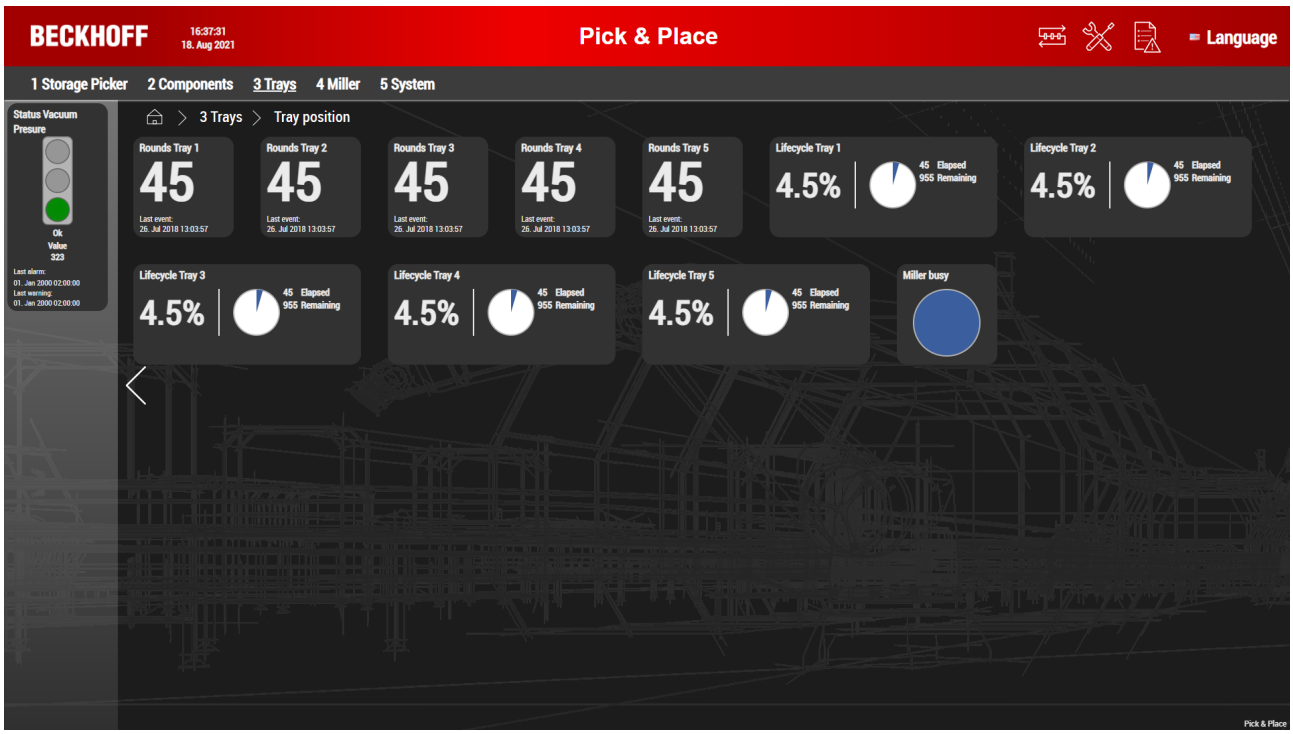
6. Am Ende der Generierung öffnet sich ihr Dashboard im Browser. Sie können ihre ursprünglichen Netzwerke über die Navigation betrachten.

7. Damit Sie ihre historischen Daten analysieren können, müssen Sie die Analyse starten. Dafür öffnen Sie das entsprechende Optionselement über das dritte Icon von oben rechts.

8. Wählen Sie den zu analysierenden Zeitbereich Ihrer aufgezeichneten Daten aus und starten Sie die Aufnahme, um die Analyseergebnisse zu verfolgen.

BECKHOFF
16:57:11
18. Aug 2021
Pick & Place
Language
1 Storage Picker
2 Components
3 Trays
4 Miller
5 System
Status Vacuum Presure
Alarm
Value
-458
2 Components
Positions component 1
Positions component 2
Positions component 3
Positions component 4
Amount red components
157
Amount yellow components
237
Position component 1
Tray
Position component 2
Storage
Position component 3
Tray
Position component 4
Storage
Position component 5
Tray
Positions component 5
Trend Line
1 Hour
12 Hours
1 Day
3 Days
7 Days
Red components
Yellow components
Pick & Place


BECKHOFF
Start / Neustart historischer Stream
Pick & Place
Sprache
Storage + Picker
Components
Trays
Miller
System
Diese HMI ist ein Beispiel für das ready to use TwinCAT Analytics Projekt



Download der fertigen SPS-Code-Generierung

Falls Sie sich nur ein fertig generiertes Analytics SPS-Projekt mit der zugehörigen HMI anschauen möchten, ist auch dafür ein Download (Sample_Analysis_PickNPlace_PLC) verfügbar. Um dieses Projekt auch lauffähig zu haben, müssen das Main Topic und die Identifikations-GUID des Storage Providers im Code angepasst werden. Entsprechende Warnungen werden ausgegeben, wodurch Sie innerhalb von TwinCAT bequem zur entsprechenden Stelle navigieren können. Dort finden Sie auch Hinweise, wo die Informationen verfügbar sind.

