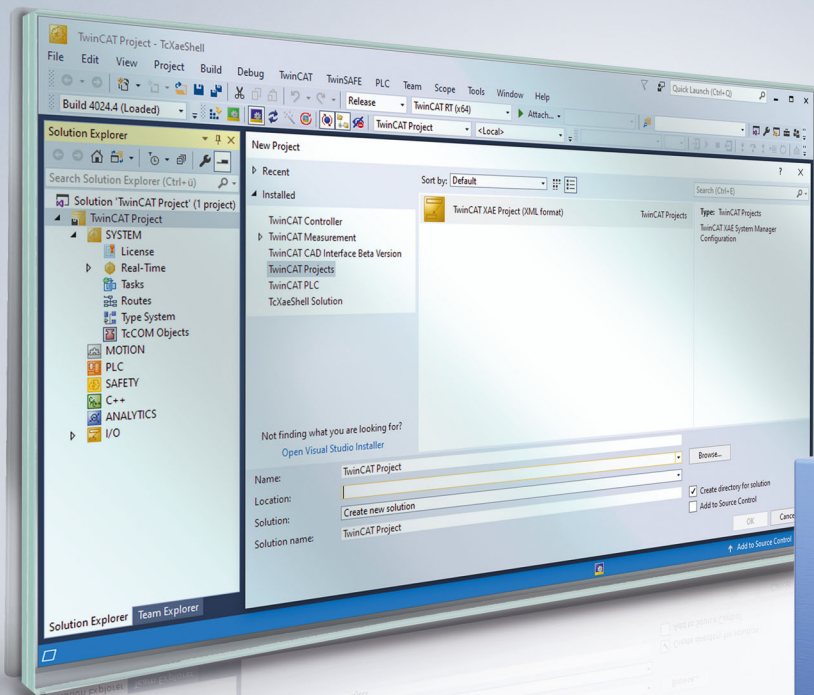


BECKHOFF New Automation Technology

Handbuch | DE

TE3520

TwinCAT 3 | Analytics Service Tool



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	14
4	Analytics Workflow – Erste Schritte.....	17
4.1	Aufzeichnung von Maschinendaten	17
4.2	Kommunikation	20
4.3	Historisierung von Daten.....	21
4.4	Analyse der Daten.....	28
5	Technische Einführung	34
5.1	Grundkonzept.....	34
6	Konfiguration.....	36
6.1	Datenquellen	36
6.1.1	Wizard	38
6.2	Data Scout	44
6.2.1	Data Track Editor	44
6.3	Editor Basics	58
6.3.1	Netzwerke	58
6.3.2	Enable Disable	62
6.3.3	Static Values	64
6.4	Netzwerke	65
6.4.1	Netzwerke als Template.....	67
6.5	Algorithmen	69
6.5.1	Analytics - Base	71
6.5.2	Analytics - Classification	123
6.5.3	Analytics - Clustering	148
6.5.4	Analytics - Compare.....	157
6.5.5	Analytics - Math.....	180
6.5.6	Analytics - Training Base	190
6.5.7	Analytics - XTS.....	193
6.5.8	Analytics - WT	200
6.5.9	Analytics - XY Path Analysis.....	202
6.5.10	Analytics - Statistics	212
6.5.11	Analytics - Erweiterung der Algorithmik	237
6.5.12	Analytics - Visualization Only	292
6.5.13	Analytics – Reporting	299
6.6	Interaktion mit Scope	303
6.6.1	Scope Konfiguration im Netzwerk Template.....	308

6.7	Arbeit mit historischen Daten	311
7	Anhang	314
7.1	FAQ – Häufig gestellte Fragen und Antworten	314

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, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte 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 weiter entwickelt.

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®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® 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 Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

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

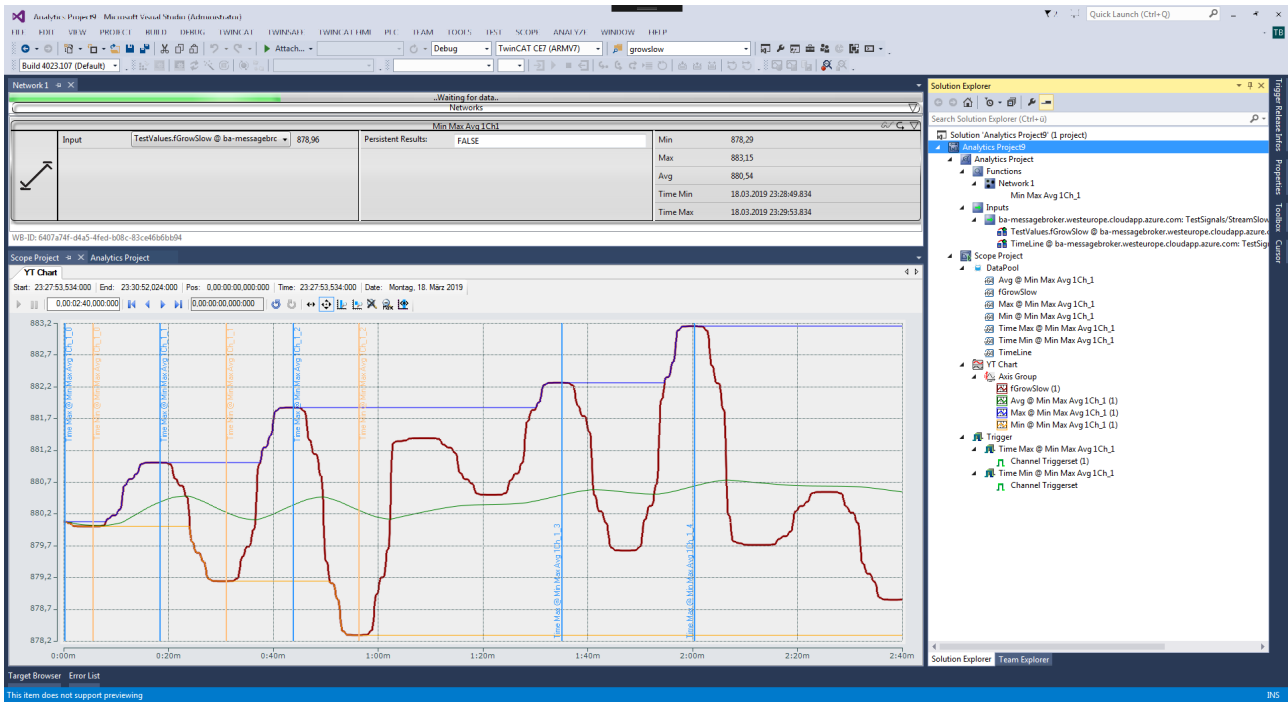
Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und 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



Das TwinCAT 3 Analytics Service Tool wird für die Inbetriebnahme von Maschinen und für Servicetechniker verwendet. Über den IoT-Anschluss oder offline über eine Dateikopie können Live- und historische Daten für eine Analyse abgerufen werden. Die Konfiguration der Analyse findet in Microsoft Visual Studio statt, wo dem Benutzer eine Toolbox von Algorithmen zur Verfügung steht, um eine Lebensdauer-, Zykluszeit-, Hüllkurven-, oder Stückzähler-Analyse zu realisieren. Die Ausgänge der Algorithmen können dabei als Eingänge weiterer Algorithmen genutzt oder als Ergebnis, direkt im grafischen Editor, ausgegeben werden. Mit einer parallel laufenden Aufzeichnung mit dem TwinCAT Scope können Signalverläufe sehr einfach visualisiert werden. Analyseergebnisse kann der Benutzer per Drag-and-drop aus dem Analytics-Konfigurator in das Charting-Tool ziehen, um die signifikanten Stellen im Datenstrom zu markieren. Diese Wechselwirkung zwischen den Produktkomponenten bietet insbesondere Vorteile bei der Diagnose von Maschinenverhalten und kann Optimierungspotenziale offenlegen. Durch die verwendeten IoT-Technologien erlangt der Anwender eine Ortsunabhängigkeit, welche es Servicetechnikern ermöglicht, von nahezu jedem Ort eine Diagnose der Anlagen und Maschinen vorzunehmen.

Komponenten

- Analytics-Konfigurator
- Base Analytics-Algorithmus
- Analytics Storage Provider Recorder
- TwinCAT Scope (TE1300 und TF3300)
- IoT-Connectivity

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.

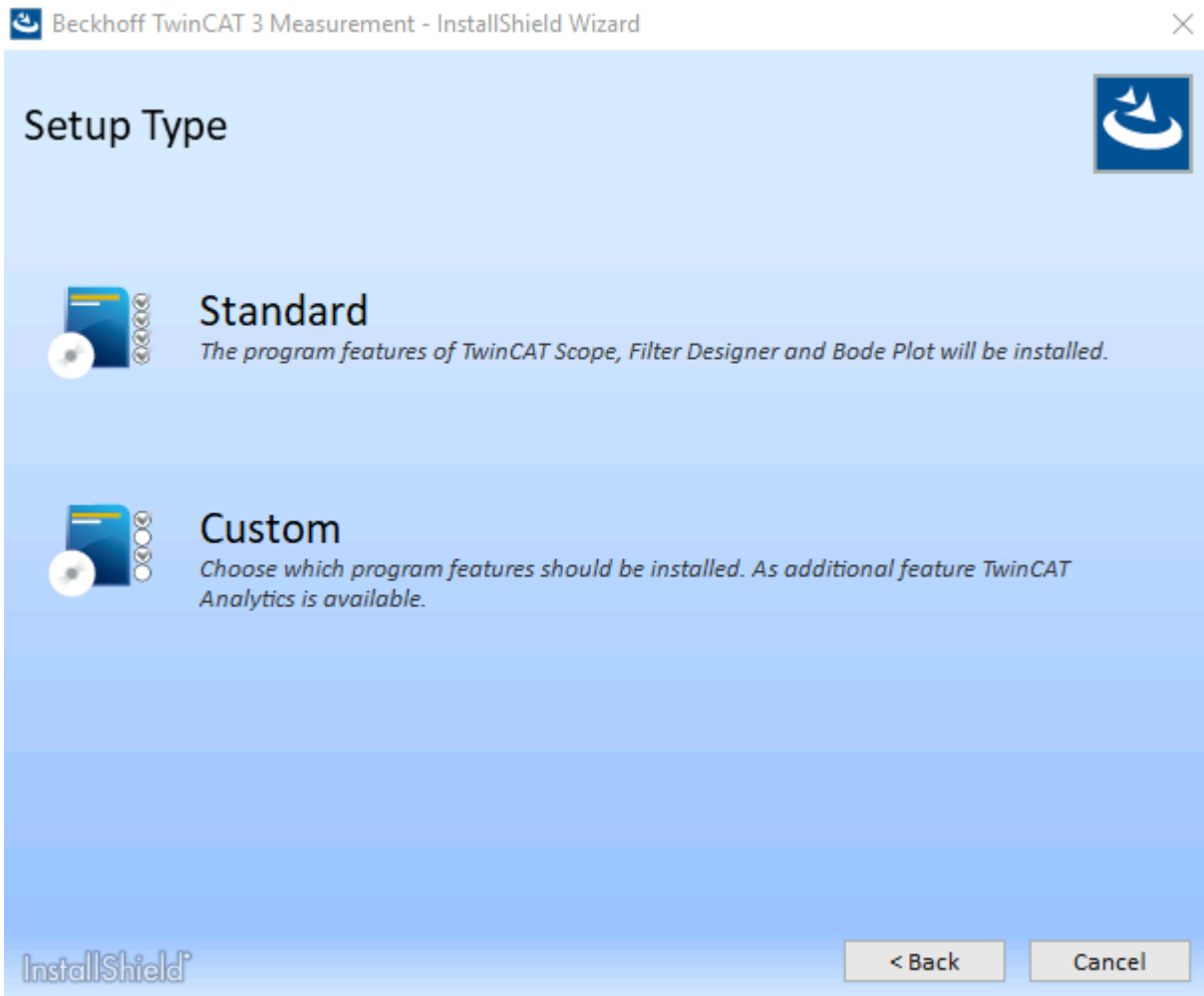
3.2 Installation und Lizenzierung

Das Setup von TwinCAT Analytics ist Teil des Setups der TwinCAT Measurement Suite. Sie haben die Möglichkeit zwischen zwei Setup-Varianten zu wählen:

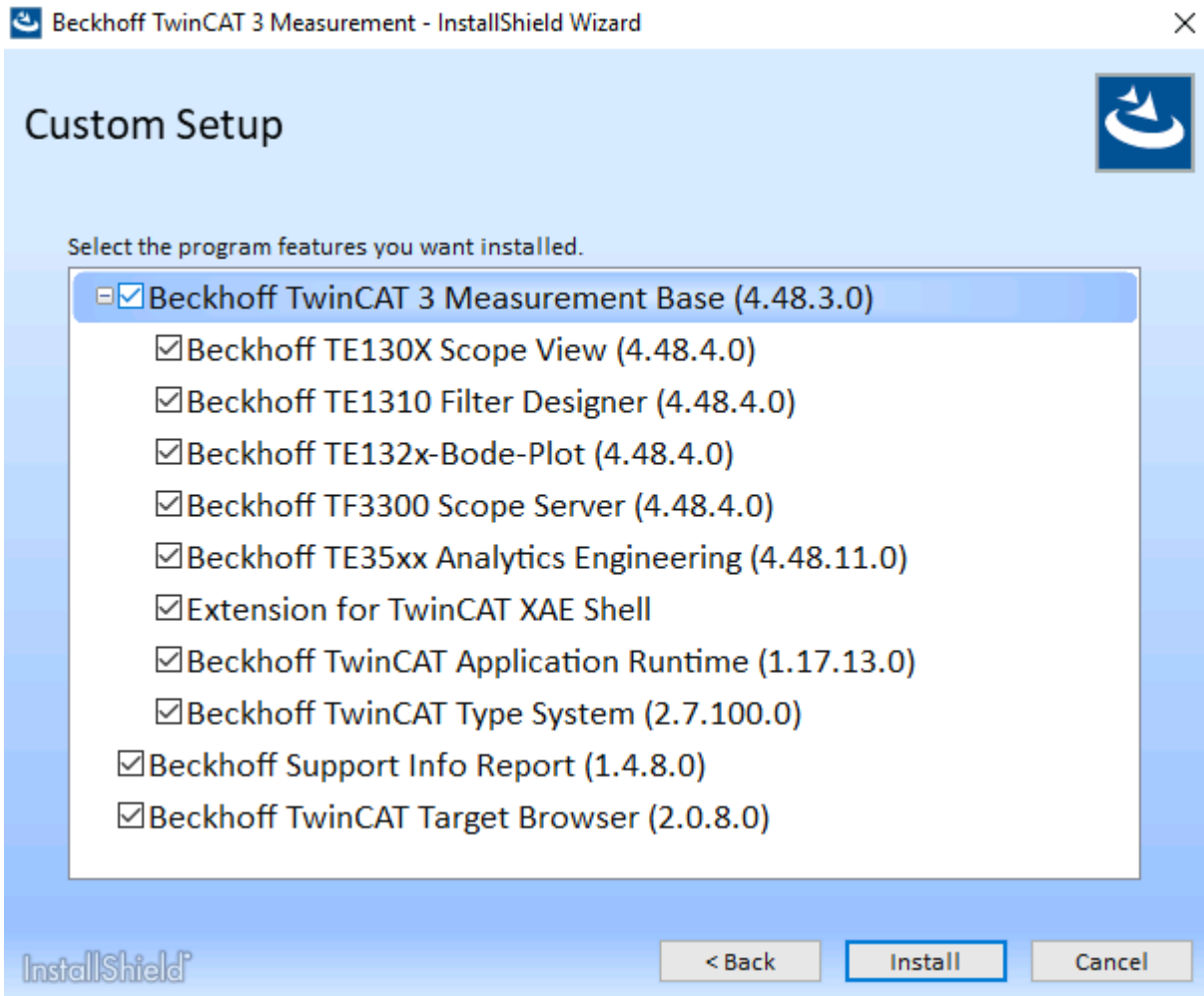
Das TC3-Measurement-Full Setup bringt die TwinCAT Measurement Komponenten und eine Visual Studio® Umgebung mit.

Das TC3-Measurement-Update Setup bringt dagegen ausschließlich die Measurement Komponenten mit.

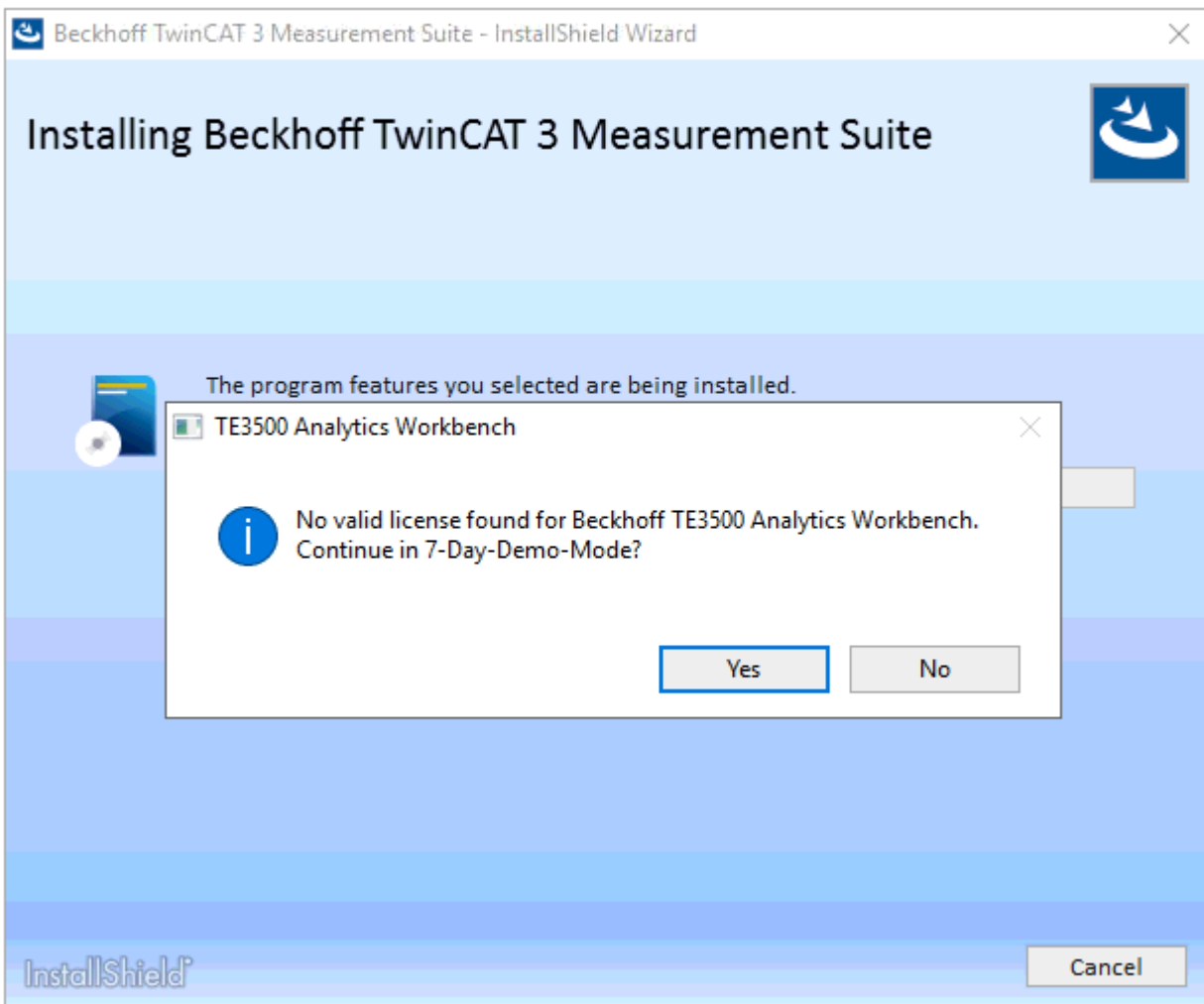
Nach der Bestätigung der Lizenzbedingungen haben Sie die Wahl zwischen Standard und Custom. Bei Standard werden automatisch alle Measurement-Komponenten inklusive TwinCAT Analytics Engineering installiert.



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**

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

- **TwinCAT 3-Lizenzen für Nicht-Beckhoff-Geräte**

i 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!

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

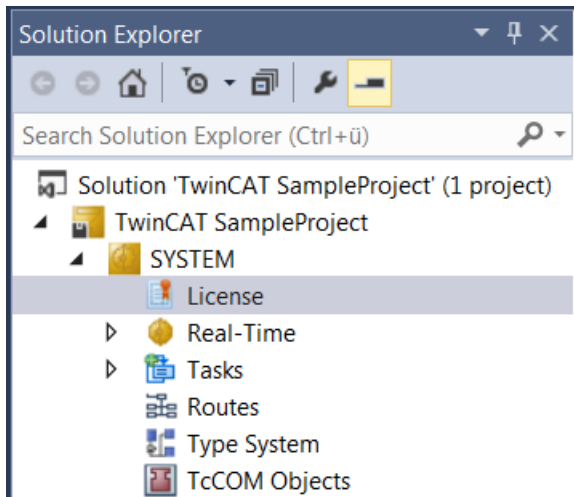
Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT 3 Lizenzierung](#)“.

Lizenzierung der 7-Tage-Testversion einer TwinCAT 3 Function



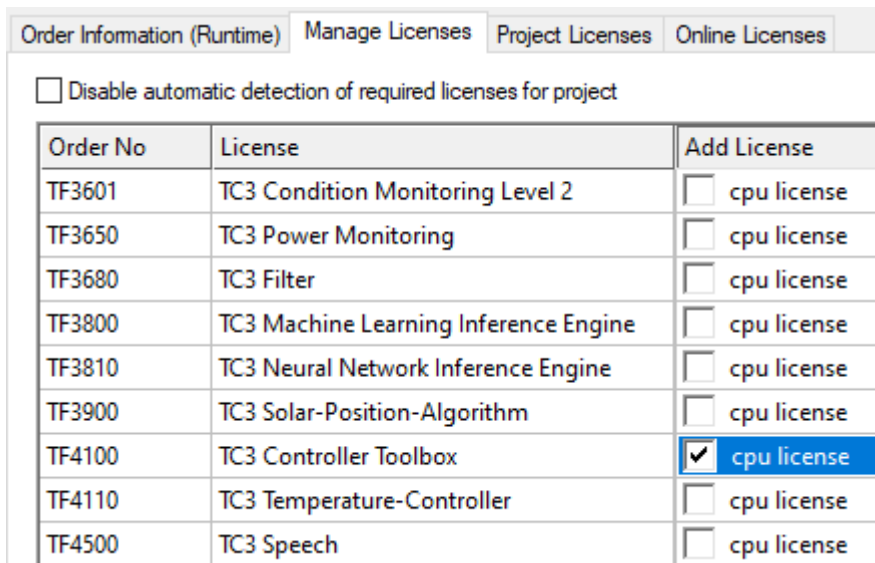
Eine 7-Tage-Testversion kann nicht für einen [TwinCAT-3-Lizenz-Dongle](#) freigeschaltet werden.