Downloads

- https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/6925371147.zip
- https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/8817652491.zip
- https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/8817654667.zip

8.2 Pick-and-place-Anwendung: DenStream

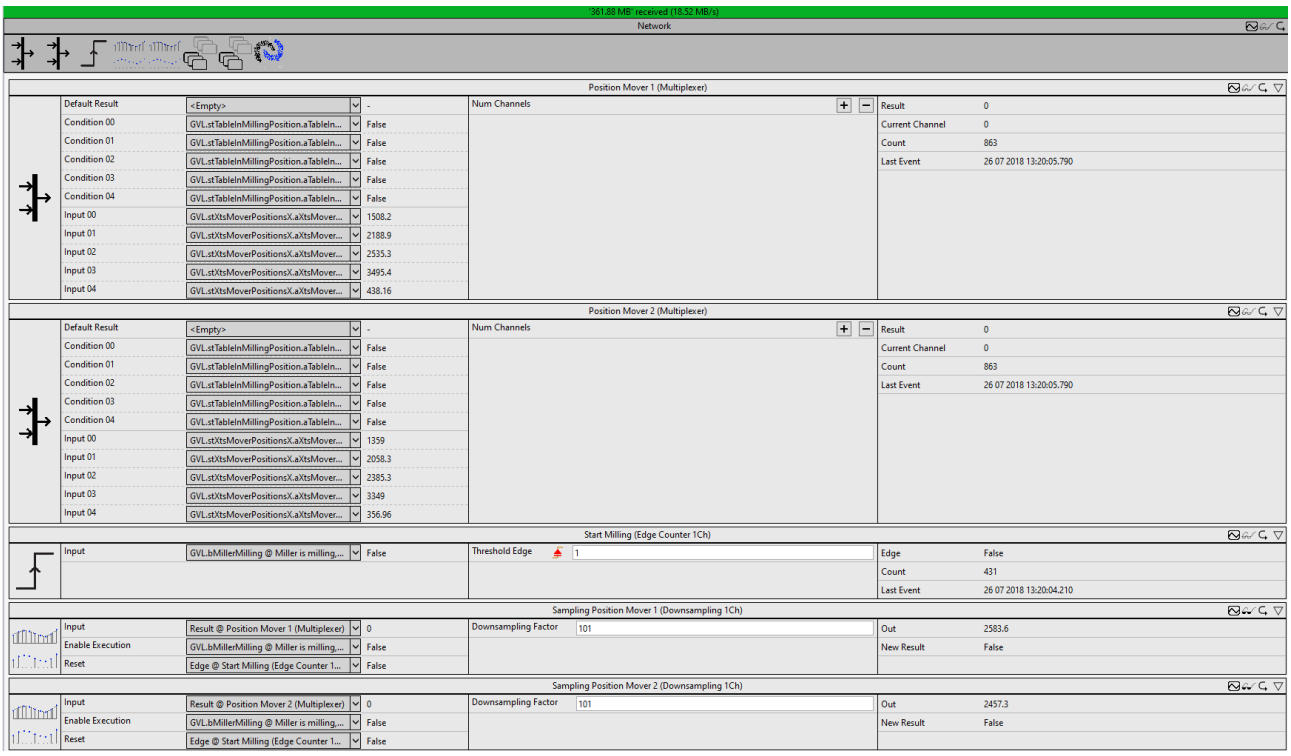
Die nachfolgend beschriebene Applikation ist eine beispielhafte Anwendung des DenStream-Bausteins.

Das Beispiel basiert auf den Daten der Pick-and-place-Anwendung. Aus diesem Grund ist ein gutes Verständnis des Pick-and-place-Beispiels eine gute Basis für das Verständnis dieser Applikation. Der Datenzugriff erfolgt auf dieselbe Weise.

Ziel der Anwendung ist es, anhand der bei dem Fräsvorgang ausgeführten Bewegung, die gefrästen Formen in die beiden Gruppen Oval (rot) und Rund (gelb) zu unterteilen.

Die untenstehenden Abbildungen zeigen die einzelnen Bausteine der Applikation.

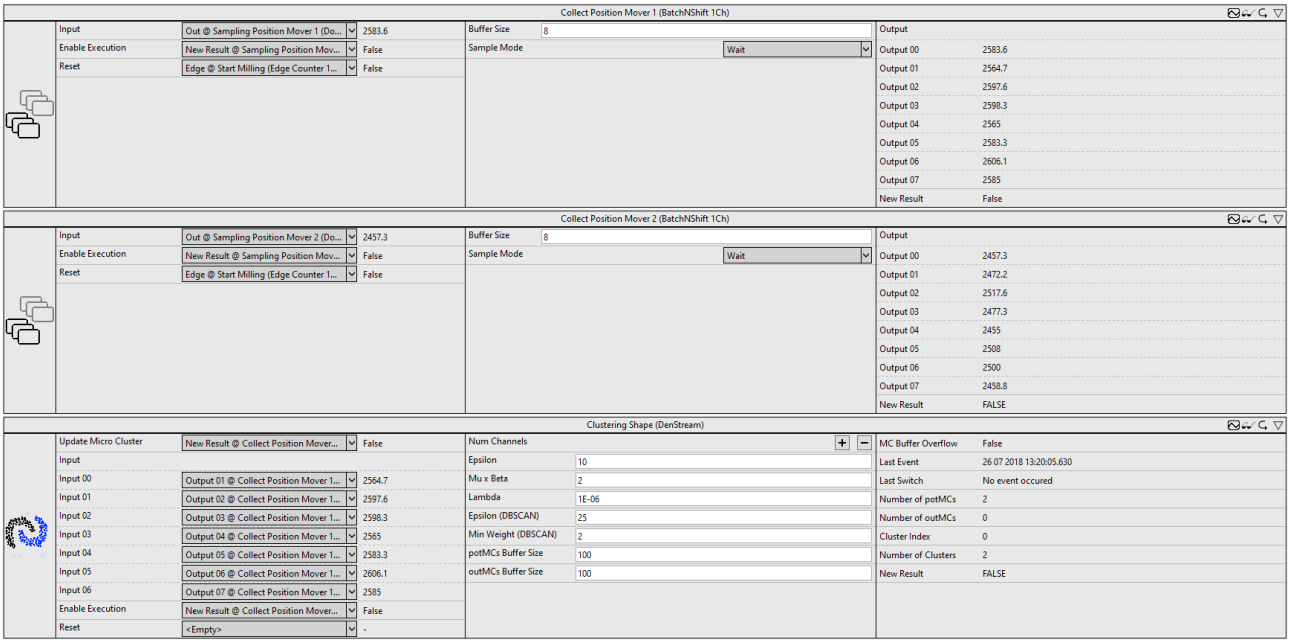
Die Ablagen für die Werkstücke sind mit jeweils zwei Movern auf der XTS-Strecke befestigt. Daher gibt es pro Ablage zwei Positionswerte, mit denen sich die gefräste Form in einem Scope-Projekt abbilden lässt. Für die Bestimmung der Cluster reicht in diesem Fall eine der Positionen aus, die zu sieben verschiedenen Zeitpunkten während des Fräsvorgangs abgetastet wird. Diese sieben Werte werden anschließend gleichzeitig als einzelne Eingänge im DenStream-Baustein verarbeitet und bilden den sieben-dimensionalen Merkmalsraum.



Mithilfe des Bausteine **Position Mover 1 (Multiplexer)** und **Position Mover 2 (Multiplexer)** werden die Positionen der beiden Mover der Ablage extrahiert, dessen Werkstück aktuell gefräst wird.

Der Baustein **Edge Counter (Start Milling)** gibt eine steigende Flanke aus, wenn der Fräsvorgang beginnt. Diese wird als Eingang nachfolgender Bausteine benötigt.