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.

Order Information (Runtime) Manage Licenses Project Licenses Online Licenses

License Device: Target (Hardware Id) Add...

System Id: 2DB25408-B4CD-81DF-5488-6A3D9B49EF19 Platform: other (91)

License Request

Provider: Beckhoff Automation Generate File...

License Id: Customer Id:

Comment:

License Activation

7 Days Trial License... License Response File...

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

Enter Security Code

Please type the following 5 characters: Kg8T4

OK Cancel

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 – Erste Schritte

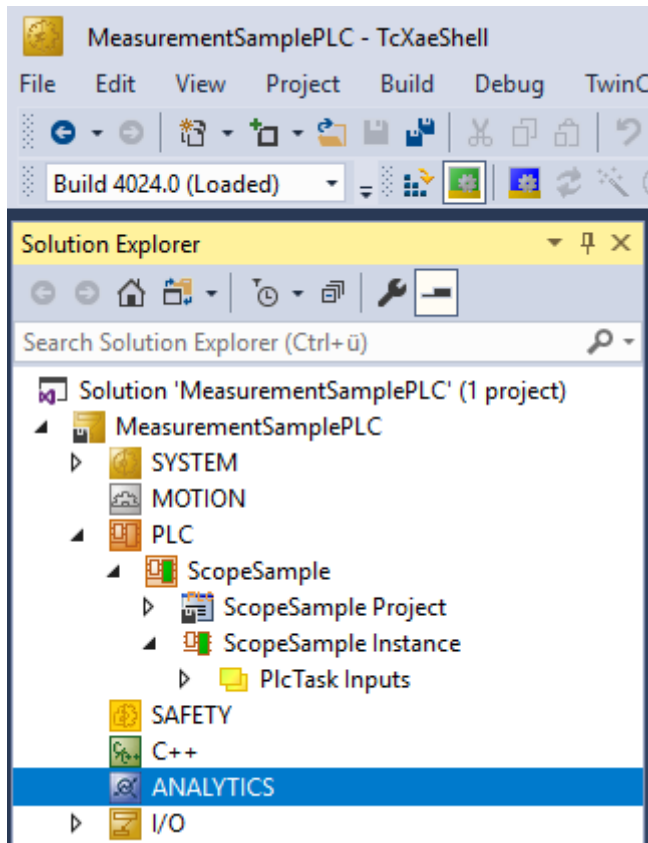
Diese Dokumentation präsentiert Schritt für Schritt den vollständigen TwinCAT Analytics Workflow. Von der Datenerfassung über die Kommunikation und Historisierung bis hin zur Auswertung und Analyse der Daten sowie zur Präsentation der Daten im webbasierten Dashboard.

4.1 Aufzeichnung von Maschinendaten

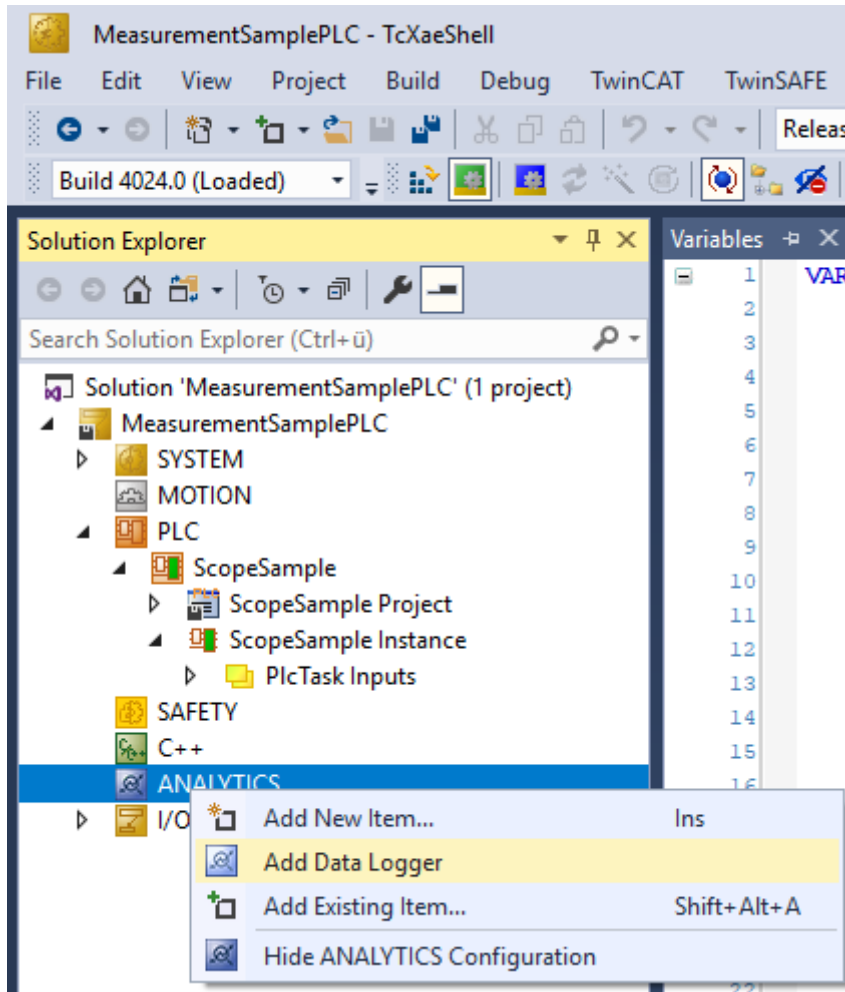
Maschinenseitig ist der Analytics Logger die Aufzeichnungseinrichtung von Prozessdaten des Maschinenabbaus, SPS, NC usw. Der Logger arbeitet im Echtzeitkontext von TwinCAT.

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.

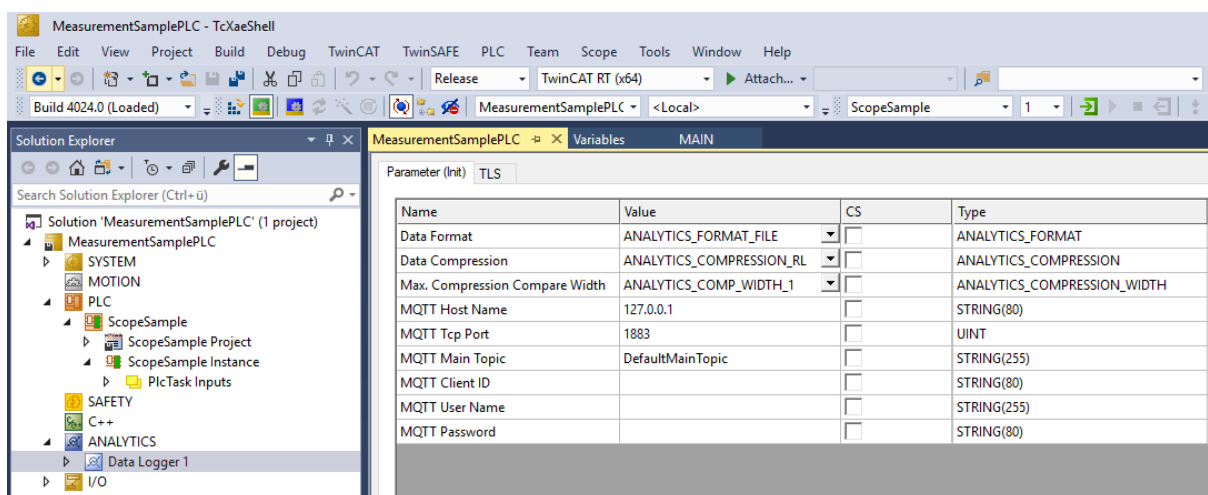
- Um den Analytics Logger zu konfigurieren, müssen Sie in Ihrem vorhandenen TwinCAT-Projekt zum Analytics-Baumknoten navigieren.



- 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.

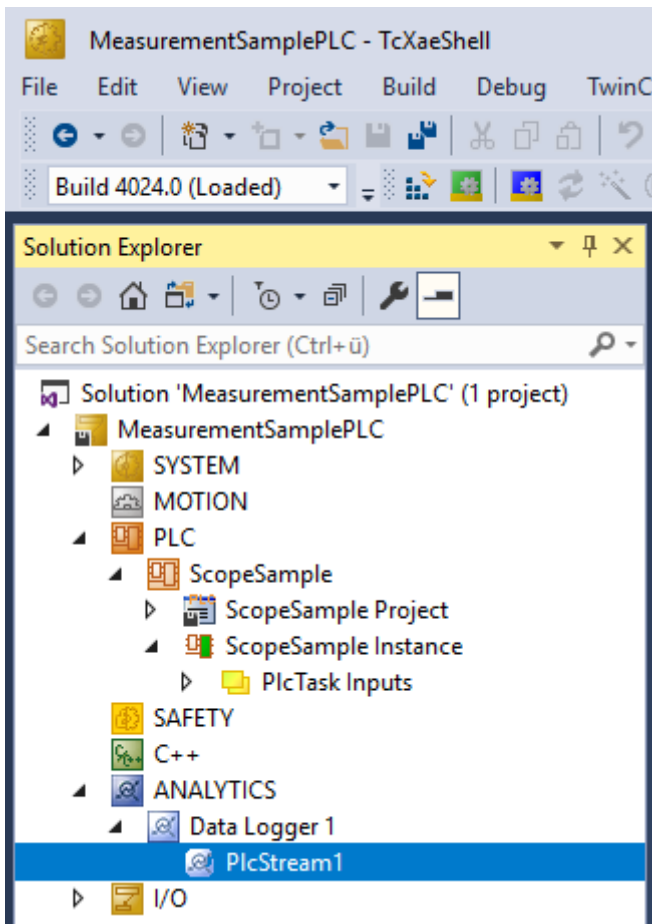
-FILE-Format: Der Analytics Logger speichert die Daten in lokalen Binärdateien und alle anderen Einstellungen sind nicht mehr notwendig. Die Dateien werden in *C:\TwinCAT\3.1\Boot\Analytics* gespeichert.

-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.

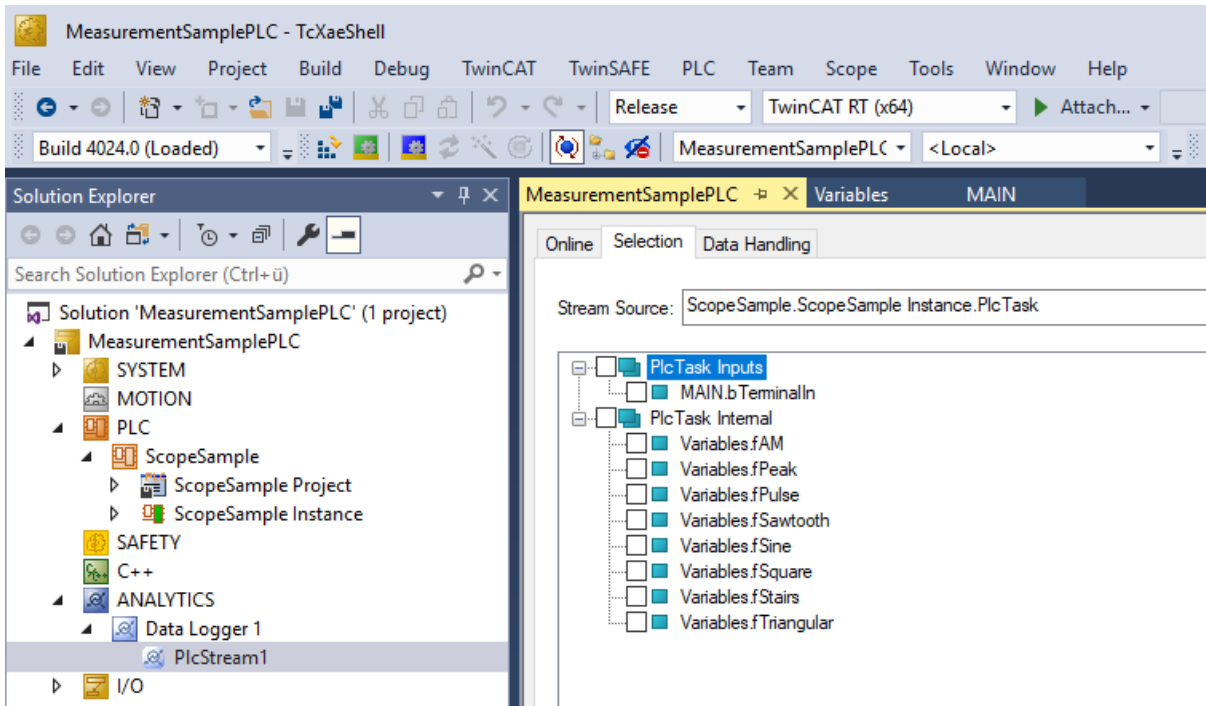
-Auf der Registerkarte **TLS** (Transport Layer Security) können die Sicherheitseinstellungen konfiguriert werden. TLS ist ein sicherer Kommunikationskanal zwischen Client und Server. Bei Verwendung von Zertifikaten ist der TCP-Port 8883 ausschließlich für MQTT über TLS reserviert! Der Analytics Logger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

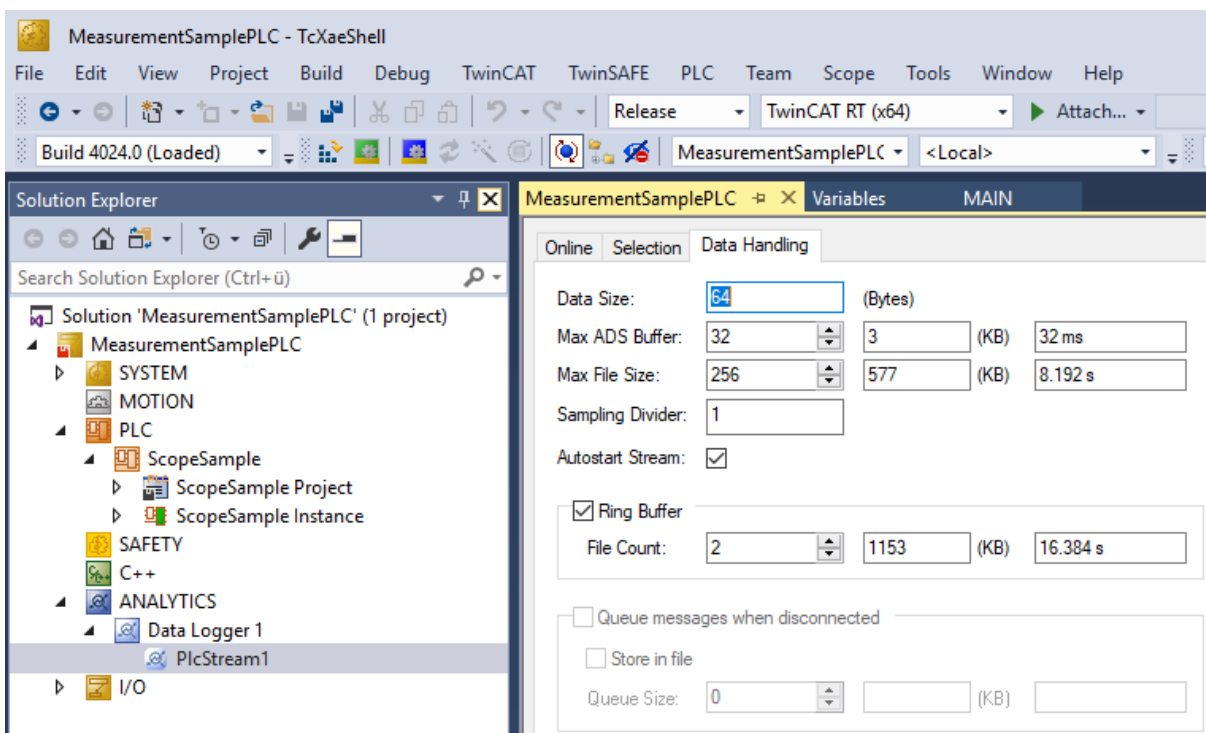


Ein zusätzlicher Geräte-Stream wird angezeigt, wenn Ihre Konfiguration ein EtherCAT-Prozessabbild liefert.

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

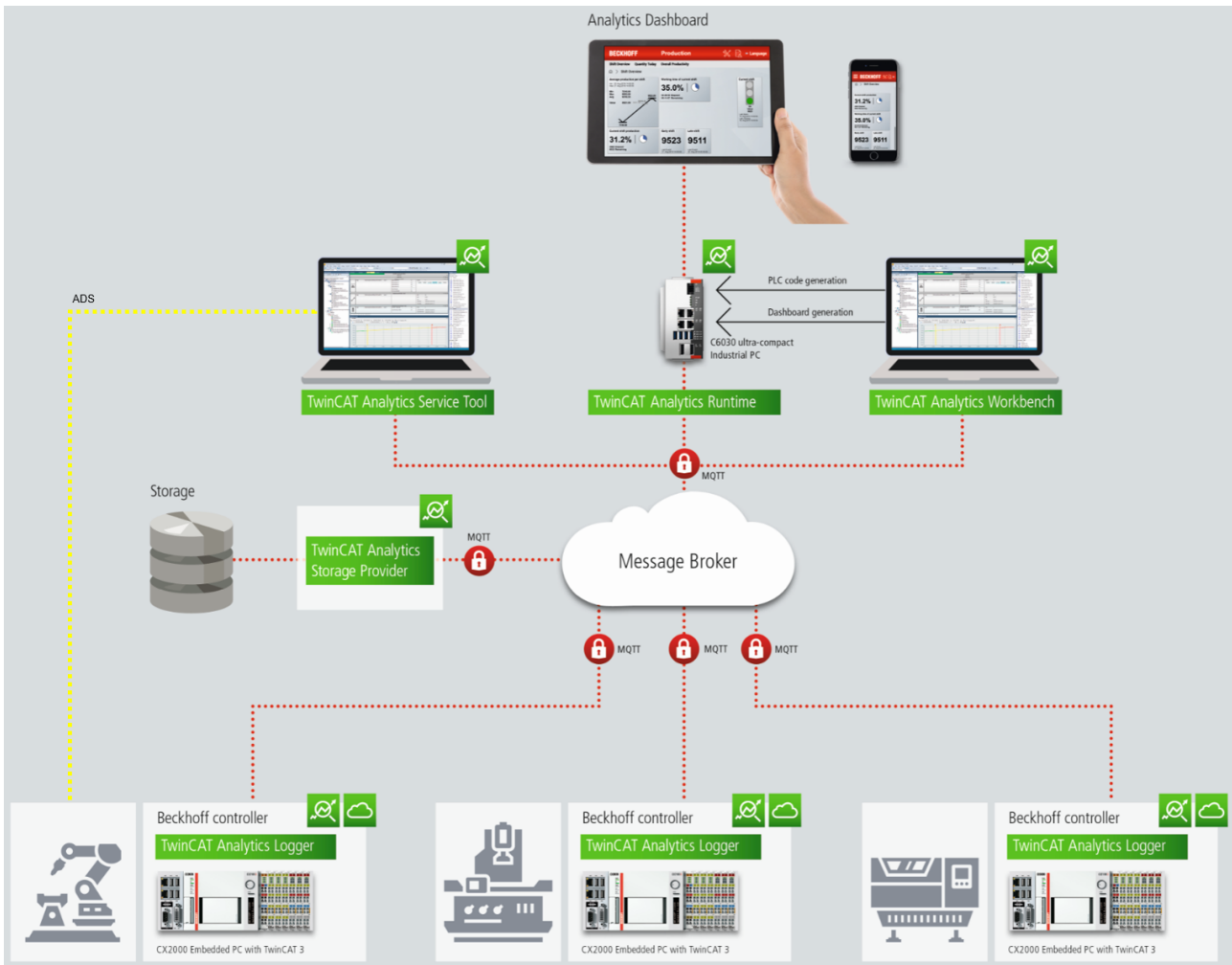


- Schließlich ist es im Karteireiter **Data Handling** möglich, die Paketgröße für die Frames zu ändern oder den Ringpuffer für Verbindungsunterbrechungen und Datei zu konfigurieren.



4.2 Kommunikation

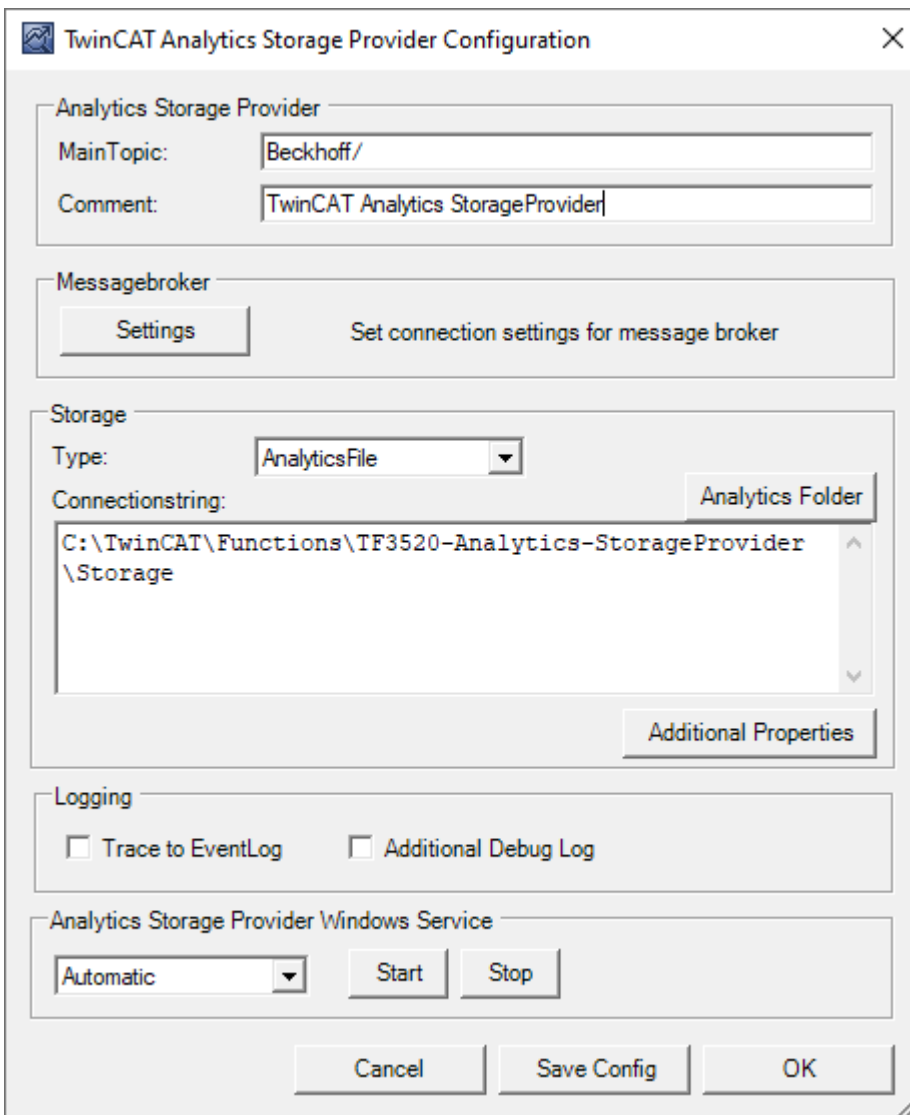
Derzeit kann der Analytics-Workflow vollständig über MQTT abgebildet werden. Die Engineering-Tools können auch über ADS auf die Daten der Maschinen zugreifen und Analysen durchführen.



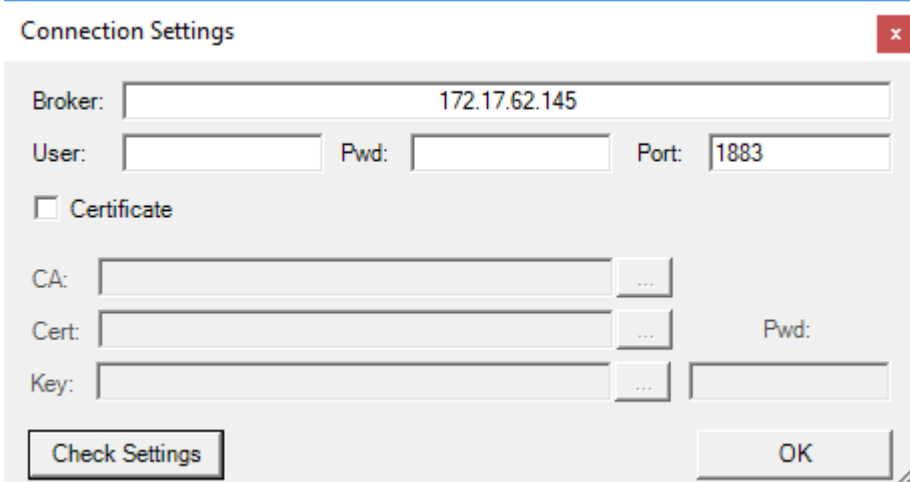
Wenn man für das IoT-Kommunikationsprotokoll MQTT wählt, muss irgendwo im Netzwerk ein nativer MQTT-Message-Broker eingerichtet werden (VM in einem Cloud-System ist ebenfalls möglich). Dieser Message-Broker sorgt für eine Entkopplung der verschiedenen Anwendungen im Analytics-Workflow.

4.3 Historisierung von Daten

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



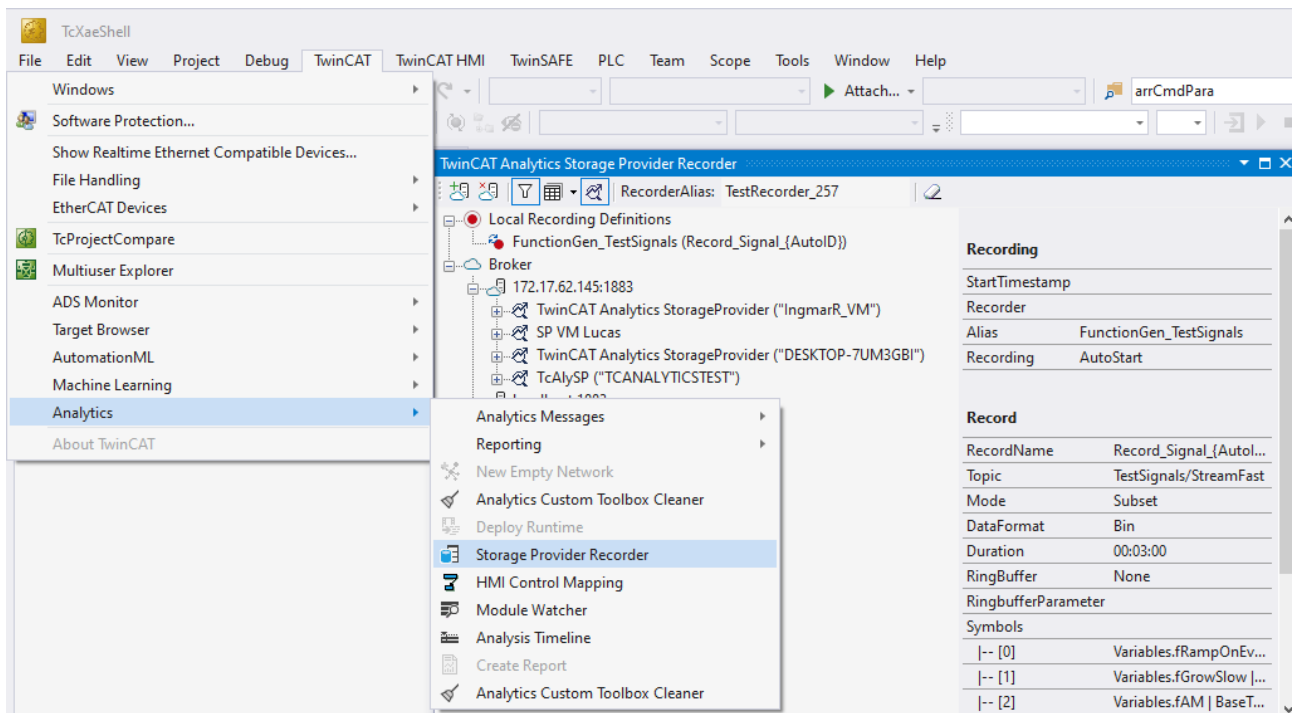
Der Hauptteil des Topics kann in der Konfiguration ebenso festgelegt werden wie der Kommentar, der zur Identifizierung dient, wenn mehr als ein Storage Provider beim Message-Broker registriert ist.



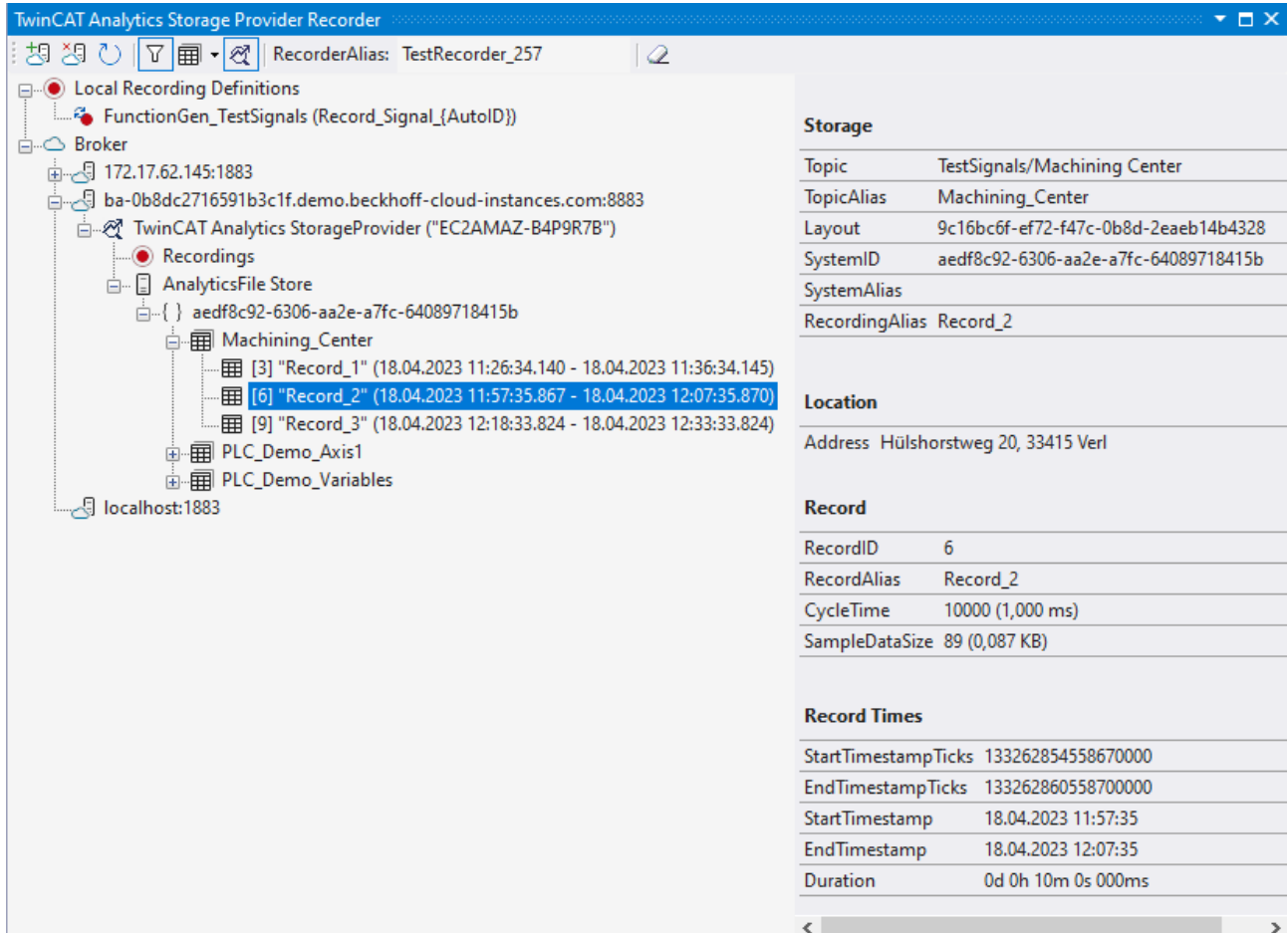
Sie können die Message-Broker-Einstellungen vornehmen und sich für einen Speichertyp entscheiden:

- Analytics File (Binärdatei)
- Microsoft SQL (Binär)
- Microsoft Azure Blob (Azure-Cloud erforderlich)

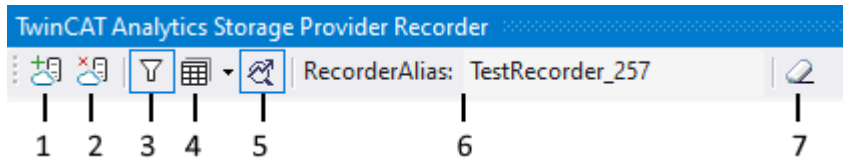
Zuletzt können Sie die Konfiguration speichern und den Dienst starten. Der nächste Schritt besteht in der Konfiguration der spezifischen Aufzeichnung. Hierzu sollten Sie in Ihrer Entwicklungsumgebung den **Storage Provider Recorder** auswählen.



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 Providern und historisierten Daten werden übersichtlich dargestellt.



Symbolleiste

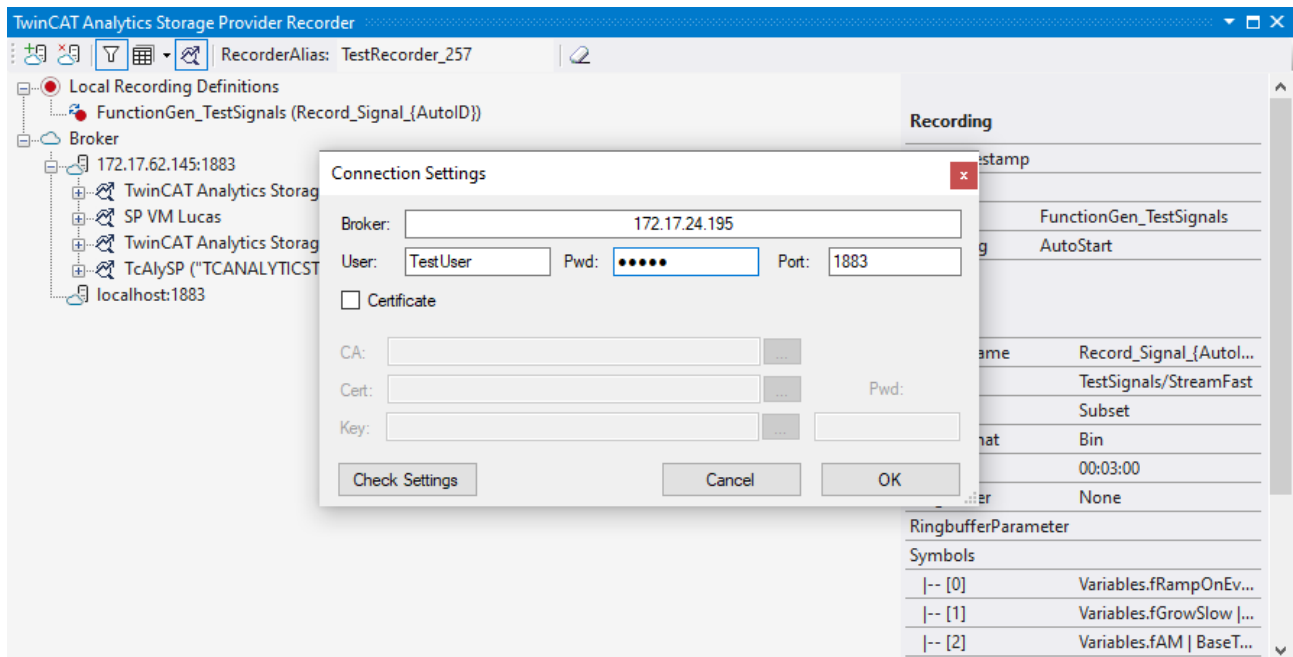


1	Neuen Broker hinzufügen
2	Ausgewählten Broker entfernen
3	Anzeige Filter (Alle / Meine aktiven Aufnahmen anzeigen)
4	Anzeige Art wählen (Alias / Topics)
5	Offline Analytics Storage Provider anzeigen / ausblenden
6	„RecorderAlias“ – Gruppierungsname von Aufnahmen
7	Meldungen aus Fehlerliste entfernen

Recorder Fenster einrichten

Vergeben Sie zunächst einen „RecorderAlias“. Dieser hilft die gestarteten Aufnahmen zu gruppieren und seine selbst gestarteten wieder zu finden. Mit dem Filter können auch fremd gestartete Aufnahmen angezeigt werden.

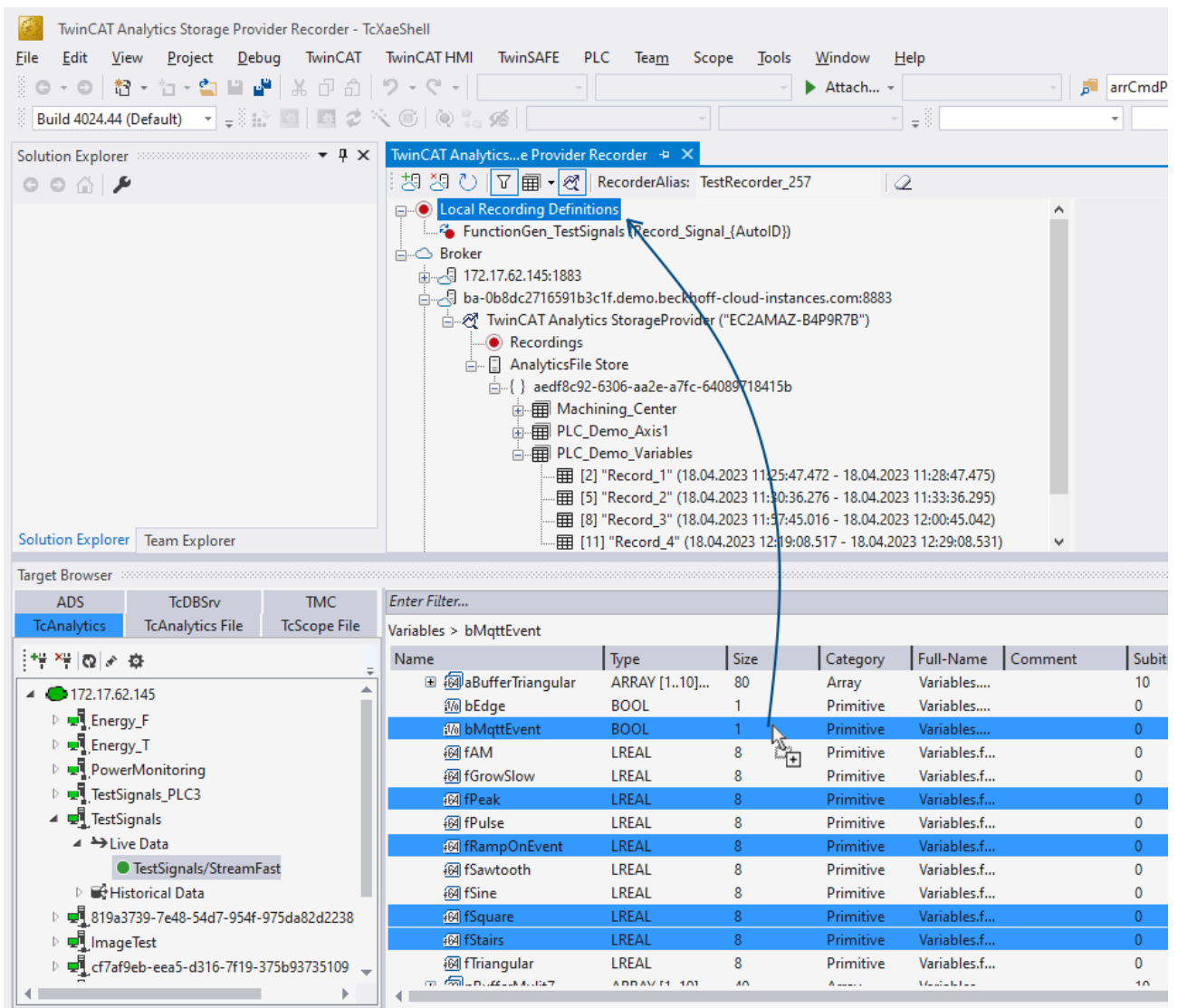
Danach können ein oder mehrere Broker eingerichtet werden. Dies geschieht über die schon bekannte Eingabemaske für MQTT-Verbindungseigenschaften.



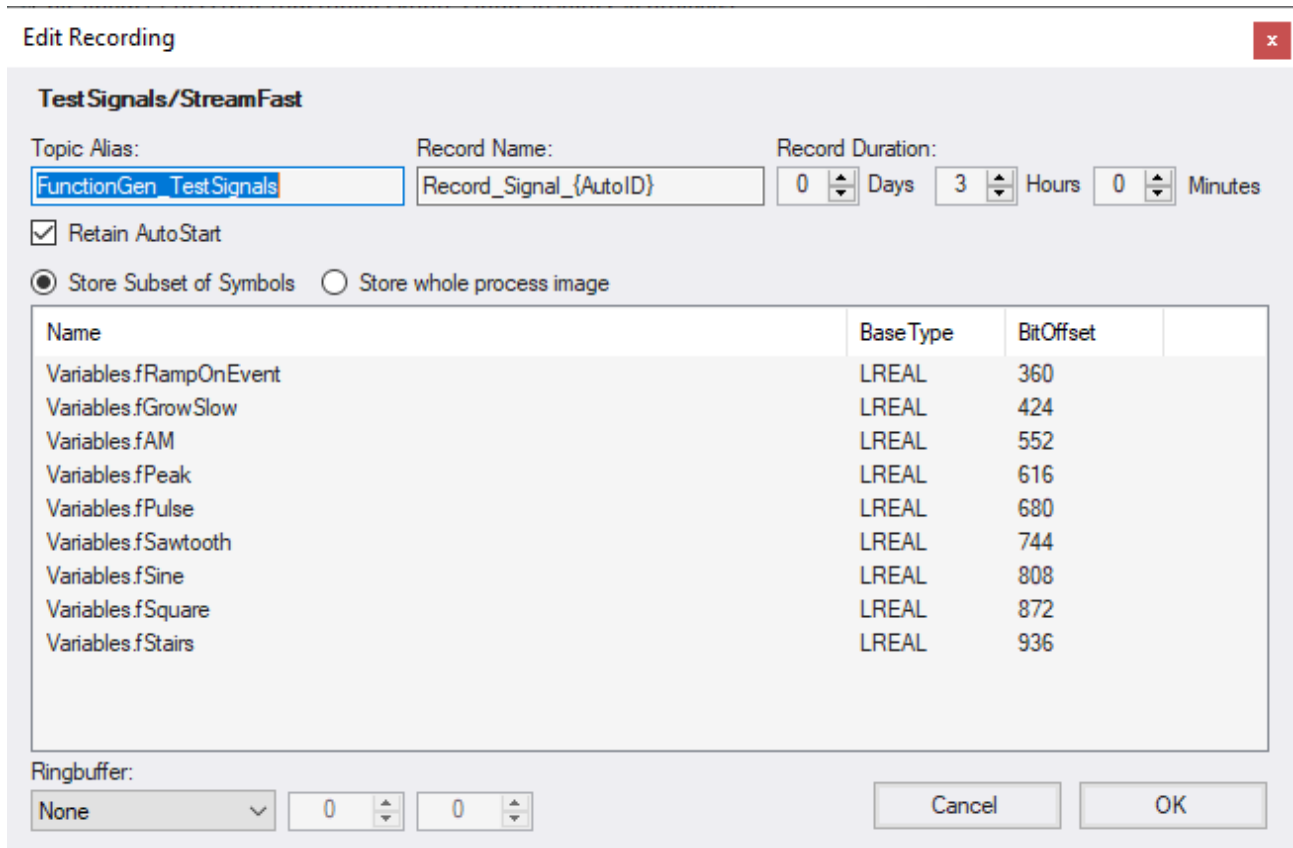
Sobald eine Verbindung zum Broker hergestellt werden konnte, werden alle darauf verbundenen Analytics Storage Provider aufgelistet.