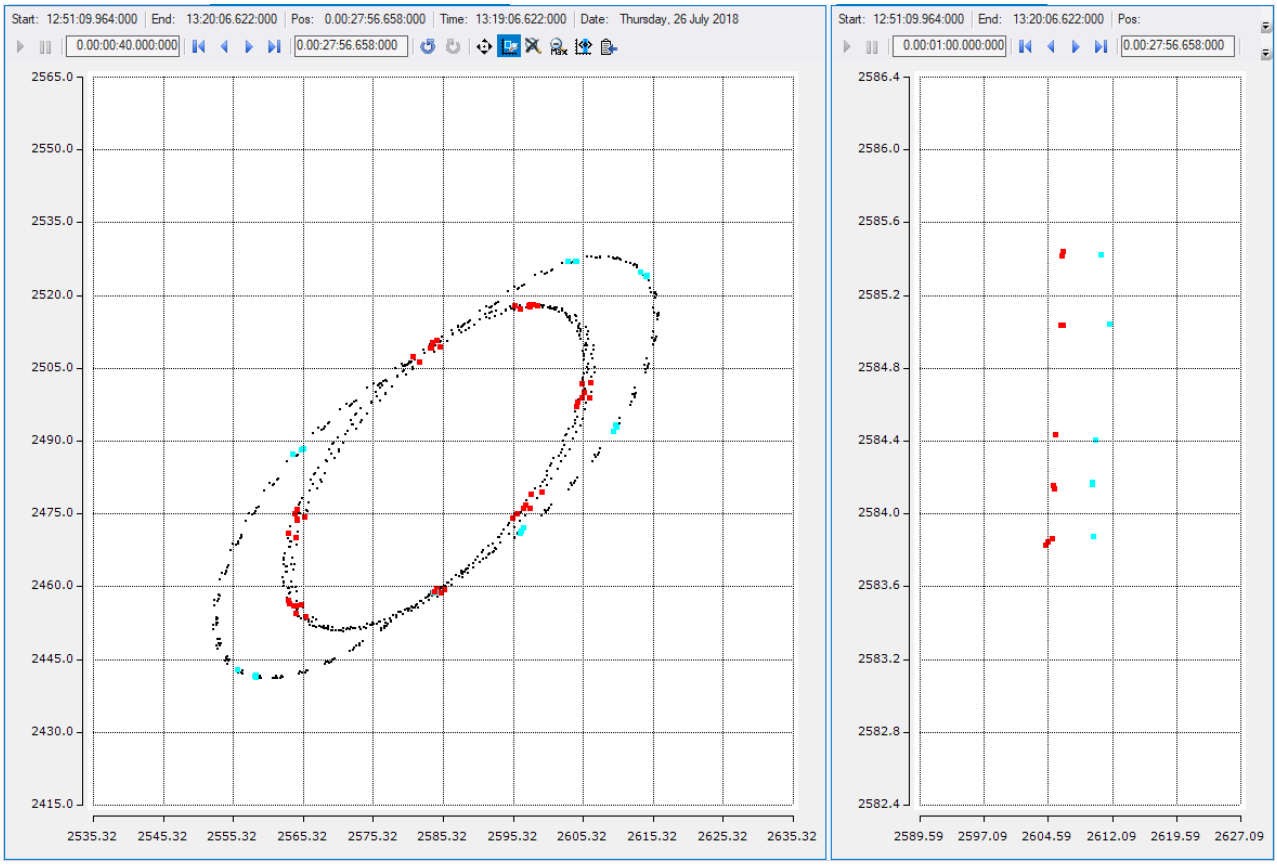
Die beiden Bausteine BatchNShift (**Collect Position Mover 1** & **Collect Position Mover 2**) speichern über den Fräsvorgang, begonnen mit der steigenden Flanke von *bMillerMilling*, über gut 700 Zyklen die Positionswerte ab. Die Bausteine sind nur dann aktiv, wenn gerade gefräst wird (*bMillerMilling* == TRUE).



Aus dem gepufferten Ausgang des ersten BatchNShift-Bausteins (**Collect Position Mover 1**) werden nun, im Abstand von etwa 100 Samples, die Eingänge des DenStream-Bausteins (**Clustering Shape**) erstellt. Sie bilden so den Merkmalsraum. Der erste Wert wird verworfen, da die Position zu Beginn des Fräsvorgangs unabhängig von der Fräsform ist und somit für die Klassifikation keinerlei Informationen beinhaltet.

Die Micro-Cluster werden nur dann aktualisiert, wenn der Baustein DownsamplingBuffer seine Ausgänge aktualisiert (*bNewResult* == TRUE). Um die Parameter-Werte für *Epsilon* und *Epsilon (DBSCAN)* richtig zu abzuschätzen, können Sie sich in einem Plot die Distanzen zwischen den einzelnen Input-Variablen für die verschiedenen Fräsformen anschauen. Da sich die Fräsformen über die Zeit nicht verändern, ist der Parameter-Wert *Lambda* sehr gering gewählt und könnte auch auf 0 gesetzt werden.

In den Scope-Charts sind die Ergebnisse aus der Analyse dargestellt.



Wird die Position der beiden Mover einer Ablage in einem XY-Chart darstellt, so zeigen diese die Bewegungskurve, die von den Ablagen während des Fräsens abgefahren wird. Die Bewegungskurve ist im linken Plot schwarz dargestellt. Im dargestellten Zeitabschnitt wurden beide Fräsformen mehrfach abgefahren. Farbig sind die Punkte dargestellt, die als Eingänge für den DenStream-Baustein genutzt wurden. Mithilfe der dynamischen Styles sind die Punkte, in Abhängigkeit des zugeordneten Cluster-Index, farblich unterschiedlich darstellbar. Es ist zu erkennen, dass die Cluster korrekt zugeordnet wurden. Die Punkte, die stellvertretend für die ovale Fräsform stehen, sind türkis, die der anderen sind rot, ihnen wurden also unterschiedliche Cluster-Indizes zugeordnet.