Aufnahmedefinition erzeugen

Zur Konfiguration der Aufzeichnung wählen Sie Ihr Ziel im Target Browser aus. Klicken Sie auf **Live Data** und wählen Sie eine oder mehrere Variablen durch Mehrfachauswahl aus und ziehen Sie sie per Drag-and-Drop auf den **Local Recording Definitions** Knoten im Recorder Fenster.



Im Recorder können Sie die ausgewählten Variablen oder das vollständige Quell-Prozessabbild der Variablen hinzufügen.

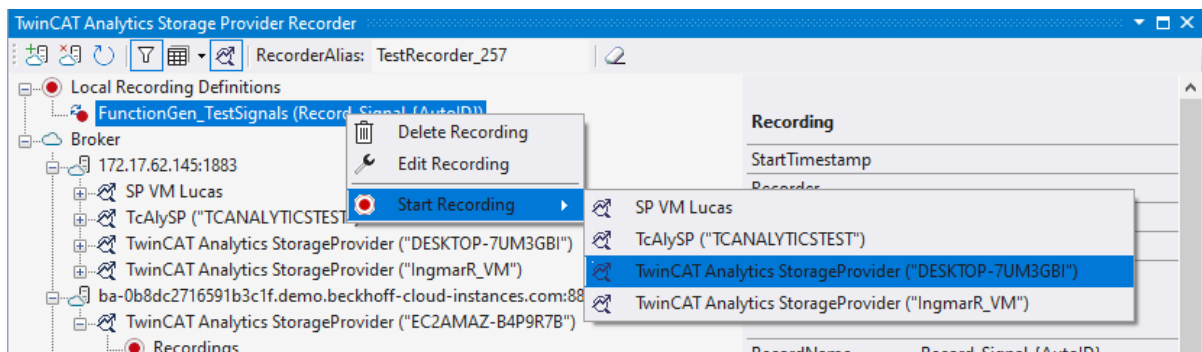


Sie können auch Aufzeichnungsamen und eine Dauer konfigurieren (anderenfalls läuft die Aufzeichnung endlos, bis sie manuell gestoppt wird). Ein Ringpuffer kann nach Storage oder Zeit eingestellt werden.

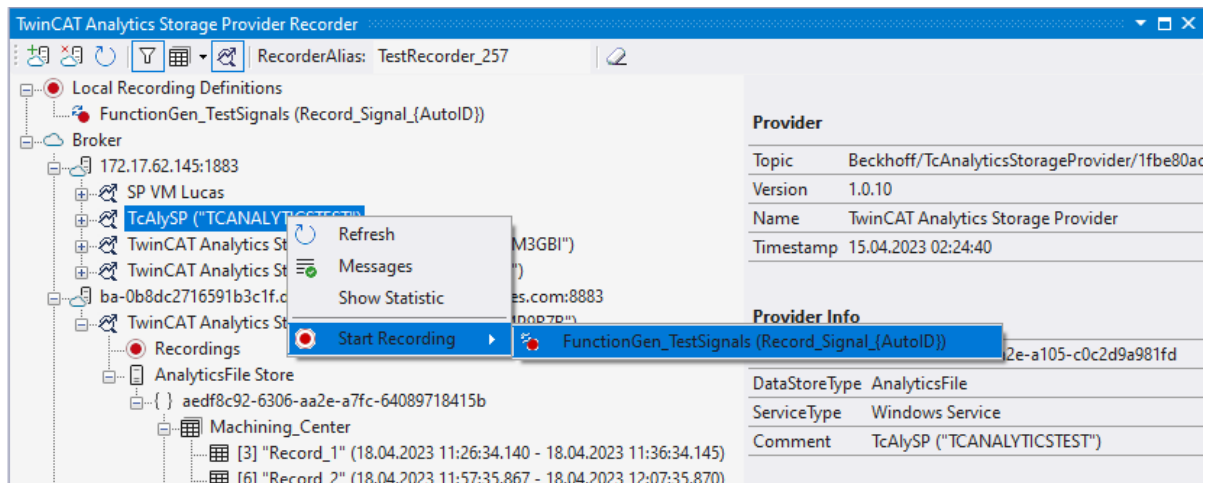
Aufnahme starten

Es gibt drei verschiedene Wege, um eine Aufnahme zu starten.

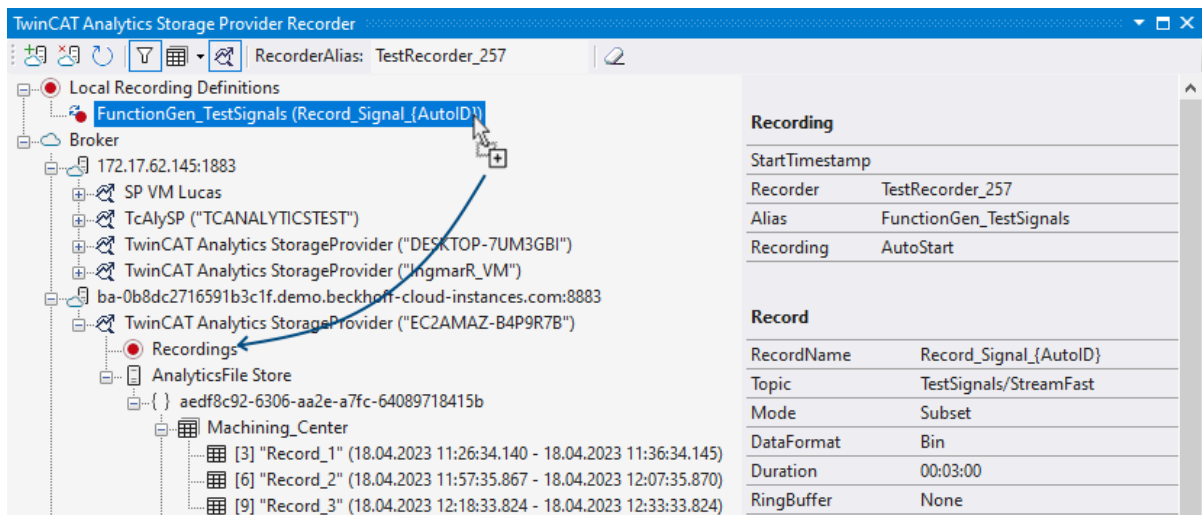
1. Über das Kontextmenü an einem lokalem Aufnahmedefinition Knoten. Hier kann der gewünschte Analytics Storage Provider ausgewählt werden, welcher die Daten aufzeichnen soll. Es werden nur die Storage Provider angezeigt, die auch Zugriff auf das ausgewählte Topic haben.



- Über das Kontextmenü am Analytics Storage Provider Knoten. Hier kann die gewünschte Aufnahme­definition ausgewählt werden. Es werden nur die Definitionen angezeigt, die auch vom Analytics Storage Provider verarbeitet werden können.

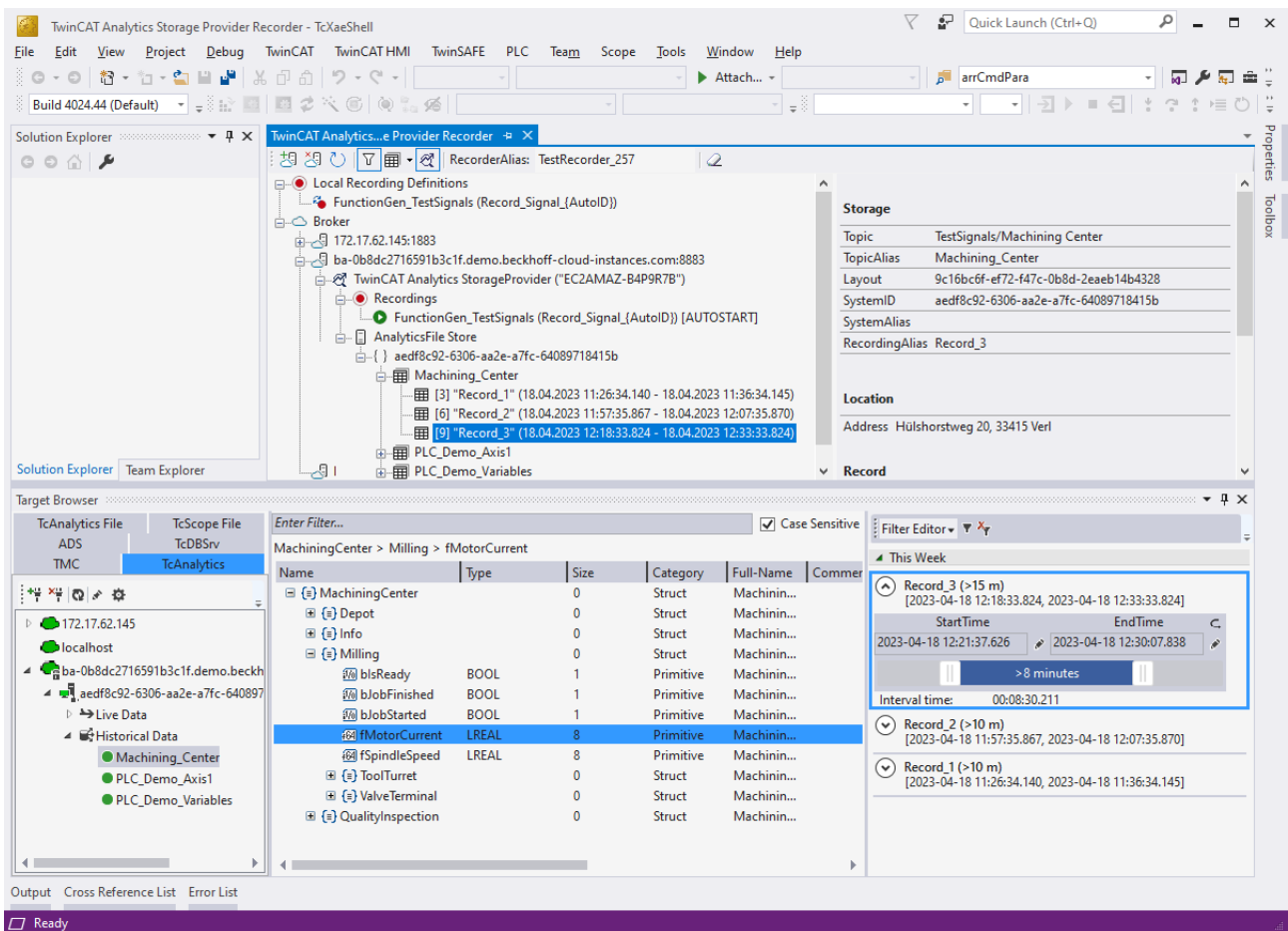


- Per Drag-and-Drop des lokalen Aufnahme­definition Knoten. Die gewünschte Aufnahme­definition kann auf den gewünschten Analytics Storage Provider per Drag-and-Drop gestartet werden. Sollte der Storage Provider die Aufnahme­definition nicht ausführen können, wird eine Fehlermeldung in der Fehlerliste ausgegeben.



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.

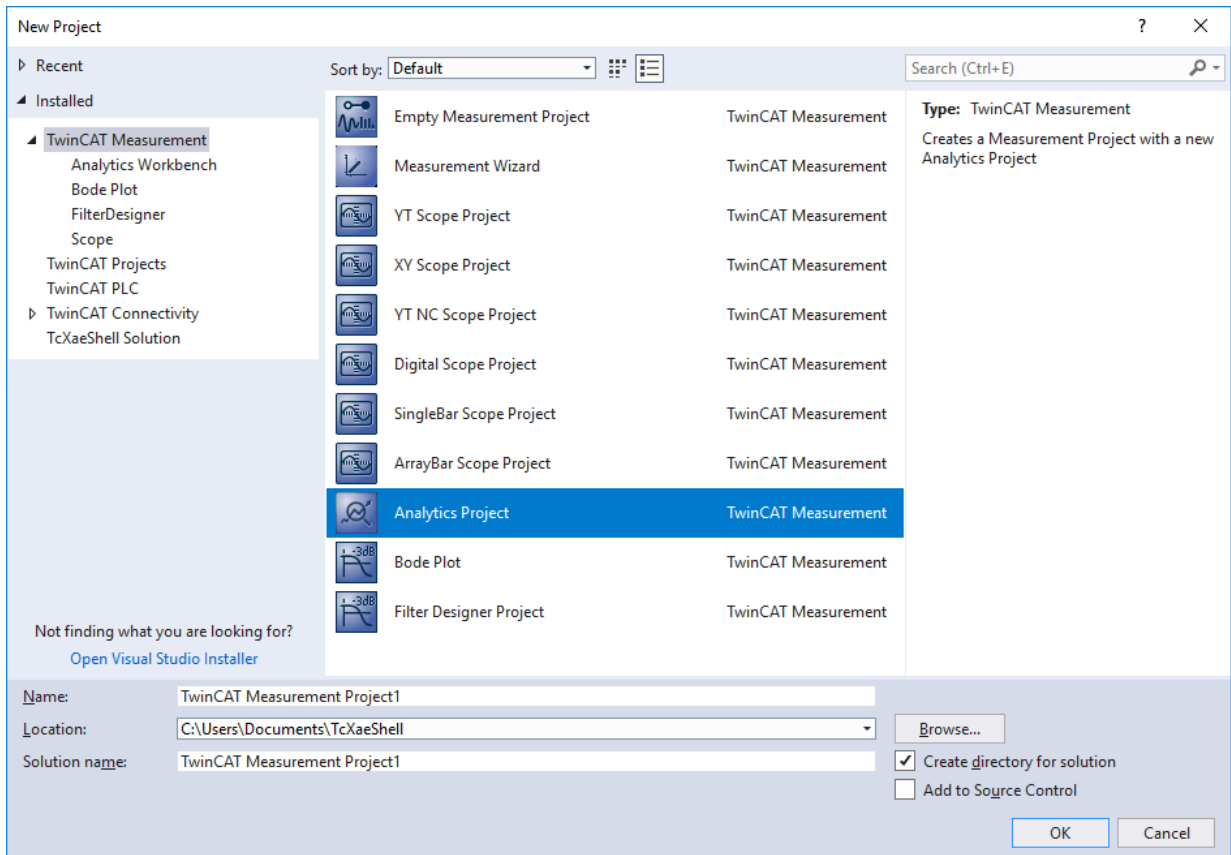


4.4 Analyse der Daten

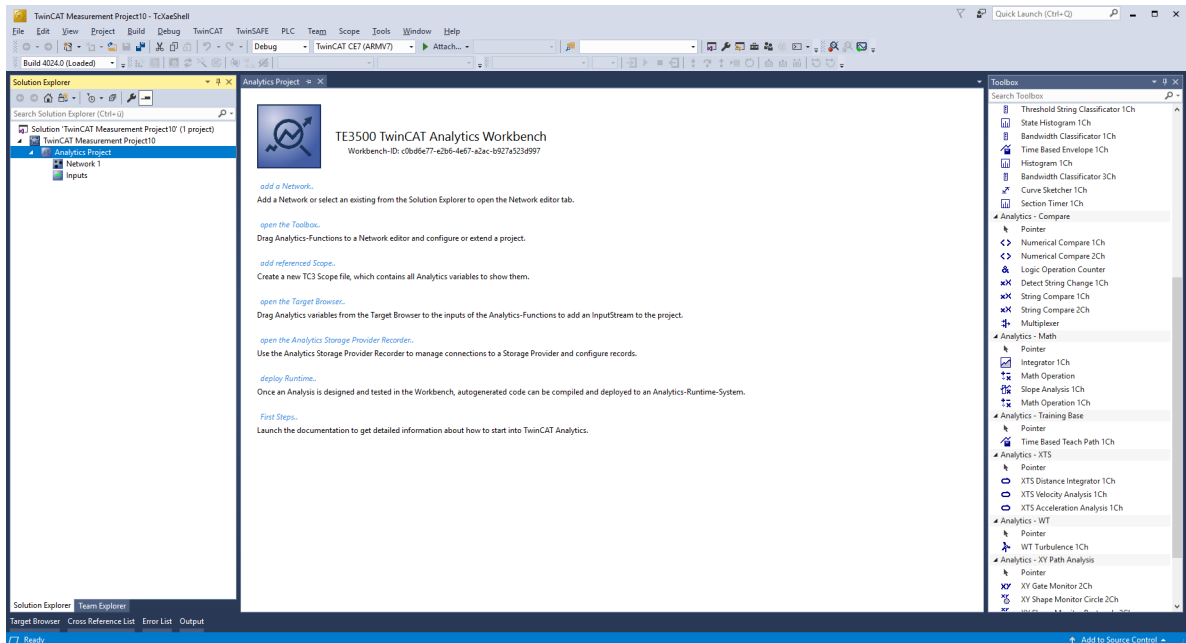
✓ Öffnen Sie Ihre TwinCAT Engineering-Umgebung, um die Analyse der Daten zu starten.

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

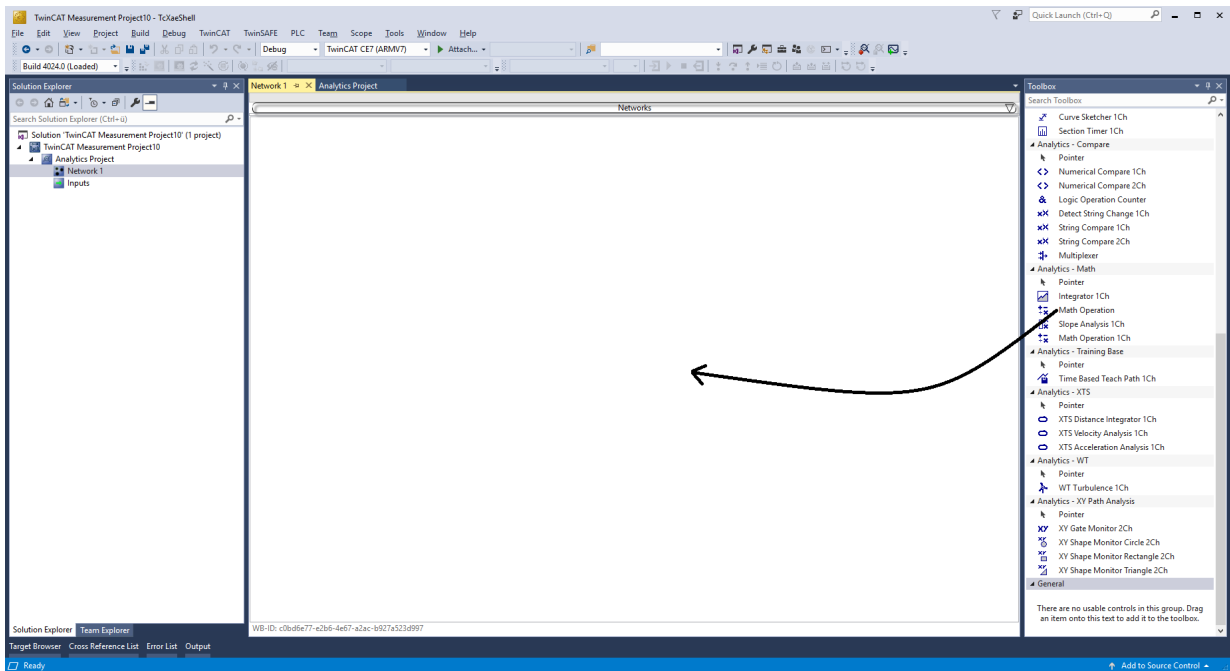
2. Wählen Sie aus **TwinCAT Measurement** die **Analytics-Projektvorlage** aus.



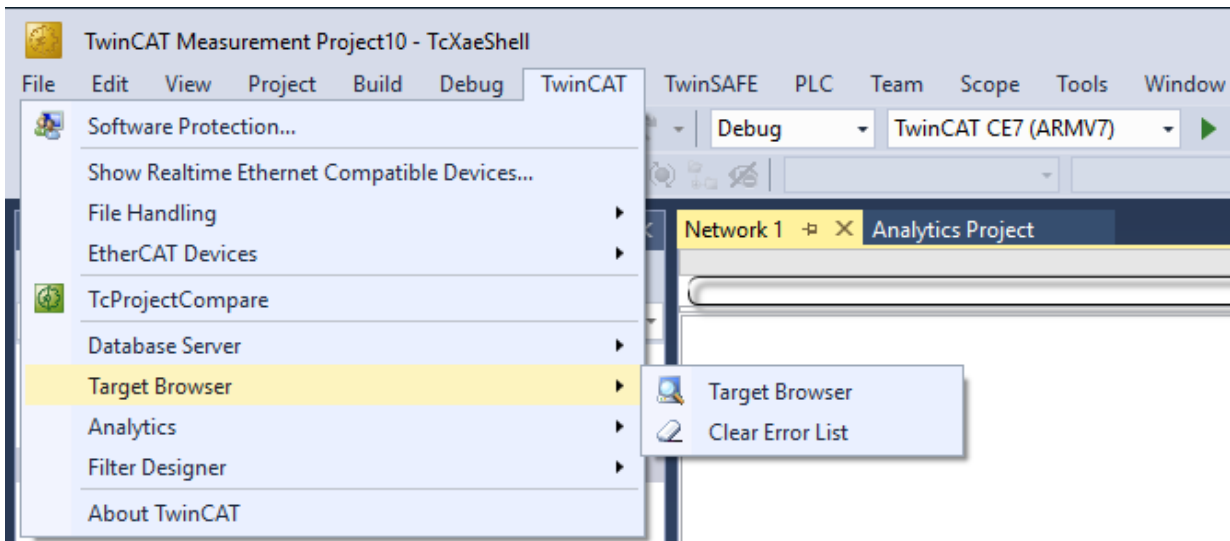
⇒ Das neue Projekt wird im Projektmappen-Explorer angezeigt. Nach einem Klick auf das Baumknotenenelement **Analytics Project** öffnet sich ein Startfenster, in dem Sie Ihre erste Aktion auswählen können. Von hier aus können Sie ein Netzwerk hinzufügen, die **Toolbox** öffnen, den **Target Browser**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



- 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 hinzuzufügen, oder fügen Sie mit Hilfe der Startseite ein Netzwerk hinzu. Das erste Netzwerk wird immer standardmäßig generiert.

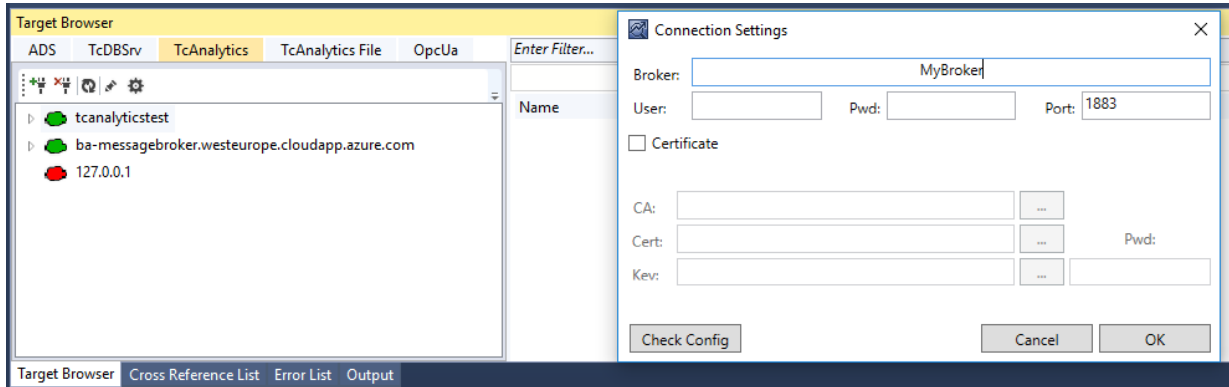


- Wenn Sie auf das Netzwerk klicken, wird ein Editor geöffnet. Nun können Sie den gewünschten Algorithmus per Drag-and-Drop in die Editor-Oberfläche ziehen.
- Nach der Auswahl des Algorithmus müssen Sie Eingangsvariablen mit den Modulen (Algorithmus) verbinden. Öffnen Sie dazu den **Target Browser**.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

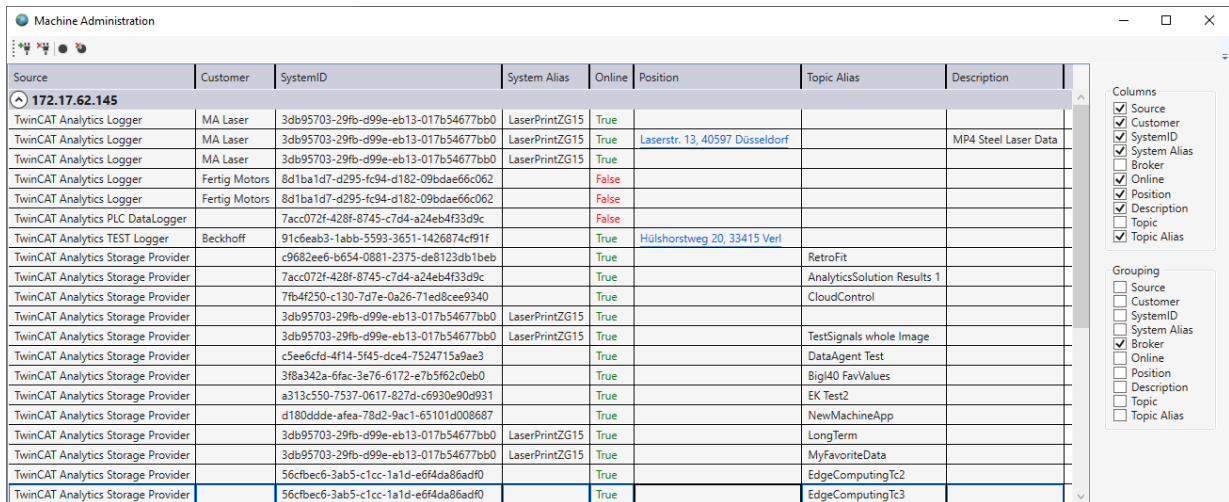


- Wählen Sie nun den Karteireiter **TcAnalytics** oder **TcAnalyticsFile** im Target Browser aus. Mit dem Karteireiter **TcAnalytics** (MQTT) geht es weiter.

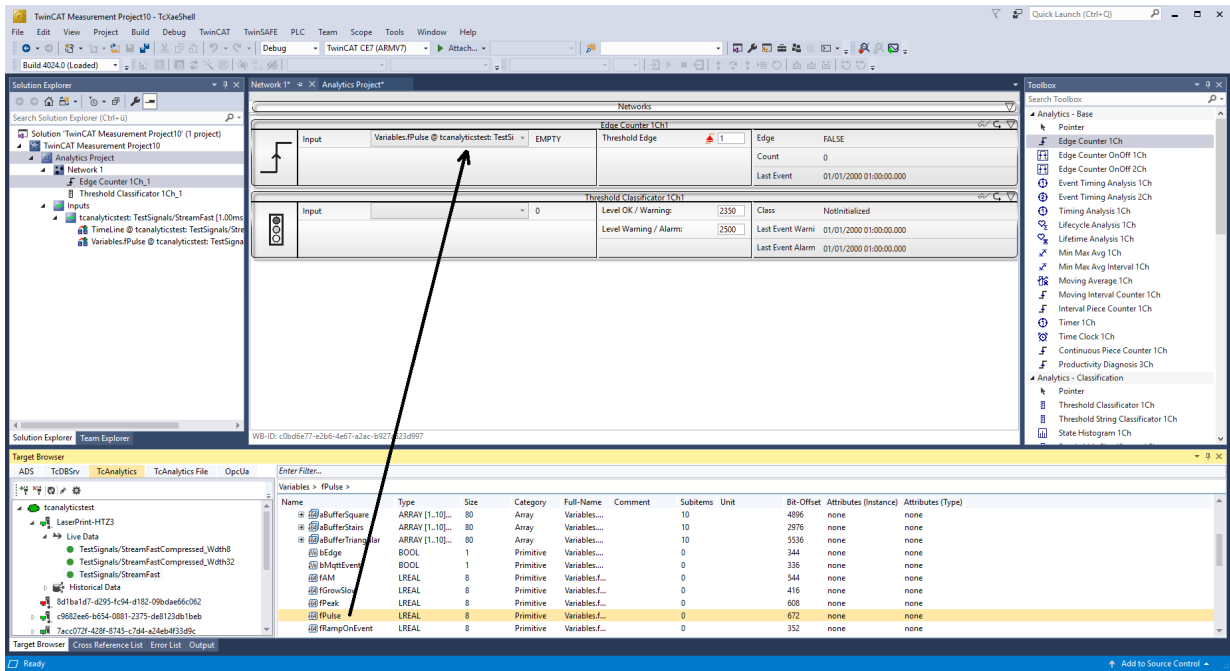
- Klicken Sie auf das grün markierte Icon in der Symbolleiste dieser Analytics-Erweiterung. Es öffnet sich ein Fenster, in dem Sie die Konnektivitätsdaten Ihres Message-Brokers angeben können.



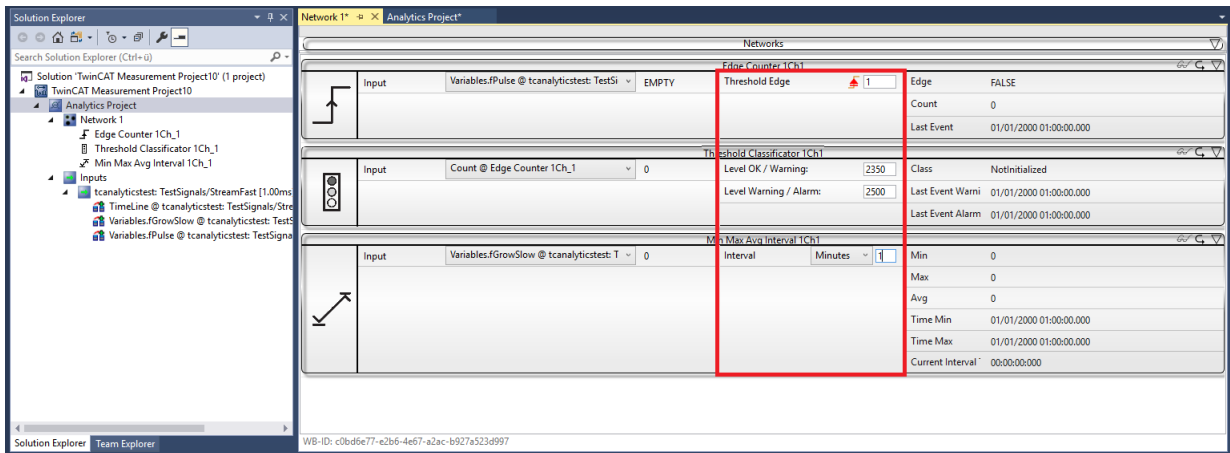
- Wählen Sie Ihren MQTT-Analytics-Client aus (TwinCAT Analytics Logger, TwinCAT IoT Data Agent oder Beckhoff EK9160). Für jede Steuerung gibt es eine eindeutige ID. Diese ID wird im Target Browser angezeigt.
- Über einen Klick auf das **Zahnrad-Icon**, gelangen Sie zur Seite Machine Administration. Hier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



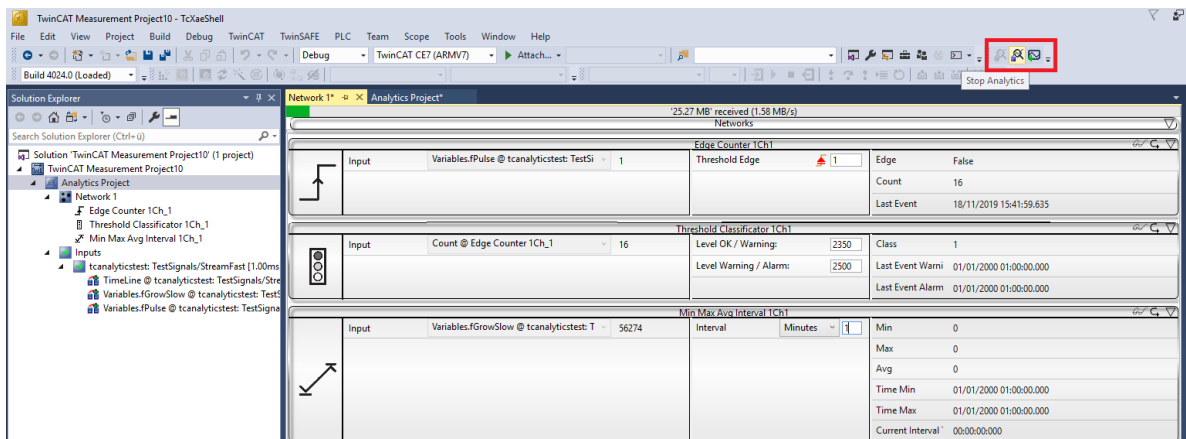
10. Im nächsten Schritt können Sie zwischen Live-Daten und historischen Daten für jeden MQTT Analytics-Client wählen. Die historischen Daten werden in diesem Fall von dem TwinCAT Analytics Storage Provider bereitgestellt.



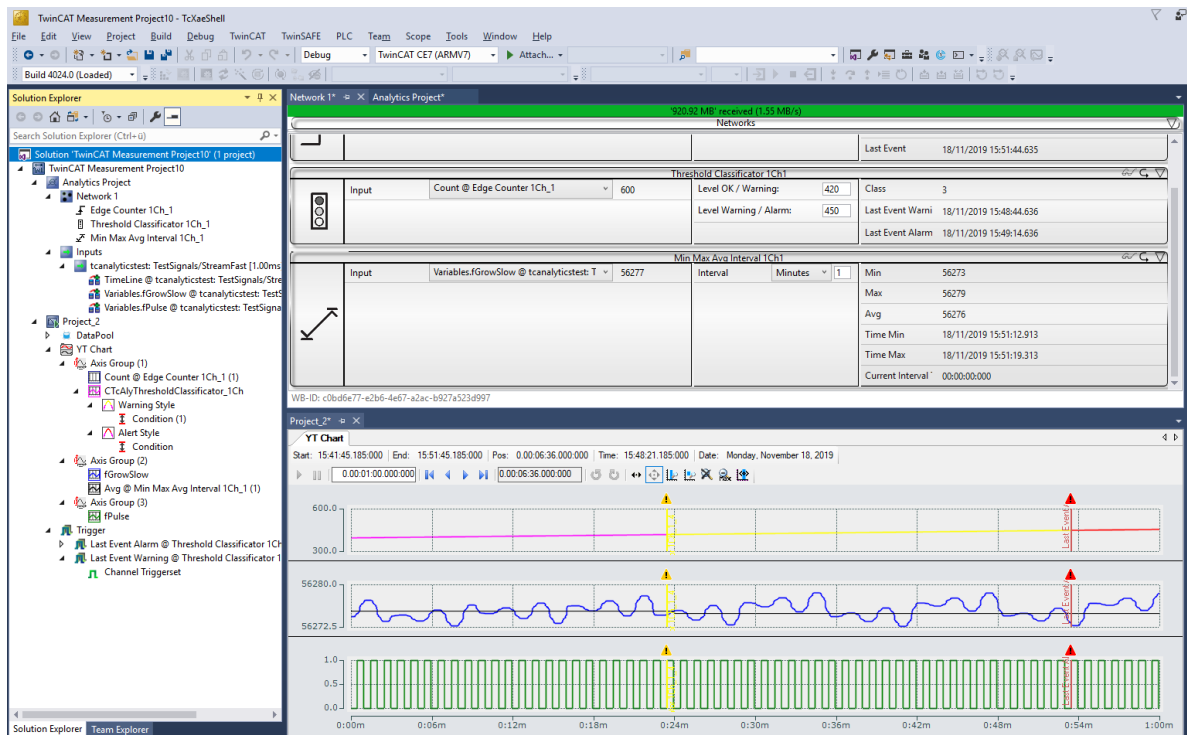
11. Sie können die Variablen per Drag-and-Drop in die Eingänge des spezifischen Algorithmus ziehen. Bei den meisten Algorithmen können Bedingungen wie Schwellen, Zeitintervalle, logische Operatoren usw. festgelegt werden. Diese Einstellungen werden in der Mitte jedes Moduls vorgenommen.



⇒ 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.

5 Technische Einführung

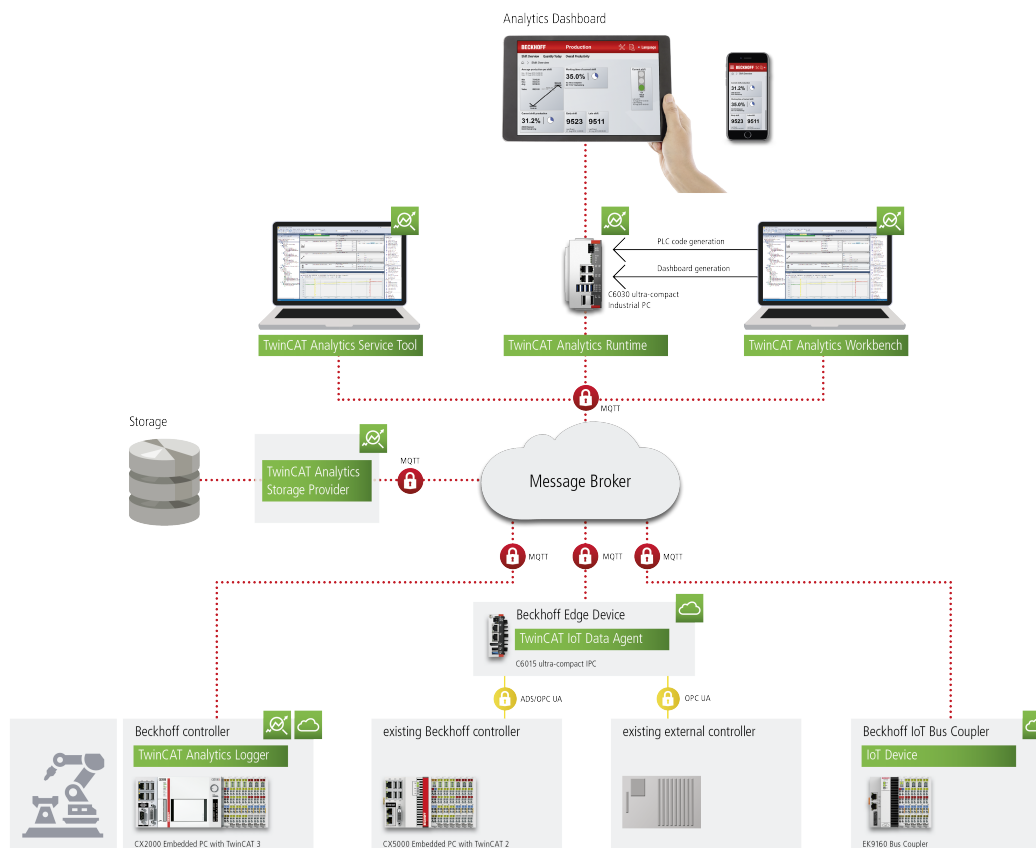
5.1 Grundkonzept

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.

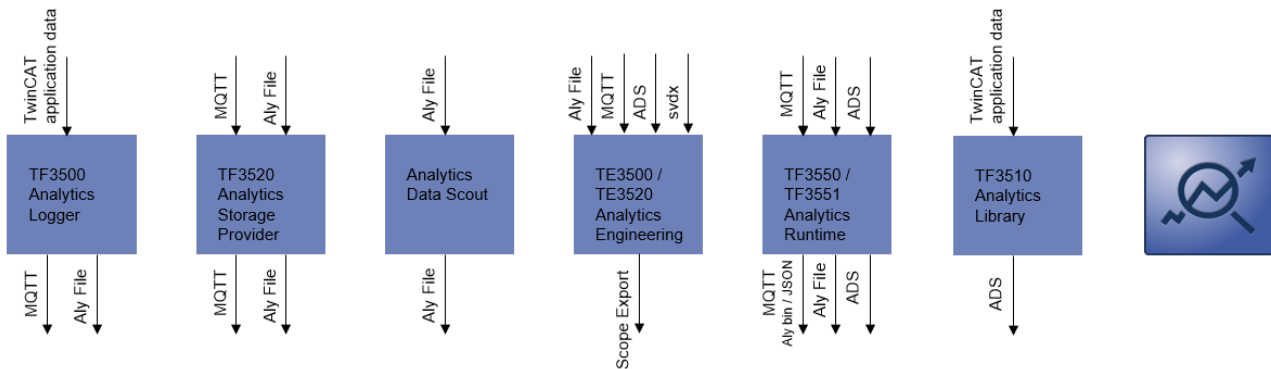
Produktnummer	Produktname
TE3500	Analytics Workbench
TE3520	Analytics Service Tool
TF3500	Analytics Logger
TF3510	Analytics Library
TF3520	Analytics Storage Provider
TF3550	Analytics Runtime – inclusive HMI Server und Client Pack
TF3551	Analytics Runtime Base – ohne HMI
TF3560	Analytics Controller Pack 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

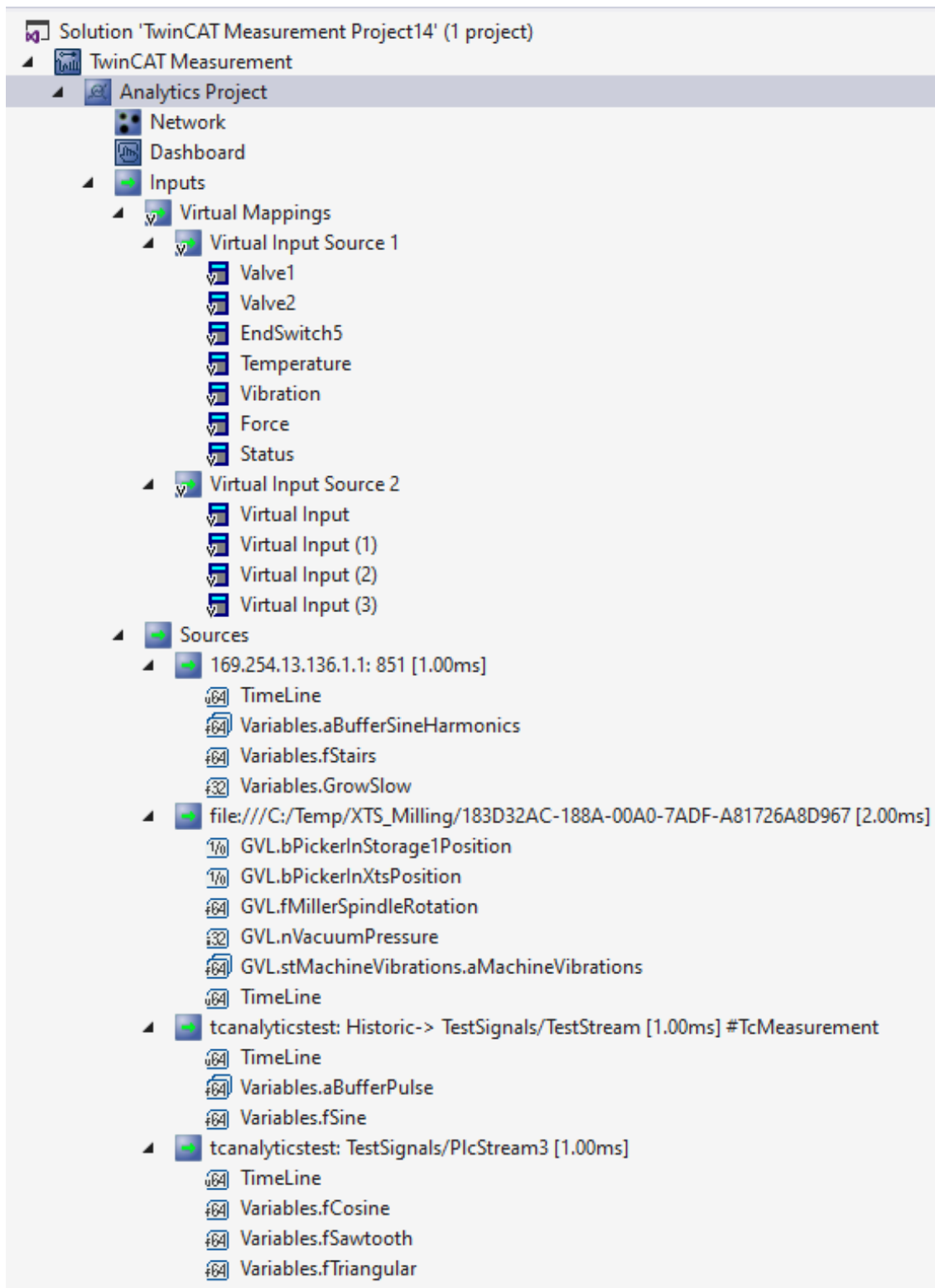
6.1 Datenquellen

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 Sources können Konfigurationen für verschiedenen Sources 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.