Der Merkmalsraum der Cluster wird nicht angezeigt. Da dieser siebendimensional ist, kann er nicht direkt visualisiert werden. Im rechten Plot sind stellvertretend zwei Merkmale (5. und 6. Abtastwert) abgebildet. Es ist zu erkennen, dass die Punkte der beiden Cluster räumlich voneinander getrennt sind

Downloads

- https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/12177933195.zip

8.3 Pick-and-place-Anwendung: Korrelation

Die nachfolgend beschriebene Applikation ist eine beispielhafte Anwendung des CorrelationFunctionReference-Bausteins.

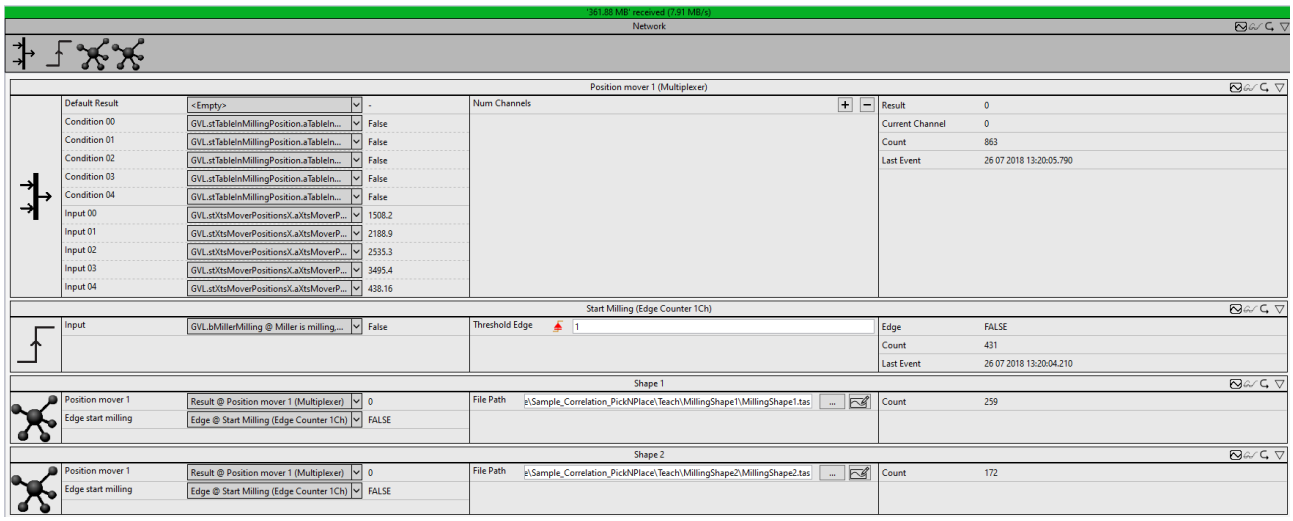
Das Beispiel basiert auf den Daten der Pick-and-place-Anwendung. Aus diesem Grund ist die ein gutes Verständnis des Pick-and-place-Beispiels eine gute Basis für das Verständnis dieser Applikation. Der Datenzugriff erfolgt auf dieselbe Weise.

Ziel der Anwendung ist es, anhand der bei dem Fräsvorgang ausgeführten Bewegung, die gefrästen Formen in zwei Gruppen Oval (rot) und Rund (gelb) zu unterteilen.

Die Bewegung eines der beiden Mover, auf denen die Ablage befestigt ist, wurde zuvor für die beiden Formen aufgezeichnet und liegt in den Dateien *MillingShape1.tas* und *MillingShape2.tas* für die beiden verschiedenen Fräsformen vor. Diese Sequenzen werden jeweils mit dem aktuellen Bewegungsmuster korreliert. Anhand des maximalen Koeffizienten, den der CorrelationFunctionReference-Baustein ausgibt, wird bestimmt, ob die Formen übereinstimmen oder nicht.

Anhand des Ausgangswertes *Maximizing Lag* des CorrelationFunctionReference-Bausteins wird bestimmt mit wie vielen Zyklen Verzögerung die Form abgefahren wird, nachdem der Fräser eingeschaltet wurde.

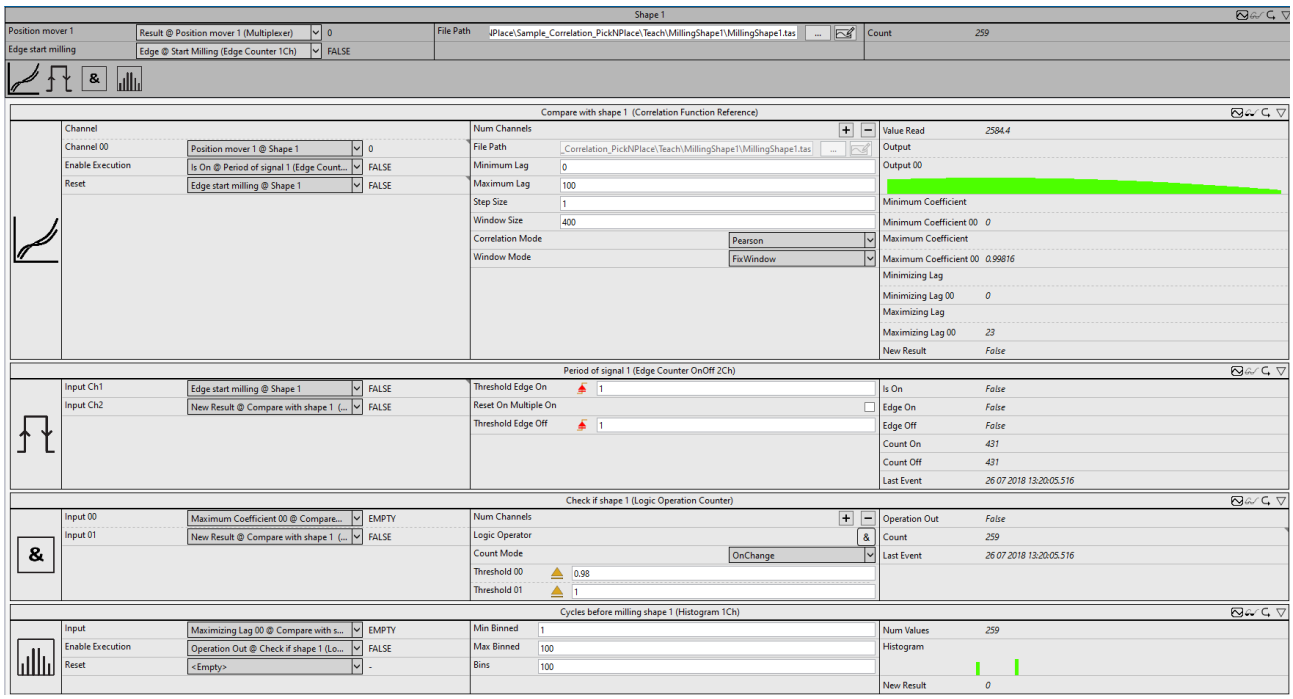
Die untenstehenden Abbildungen zeigen die einzelnen Bausteine der Applikation.



Mithilfe des Bausteins **Position Mover 1 (Multiplexer)** wird die Position des ersten Movers extrahiert, der zu der Ablage gehört, dessen Werkstück aktuell gefräst wird. Dies ist der Mover, für den das Bewegungsprofil über den Verlauf des Fräsvorgangs für die beiden Formen vorliegt. Für die Analyse ist es nicht erforderlich, beide Mover einer Ablage zu betrachten, da die Mover in einem konstant bleibenden Abstand zueinander montiert sind und damit dasselbe Bewegungsprofil aufweisen.

Der Baustein **Edge Counter (Start Milling)** gibt eine steigende Flanke aus, wenn der Fräsvorgang beginnt. Diese Flanke wird als Eingang für nachfolgende Bausteine benötigt.

Die Berechnungen der Korrelationen werden für jede Fräsform einzeln in einem separaten Netzwerk durchgeführt.