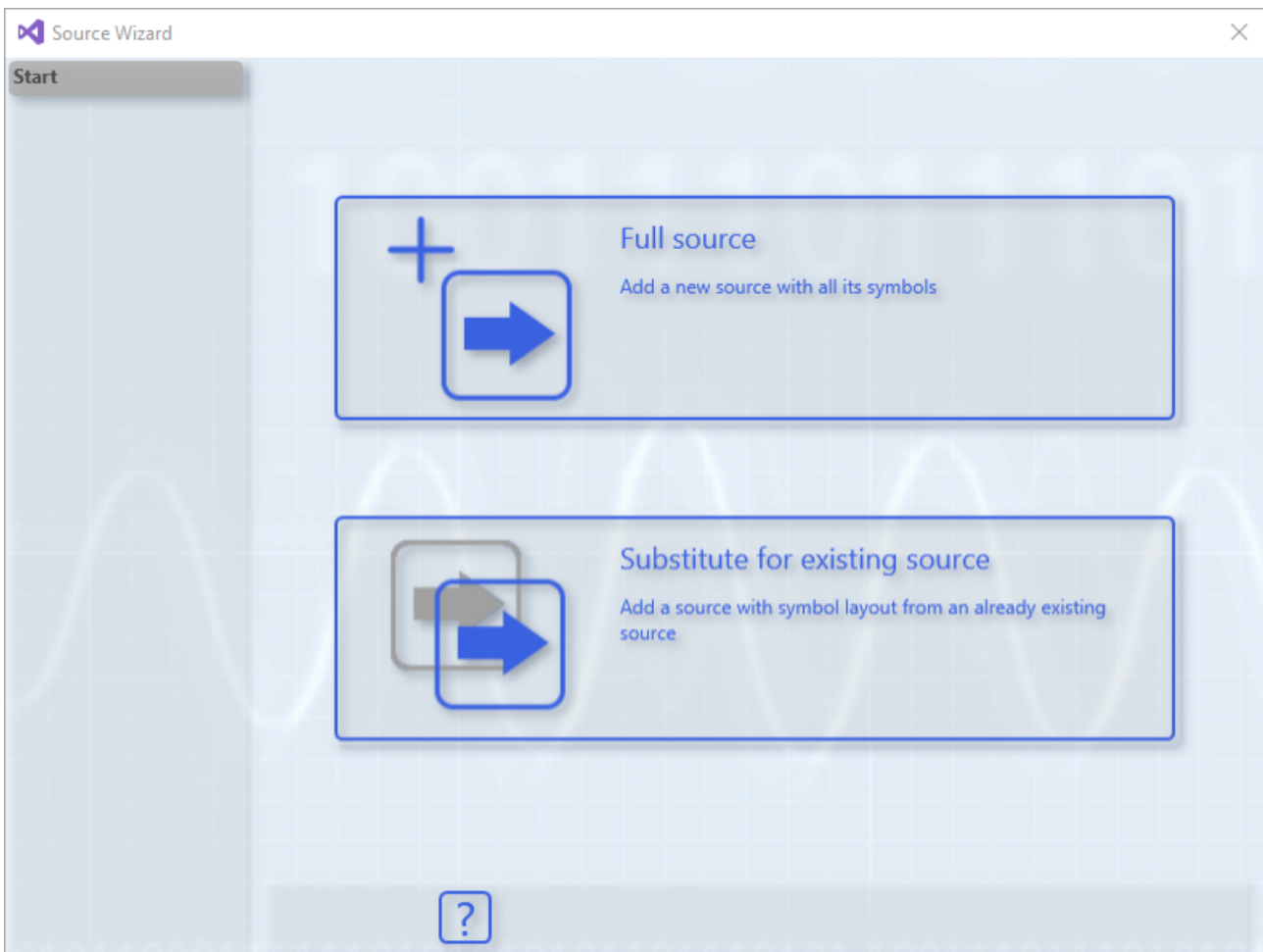


Um den Source Wizard nutzen zu können muss der TwinCAT Target Browser installiert sein.

Seiten:

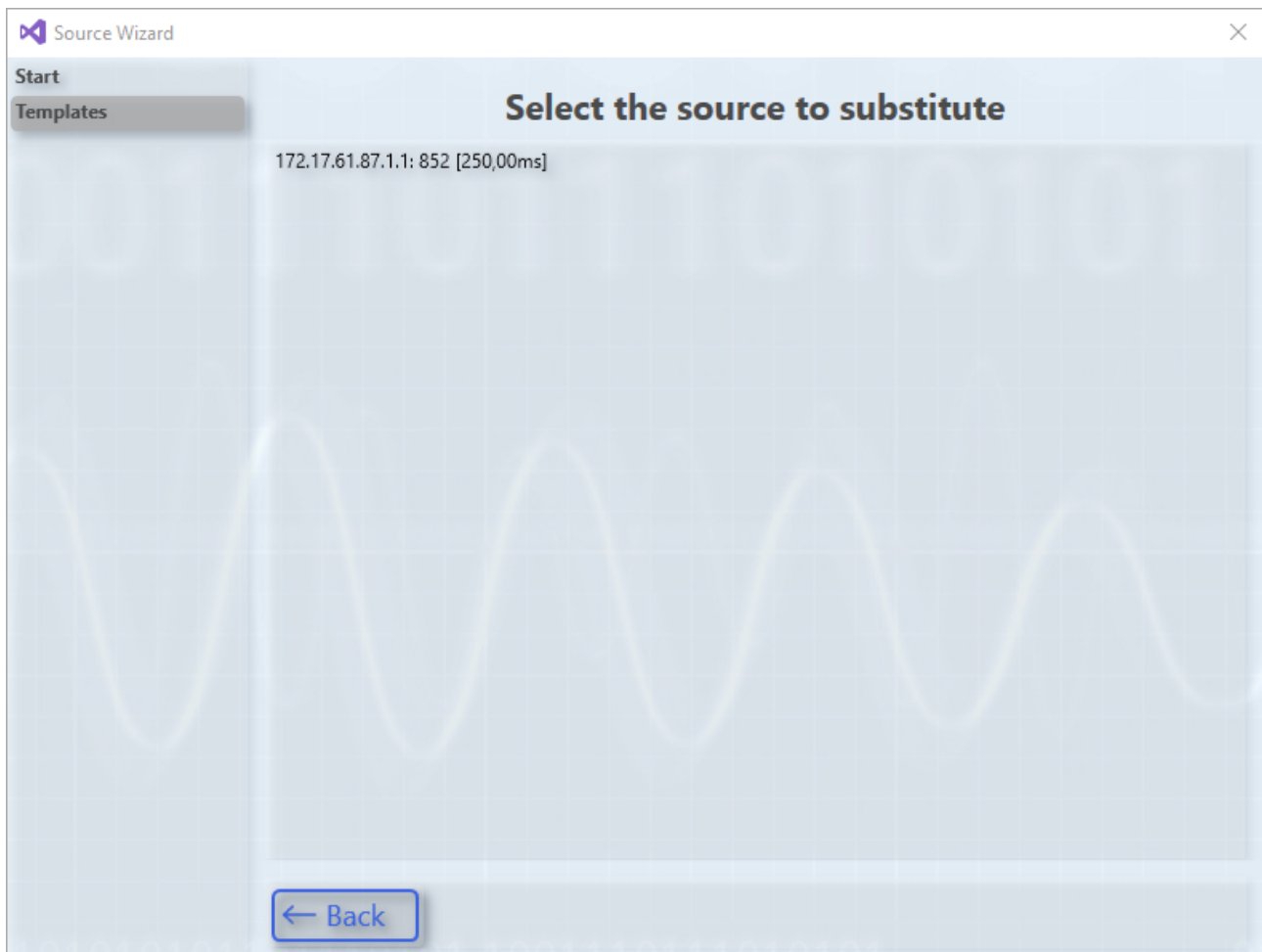
Der Source Wizard ist in mehrere Seiten aufgeteilt. Auf der linken Seite werden die bereits gezeigten Seiten aufgelistet. Mithilfe dieser Liste, sowie der Buttons **Back** und **Next** können bereits geöffnete Seiten erneut gezeigt werden.

Start



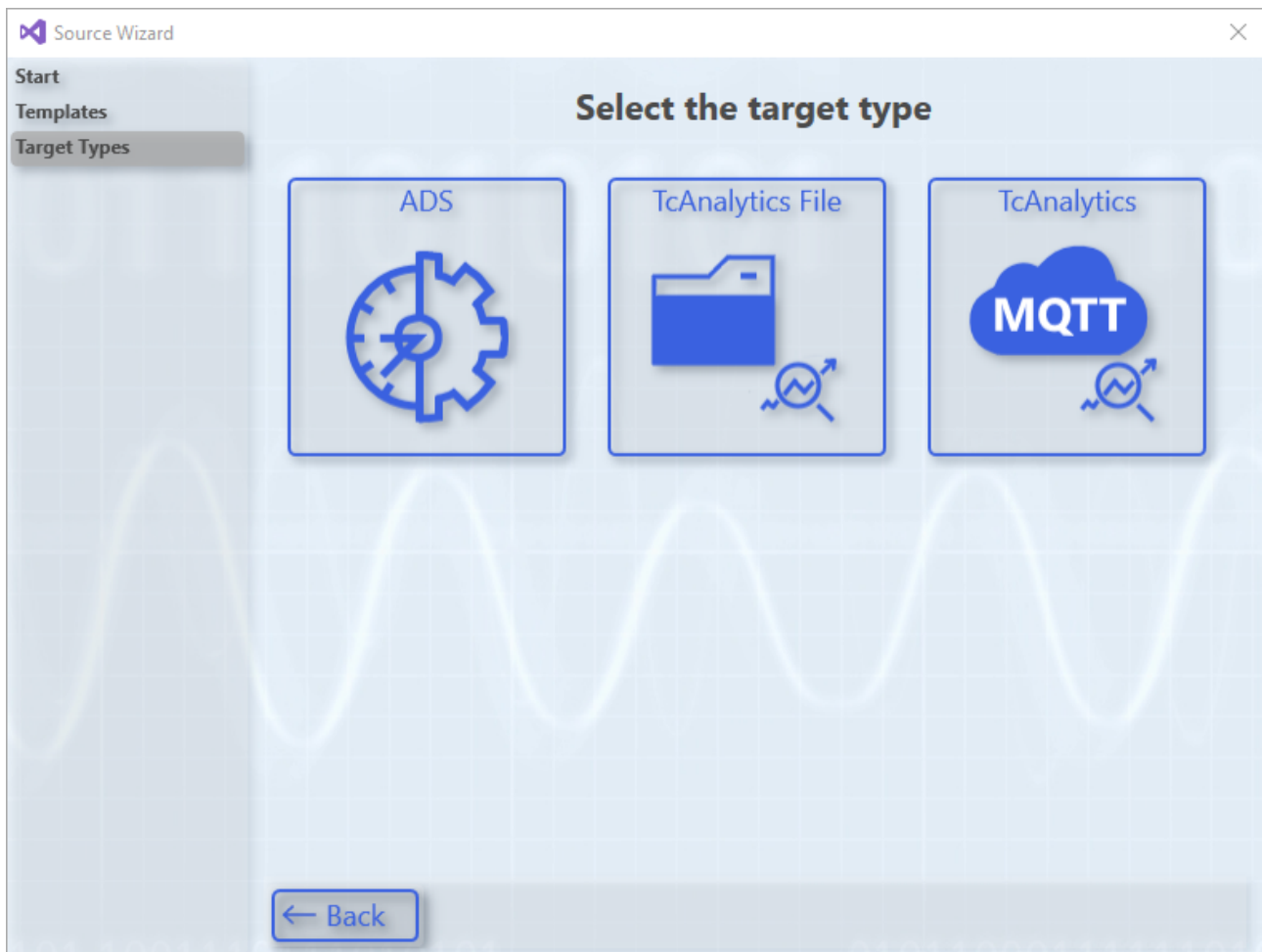
Auf dieser Seite kann ausgewählt werden ob eine gesamten Symbole einer Quelle hinzugefügt oder nur Symbole, die anhand einer schon vorhandenen Quelle gesucht werden, hinzugefügt werden sollen. Die Option zum Ersetzen einer Quelle kann nur gewählt werden, wenn bereits eine Quelle im Projekt vorhanden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

Vorlagen



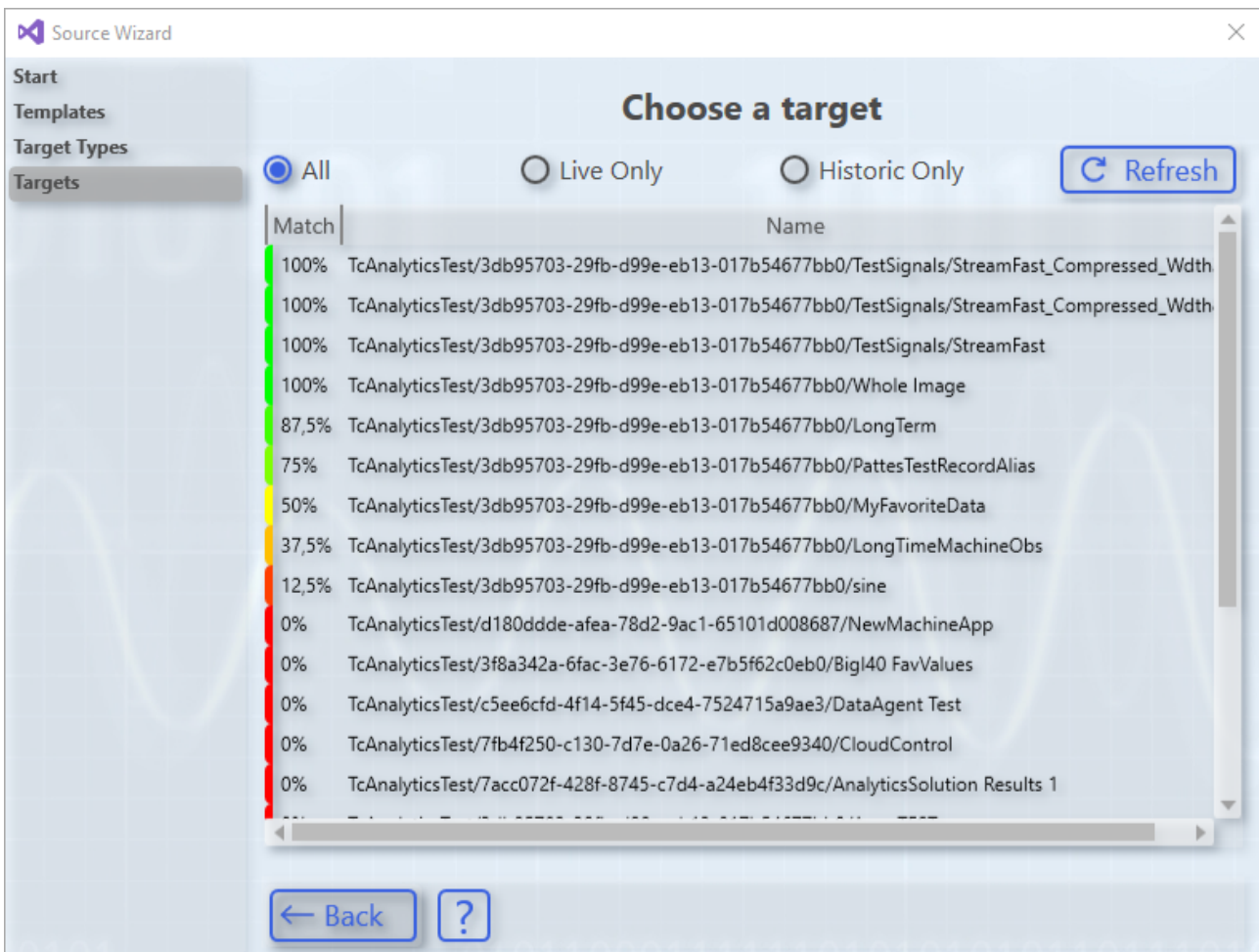
Diese Seite wird nur angezeigt, wenn auf der Start Seite die Option Ersetzen ausgewählt worden ist. Hier werden bereits vorhandenen Quellen des Projektes angezeigt. Zum Fortfahren muss eine Quelle ausgewählt werden die dann als Vorlage für die neue dient.

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 kompatibel 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.

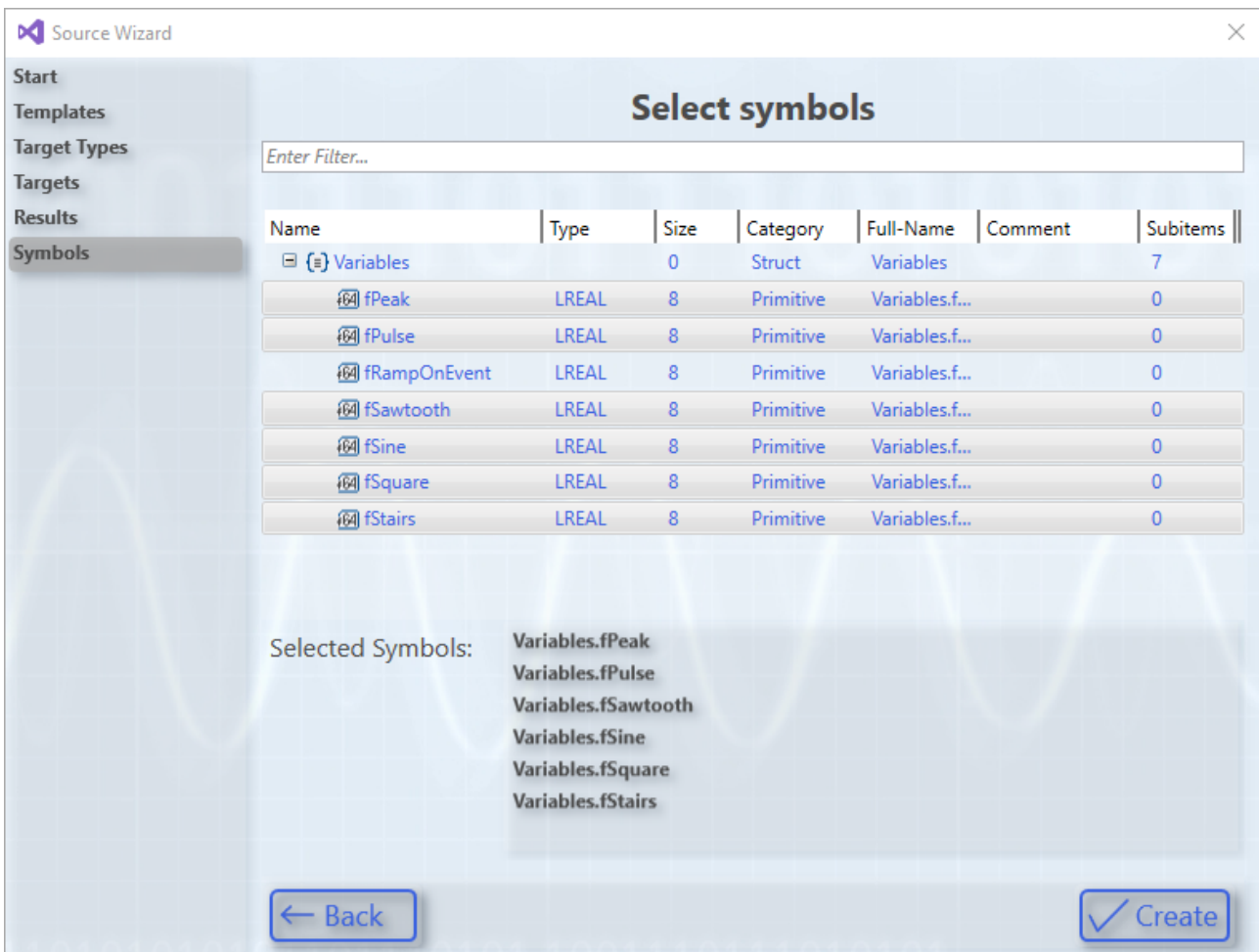
Ergebnisse



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.