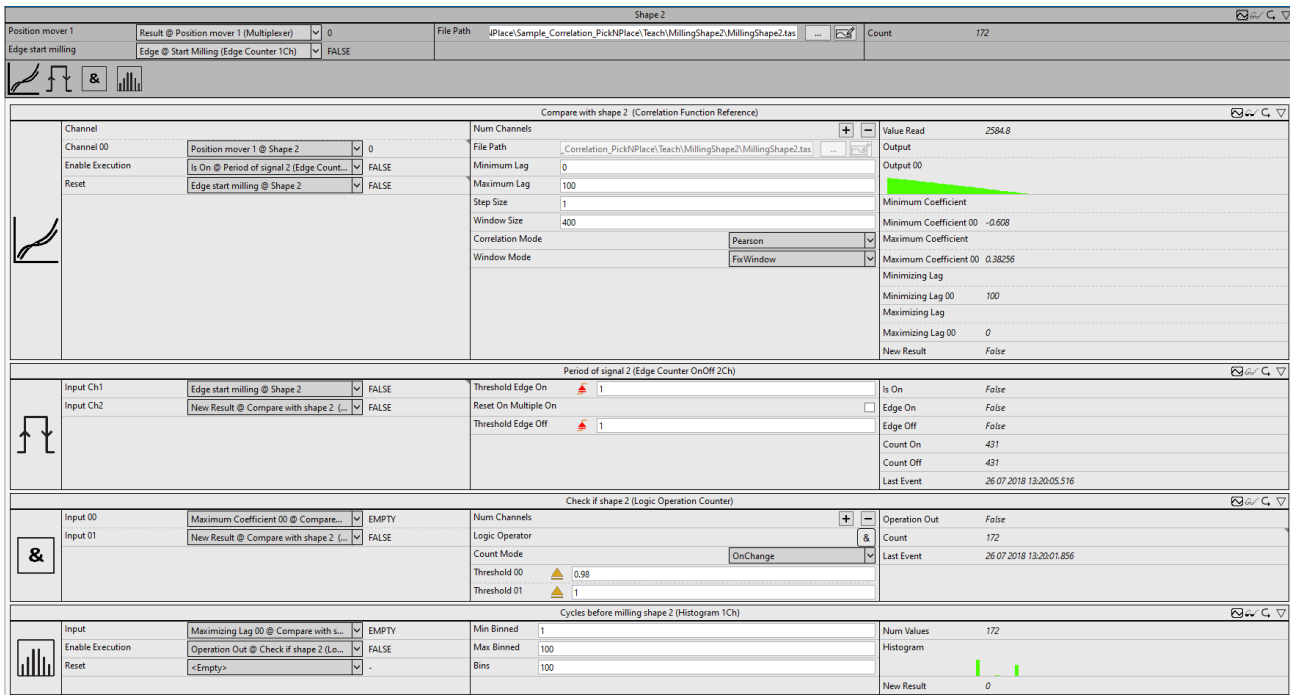


In dem Baustein **Compare with Shape 1 (Correlation Function Reference)** wird die Position des Movers (*Result*-Ausgang des Multiplexer-Bausteins) mit dem aufgenommenen Signal der ersten Form korreliert. Die Korrelationsberechnung startet jeweils mit dem Einschalten der Fräse. Dies wird durch das Verknüpfen des *Reset*-Eingangs mit der steigenden Flanke zum Einschalten der Fräse (*Edge-Ausgang* des **Start Milling (Edge Counter 1Ch)**) erreicht. Denn mit jeder steigenden Flanke dieses Bausteins beginnt die Korrelationsberechnung erneut.

Vor dem Start der Applikation muss der Pfad zu der Referenzdatei gesetzt werden. Die Referenzdatei befindet sich im Projektverzeichnis in dem Ordner *„Teach\MillingShape1\MillingShape1.tas“.*

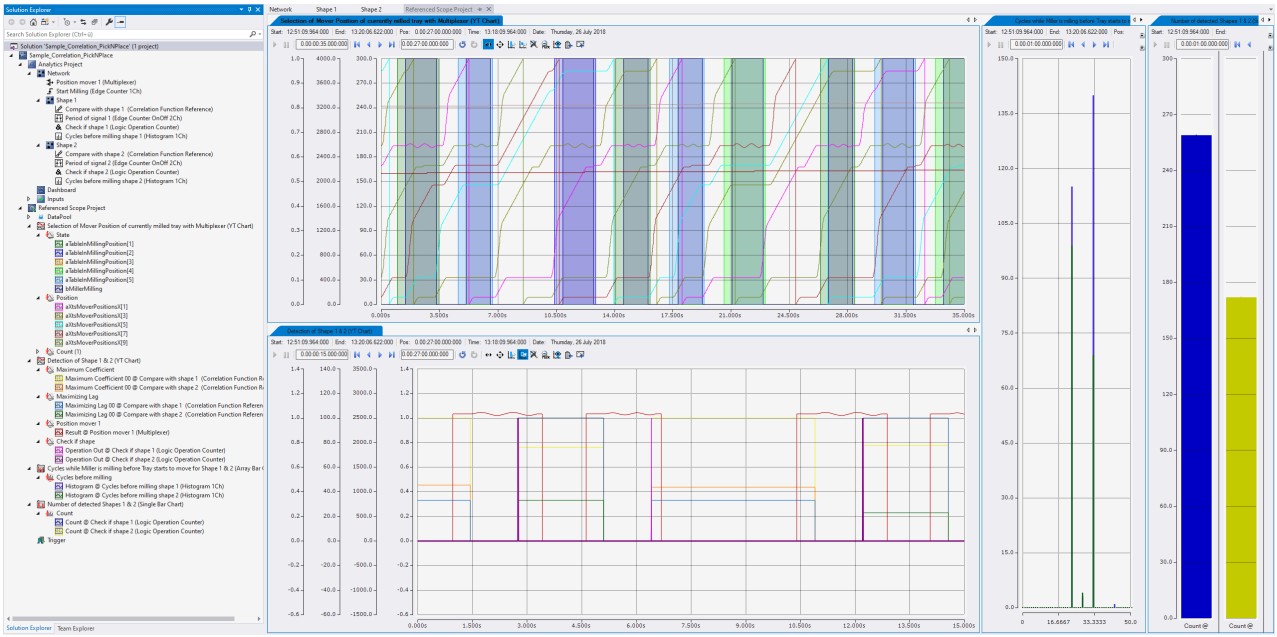
Um nur dann neue Ergebnisse zu erhalten, wenn gefräst wird, ist der *Enable Execution*-Eingang des Korrelationsbausteins verknüpft. Hier wird der Ausgang *Is On* des nachfolgenden Bausteins **Period of Signal 1 (Edge Counter OnOff 2Ch)** verknüpft. Dieser wird TRUE, wenn der Fräser eingeschaltet wird und wieder FALSE, wenn alle Werte aus der Datei einmal für die Korrelationsberechnung genutzt wurden und der Korrelationsbaustein seine Ausgänge aktualisiert hat. Der Baustein **Check if shape 1 (Logic Operation Counter)** gibt TRUE zurück, wenn der maximale Pearson-Korrelationskoeffizient nach dem Vergleich der Sequenzen größer als 0.98 ist. Auf diesem Weg wird auch die Anzahl der gefrästen Teile mit der Form 1 gezählt.

Für die Fräsvorgänge, die der ersten Form zugeordnet werden, wird im Baustein **Cycles before Milling Shape 1 (Histogram 1Ch)** die Verteilung der Anzahl an Zyklen ermittelt, die zwischen Einschalten des Fräsers und Abfahren der Kontur gelegen haben



Die Korrelationsberechnung der zweiten Fräsform erfolgt analog in einem separaten Netzwerk.

In den unten dargestellten Scope-Plots werden die Ergebnisse aus der Analyse visualisiert.



Im oberen linken Chart sind die Eingänge und der *Result*-Ausgang des Multiplexers dargestellt. Erkennbar ist, dass der Mover eine sinusförmige Bahn abfährt, wenn sich die entsprechende Ablage in der Position der Fräse befindet.

Im unteren linken Chart ist der *Result*-Ausgang des Multiplexers rot abgebildet. Die beiden Bausteine **Check if shape 1 (Logic Operation Counter)** und **Check if shape 2 (Logic Operation Counter)** haben jeweils einen Ausschlag (pink und lila), wenn die korrelierte Sequenz einen maximalen Koeffizienten (gelb und orange) größer als 0.98 erzielt hat. Im gezeigten Ausschnitt kann überprüft werden, dass diese Ergebnisse mit den verschiedenen Kurven des *Result*-Ausgangs des Multiplexers übereinstimmen. In blau und grün sind außerdem die entsprechenden *Maximizing Lag*-Ausgänge der Korrelationsbausteine dargestellt.

Im linken der beiden Bar Charts sind die *Maximizing Lag*-Ausgänge der Korrelationsbausteine dargestellt. Sie ergeben sich aus den Ergebnissen der Histogramm-Bausteine. Beide Formen erzeugen große Ausschläge bei einem Lag von 23 sowie einem Lag von 33 und einen kleinen Ausschlag bei einem Lag von Lag 28. Die erste Form erzeugt einen einmaligen Lag von 43 Zyklen.

Das rechte Bar Chart zeigt die Anzahl von Teilen die Form 1 und 2 zugeordnet wurden. Der ersten Form wurden 259 und der zweiten Form 172 Teile zugeordnet.

Downloads

- https://infosys.beckhoff.com/content/1031/TE3500_TC3_Analytics_Workbench/Resources/12177935499.zip

9 Anhang

9.1 FAQ – Häufig gestellte Fragen und Antworten

In diesem Abschnitt werden häufig gestellte Fragen beantwortet, um Ihnen die Arbeit mit der TwinCAT Analytics Workbench zu erleichtern. Falls Sie weitere Fragen haben, wenden Sie sich bitte an unser Support-Team support@beckhoff.com.

1. Werden in TwinCAT Measurement Produkten Open Source Softwarekomponenten verwendet? [▶ 570]

Werden in TwinCAT Measurement Produkten Open Source Softwarekomponenten verwendet?

Ja, es werden verschiedene Open Source Komponenten verwendet. Eine Auflistung dieser samt Lizenzbedingung finden Sie im Verzeichnis *…\TwinCAT\Functions\TwinCAT Measurement\Legal*.

Trademark statements

Beckhoff®, ATRO® , EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar®, and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Chrome, Chromium and Google are trademarks of Google LLC.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

Mehr Informationen:
www.beckhoff.com/TE3500

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com