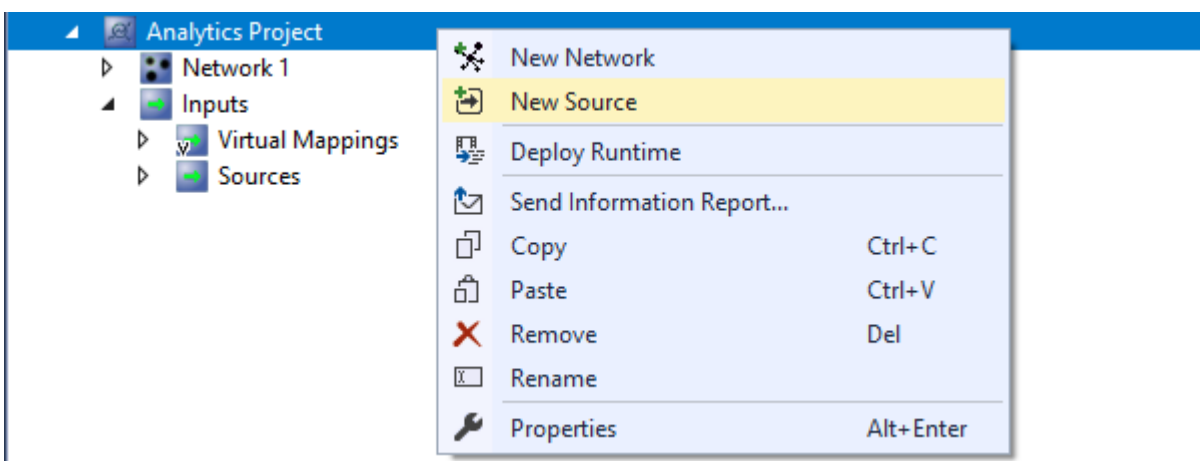
Symbole



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 derzeit 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.

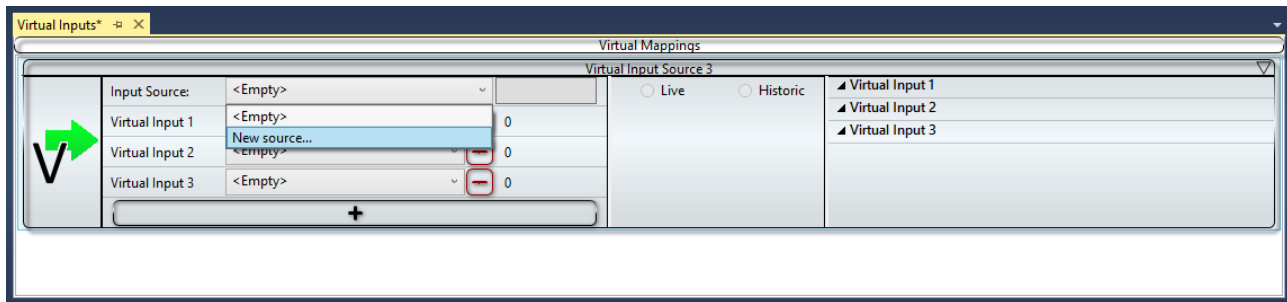
Aufruf:

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 derzeit 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 hinzugefügt 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.

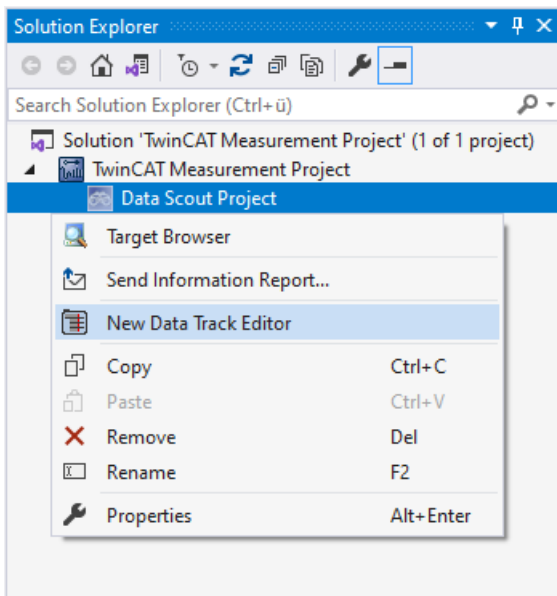
Das Data Scout Projekt kann in der Engineering Umgebung unterhalb des Measurement Projektes angelegt werden und wird auf den folgenden Seiten genauer erläutert.

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

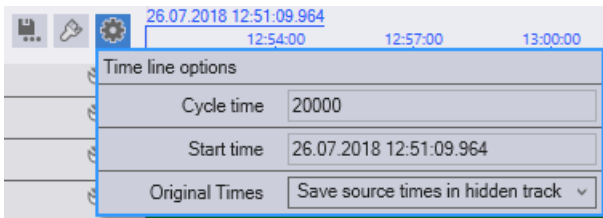


1. [Tracks \[► 46\]](#) stellen die Symbolik und Daten im Data Track Editor dar.
2. Das [Chart \[► 53\]](#) wird genutzt um die in den Tracks liegenden Daten als Graphen zu visualisieren.
3. Die [Toolbar \[► 54\]](#) enthält Funktionen, um z.B. den angezeigten Bereich zu ändern oder zum Ausscheiden von Bereichen.

Zum [Exportieren \[► 55\]](#) muss der  Button geklickt werden.

Timeline

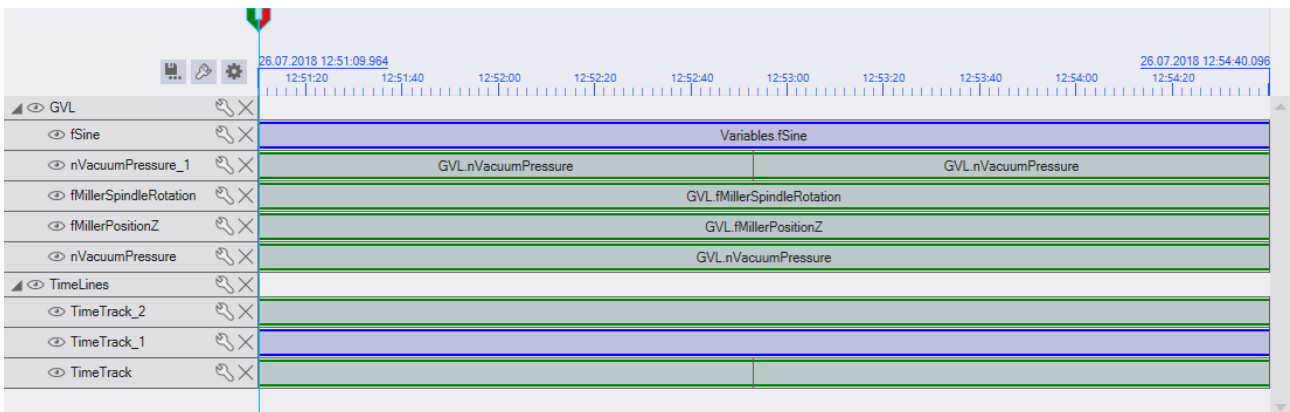
Da im Track Editor verschiedene Quellen hinzugefügt werden können, werden neue Zeiten vergeben. Der Startzeitpunkt und die Zykluszeit können in den Timeline Options eingestellt werden. Diese Zeiten werden im Export und im Chart verwendet. Die Start- und Zykluszeit der ersten hinzugefügten Quelle wird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

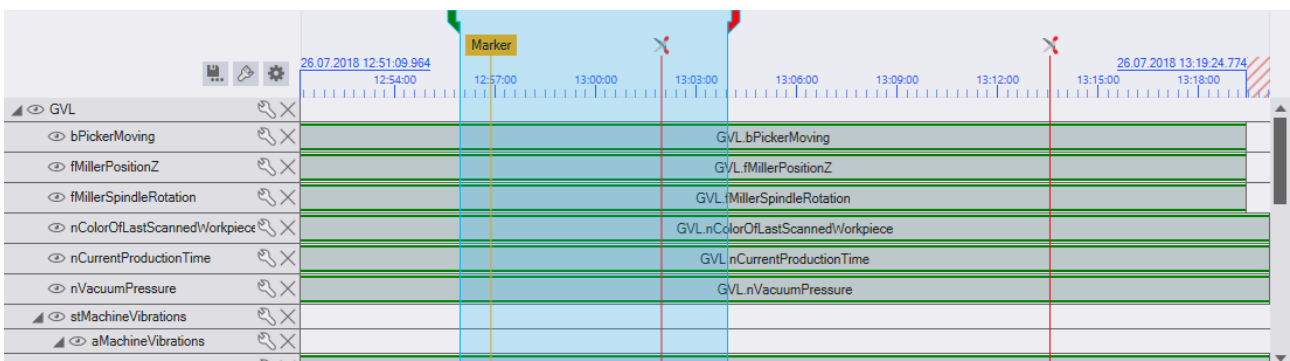
In der Beispiel Darstellung wurden die Time Tracks sichtbar generiert. Es wurden automatisch drei erstellt da in den darüberliegenden Tracks drei Kombinationen gefunden worden sind. Zum einen die Track mit einem blauen Datenelement, zum anderen eine Track mit zwei grünen Datenelementen und noch drei 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.
	<p>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.</p> <p>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.</p> <p>Die Selektion kann nicht kleiner als 10 Werte sein.</p>
	<p>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.</p>
	Die Timeline hat am linken Ende die Startzeit markiert. In gleichmäßigen Abständen folgen dann Markierungen, welche die dort anliegende Zeit anzeigen.
	<p>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.</p> <p>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.</p>
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

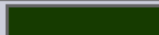
Container Tracks sind ähnlich wie Datentracks aufgebaut. Allerdings können keine Datenelemente in Container Tracks liegen, dafür können sie aber Tracks enthalten. Enthaltene Tracks werden unterhalb der

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:

Data Track Options	
Name	nVacuumPressure
Data Type	Int
Line Color	

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:

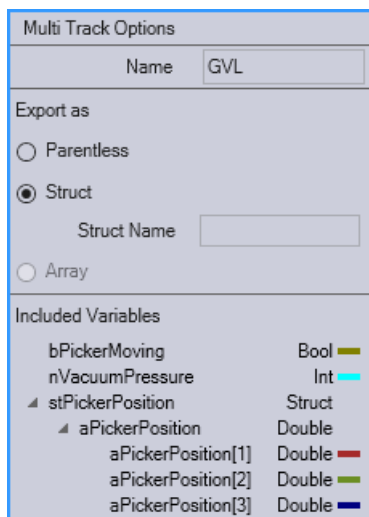
Container Track Options	
Name	aPickerPosition
Container type	
<input type="radio"/>	Struct
<input checked="" type="radio"/>	Array
Array Base Type	Double
Oversampling	<input type="checkbox"/>

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	<p>Es kann, ähnlich zum Containertyp der Container Track, eingestellt werden, wie die Multi Track im Export behandelt werden soll.</p> <p>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.</p> <p>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.</p> <p>Bei Parentless wird die Multi Track im Export nicht berücksichtigt und es ist so als würden die enthaltenen Variablen auf der Ebene der Multi Track liegen.</p>
Enthaltene Variablen	Hier wird die, in der Multi Track vorliegende, Symbolik angezeigt. Alle Datenelemente der Multi Track müssen die gleiche Symbolik haben.

Datenelemente

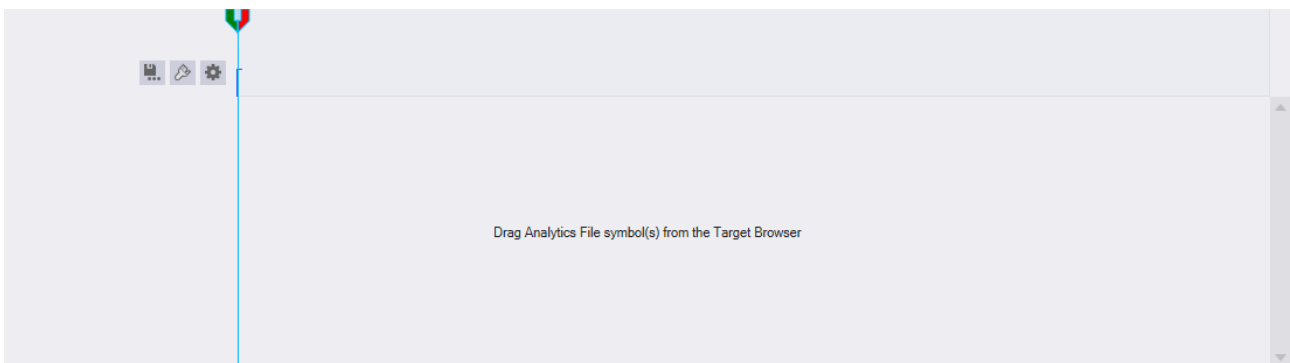
In Tracks können Datenelemente liegen. Diese enthalten eine oder mehr Variablen aus einer Quelle, sowie einen Zeitbereich aus der Quelle. Mithilfe des Zeitbereichs und der Zykluszeit der Quelle wird errechnet wie viele Samples im Dataelement liegen. Die Länge eines Datenelements wird festgelegt indem 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 Datenelement 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 Datenelement wird der Originalname des Symbols aus der Quelle angezeigt. Wenn mehrere Symbole vorliegen, wird nur die Anzahl der Symbole angezeigt.

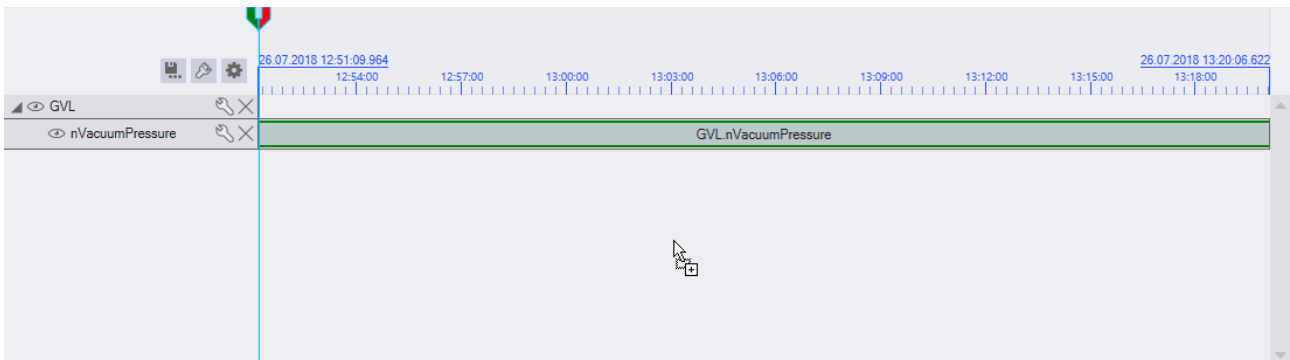
Per Rechtsklick auf ein Datenelement kann das Kontext Menü geöffnet werden. Darüber kann das Datenelement auch komplett entfernt werden, ohne die Funktionen aus der Toolbar zu nutzen.

Hinzufügen per Drag and Drop

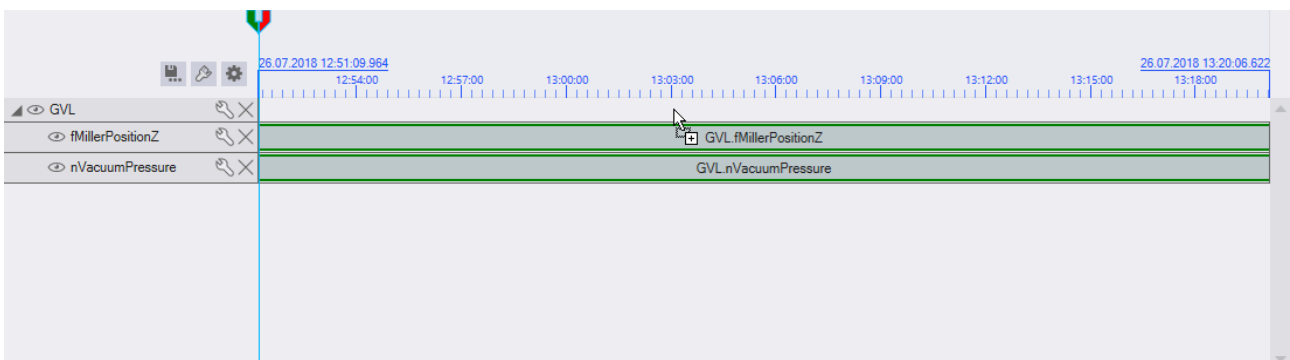
Die Tracks eines Data Track Editors werden in einen gesonderten Bereich angezeigt. Dieser ist zu Beginn leer und kann mithilfe des Target Browsers gefüllt werden.



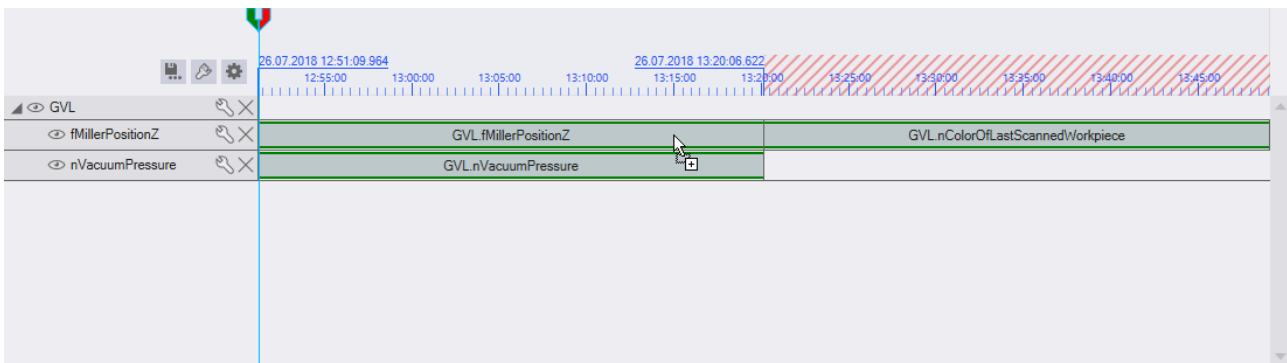
Wenn aus dem Target Browser ein oder mehr Symbole gewählt werden und in den leeren Bereich des Track Editors gezogen werden, dann wird eine neue Container Track erstellt. Zu der neuen Container Track werden alle ausgewählten Symbole als Tracks hinzugefügt.



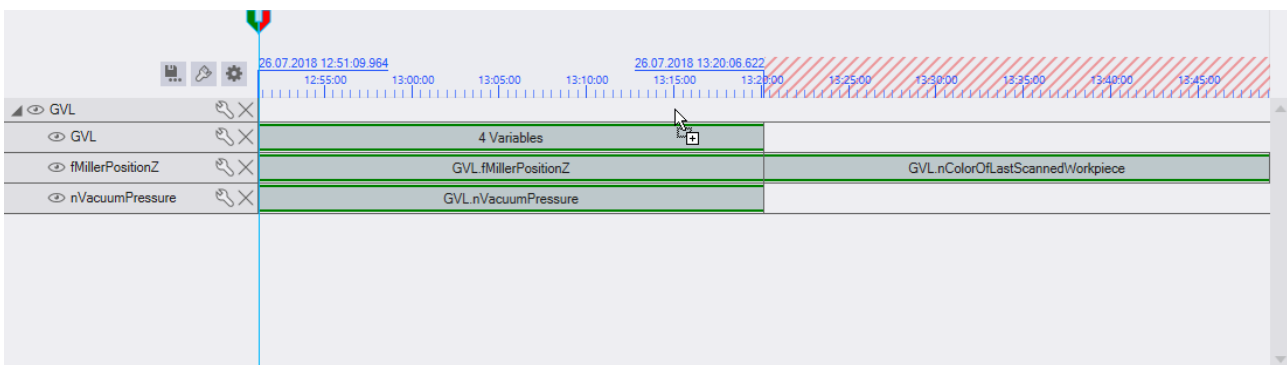
Wird auf den Daten Bereich einer Container Track, per Drag Drop, ein Symbol gezogen, wird eine neue Track unterhalb dieses Containers erstellt.



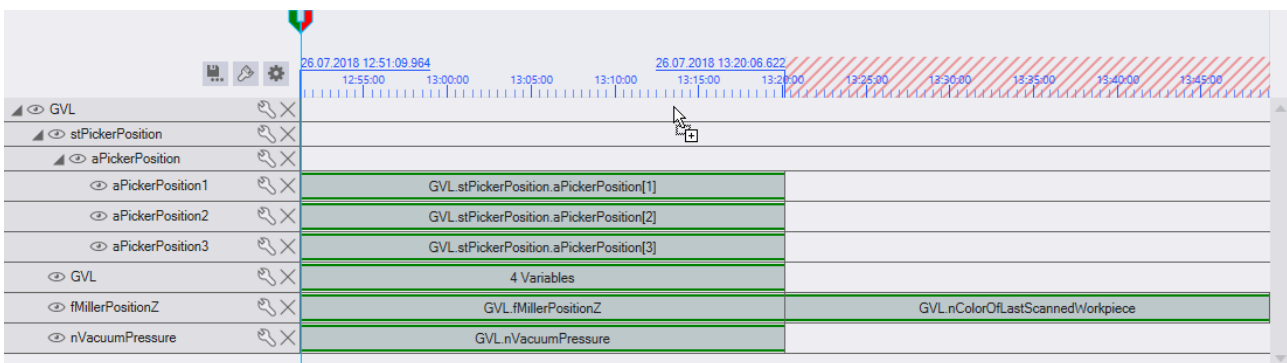
Wenn auf eine Datentrack gezogen wird, kann zu dieser Track ein neues Datenelement hinzugefügt werden. Das ist allerdings nur möglich, wenn das hinzuzufügende Symbol mit der Track kompatibel ist. Es kann keine Struktur auf eine Track mit primitiven Datentypen gezogen werden.



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

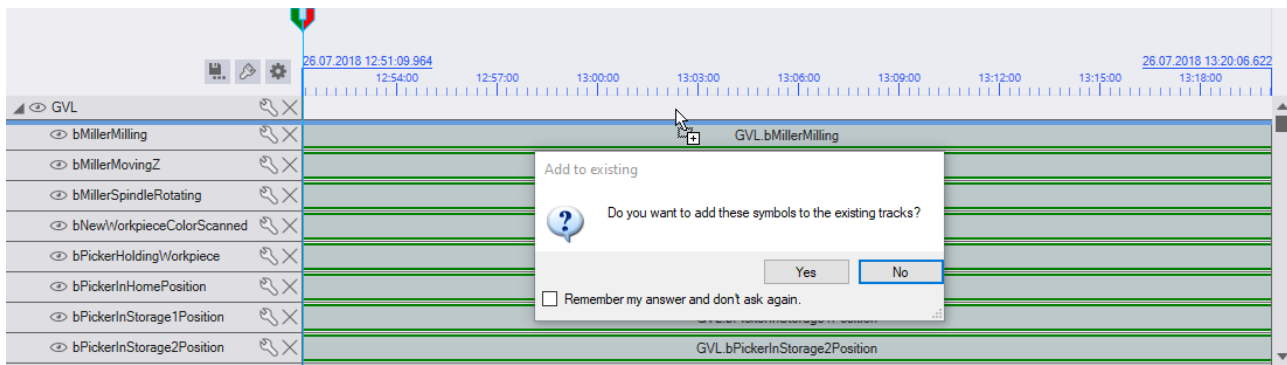


Wenn Strukturen oder Arrays hinzugefügt werden sollen, kann dies wie bei Symbolen mit primitivem Datentyp passieren. Allerdings werden für die Oberэлеnte von Strukturen und Arrays Container Tracks erstellt.

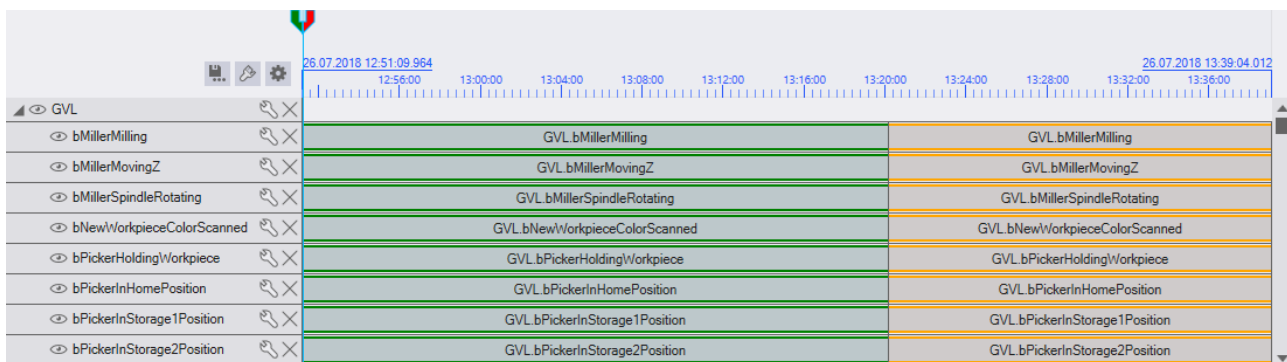


Automatisches Zusammenfügen

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

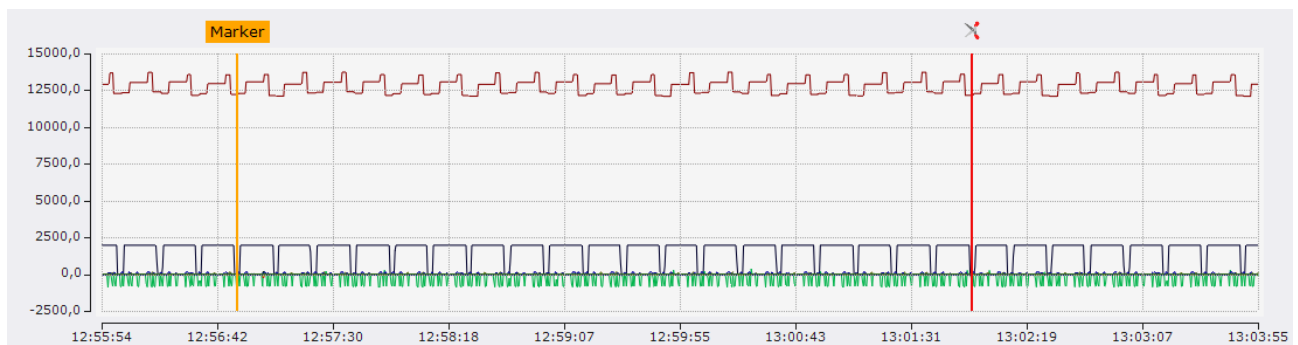
Name	Type	Size	Category
bPickerInXtsPosition	BOOL	1	Primitive
bPickerMoving	BOOL	1	Primitive
fMillerPositionZ	LREAL	8	Primitive
fMillerSpindleRotatic	LREAL	8	Primitive
nColorOfLastScanne	INT	2	Primitive
nCurrentProduction	ULINT	8	Primitive
nVacuumPressure	DINT	4	Primitive
stMachineVibration	STRUCT	...	Struct
stPickerPosition	STRUCT	...	Struct
stTableInMillingPos	STRUCT	...	Struct
aTableInMillingP	ARRAY [1..5]...	5	Array
aTableInMillir	BOOL	1	Primitive

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 Ausschneid-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.

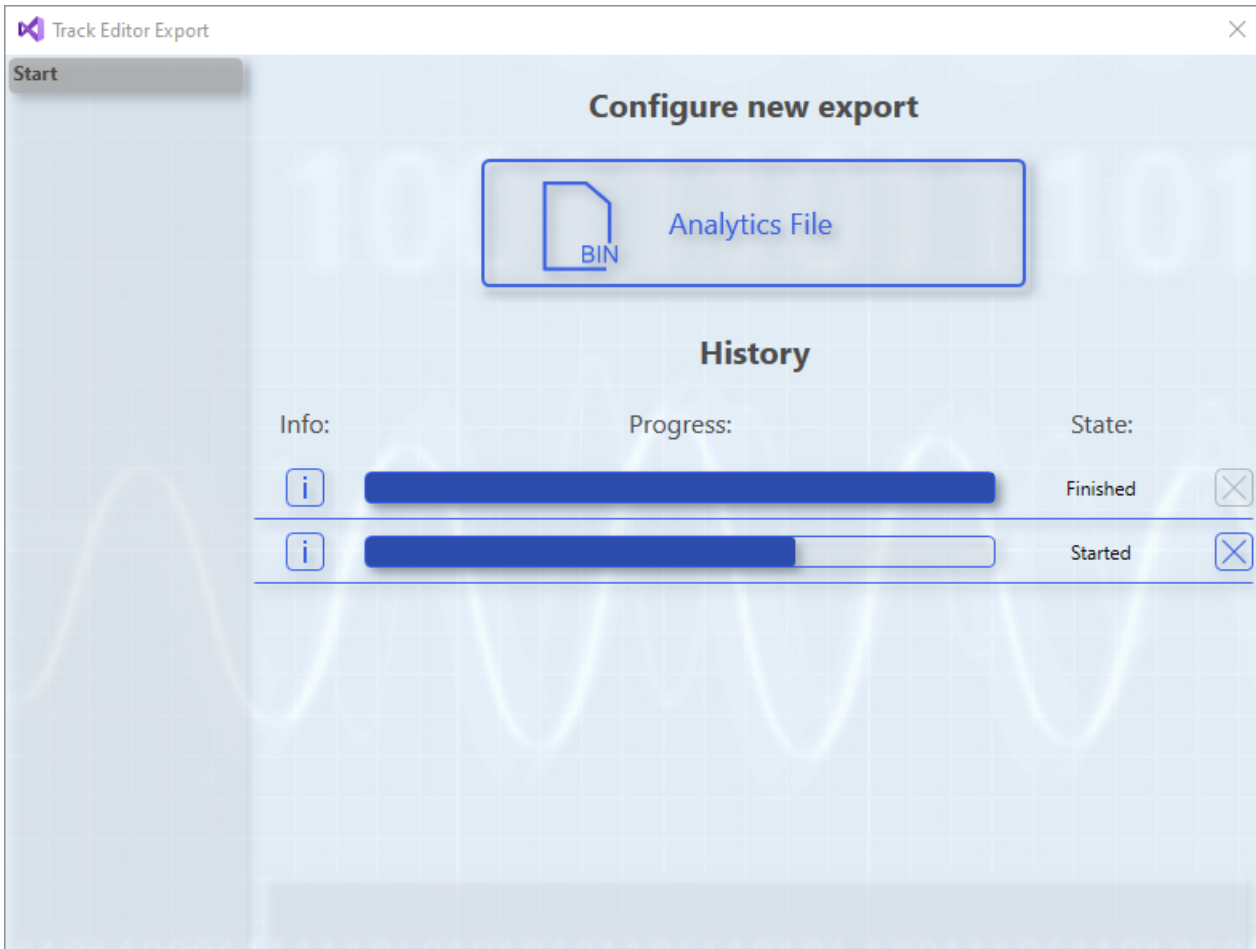
Diese Funktionen können dazu genutzt werden, um die Daten der Tracks zu bearbeiten.

	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.

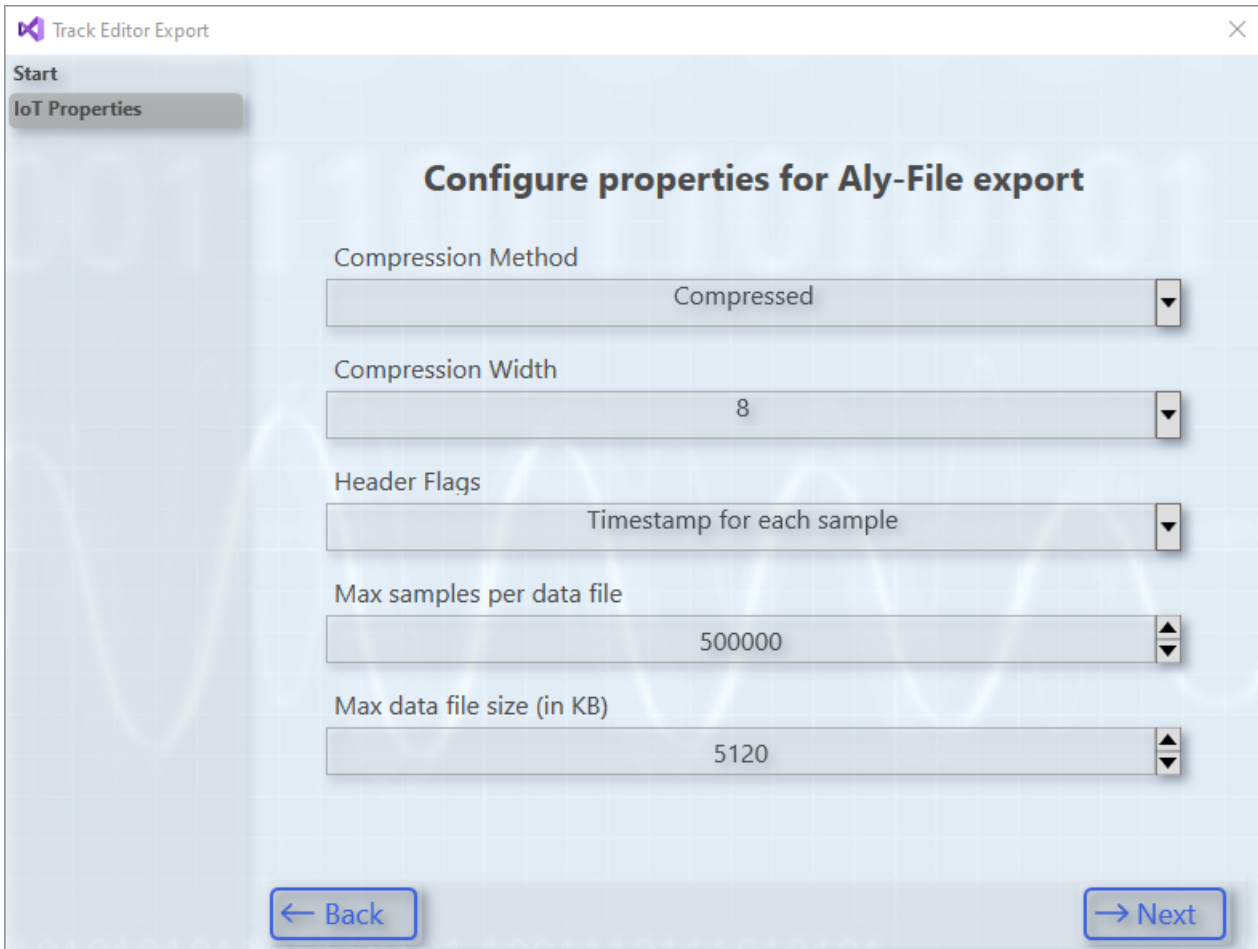
Startseite



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 anzeigt 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 abubrechen.

Wenn auf einen der Buttons geklickt wird, mit denen ein neuer Export gestartet werden kann, wird eine oder mehrere 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 angegeben 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

Track Editor Export

Start

IoT Properties

Export

Verify and start export

General:

Export start time:
26.07.2018 12:51:09.964

Export end time:
26.07.2018 13:20:06.622

Format Options:

Include original times: True
 Compression: Compressed, Width 8
 Header flags: Timestamp per sample
 Max data file size: 5120000
 Max sample count per file: 500000

Enter filepath...

Data Scout Export

Use symbolguid as name

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.

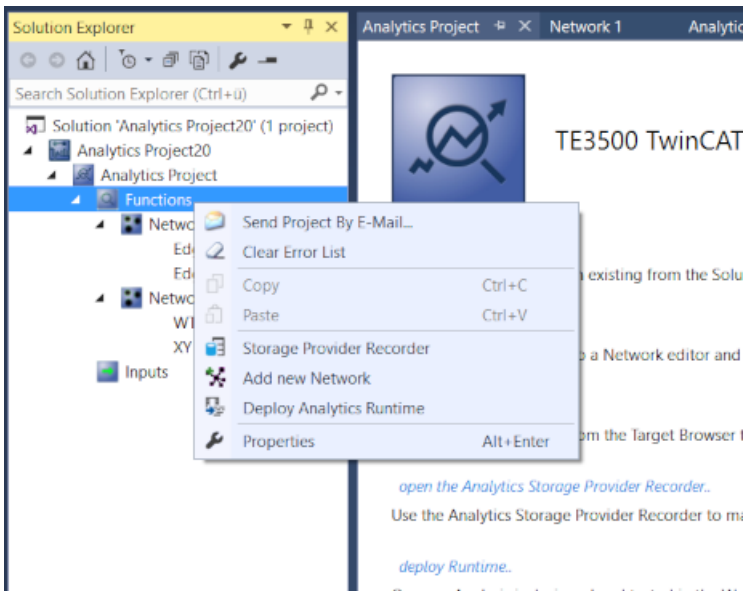
Wenn auf den Create Button geklickt wird, startet der Export im Hintergrund. Dann schließt sich diese Seite und der Wizard springt zurück auf die Startseite. Hier steht dann auch der neue Export in der Historie und der Fortschritt kann verfolgt werden.

6.3 Editor Basics

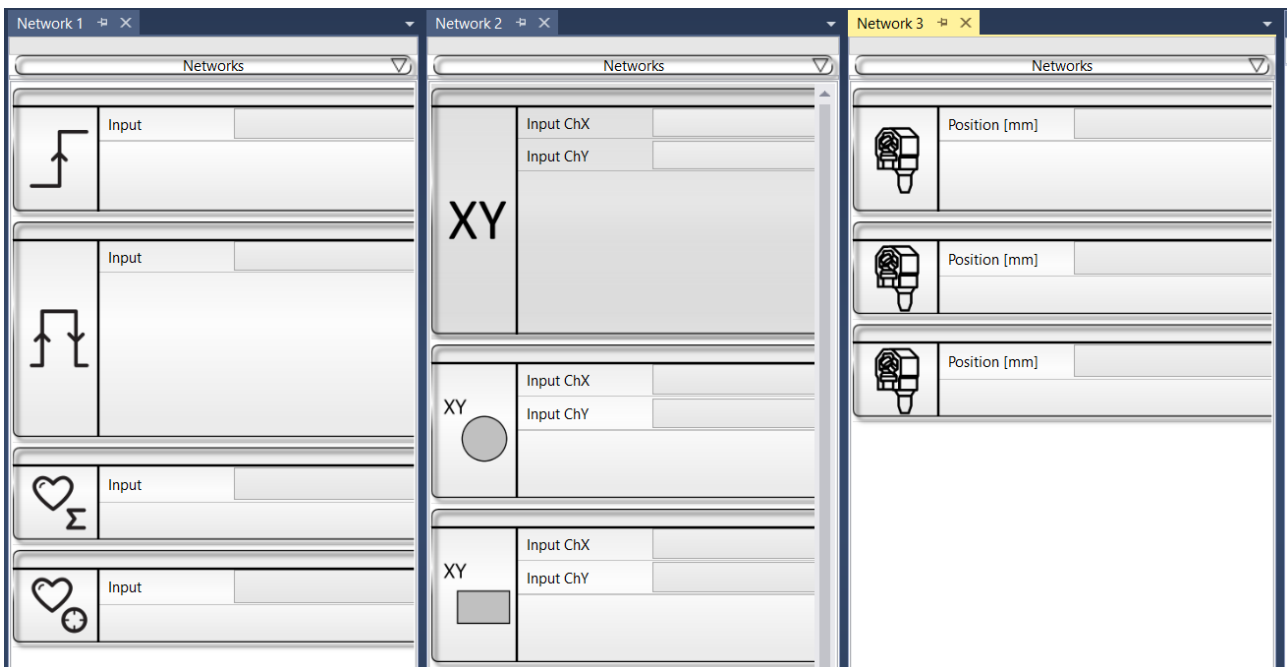
6.3.1 Netzwerke

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.

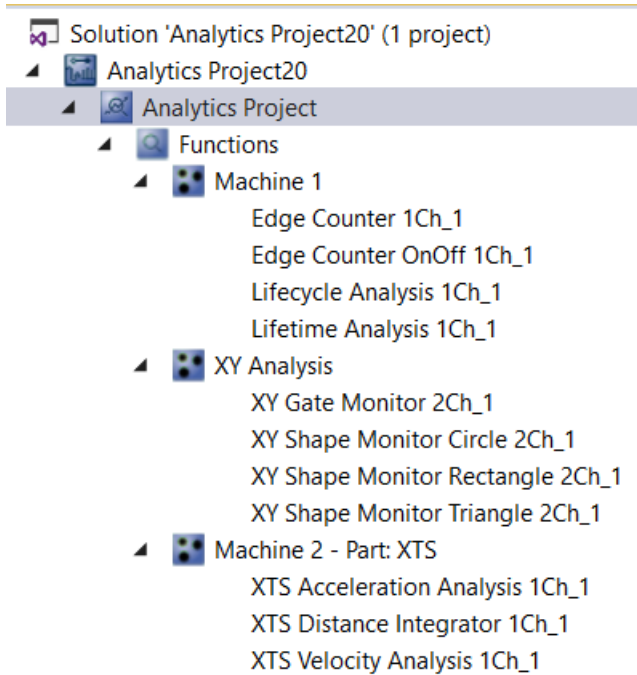
Ein Netzwerk kann direkt von der Analytics Project-Startseite oder über das Kontextmenü hinzugefügt werden.



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



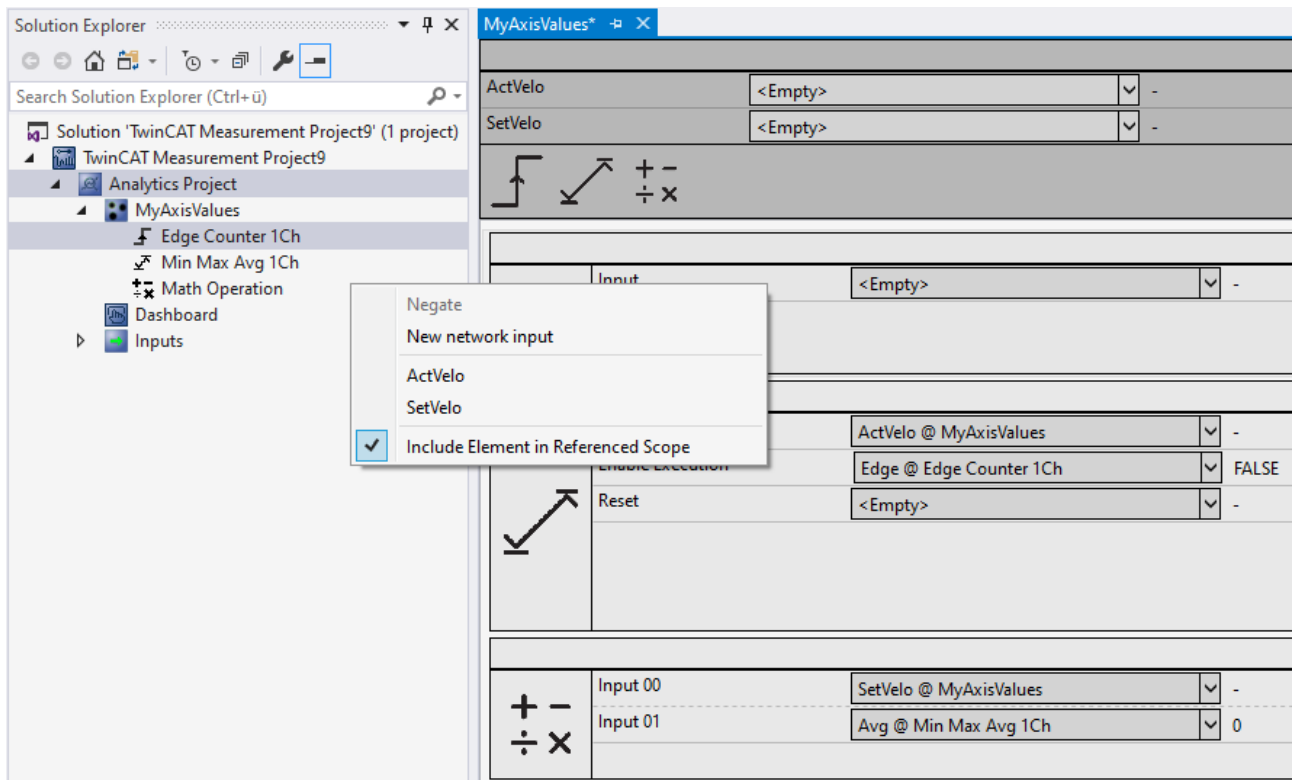
Des Weiteren haben Sie die Möglichkeit, die Netzwerke umzubenennen (F2 auf ausgewähltem Netzwerkelement im Projektmappen-Explorer), zum Beispiel, um Netzwerke für verschiedene Maschinen, Maschinenteile oder andere inhaltliche Verbindungen zu erstellen.



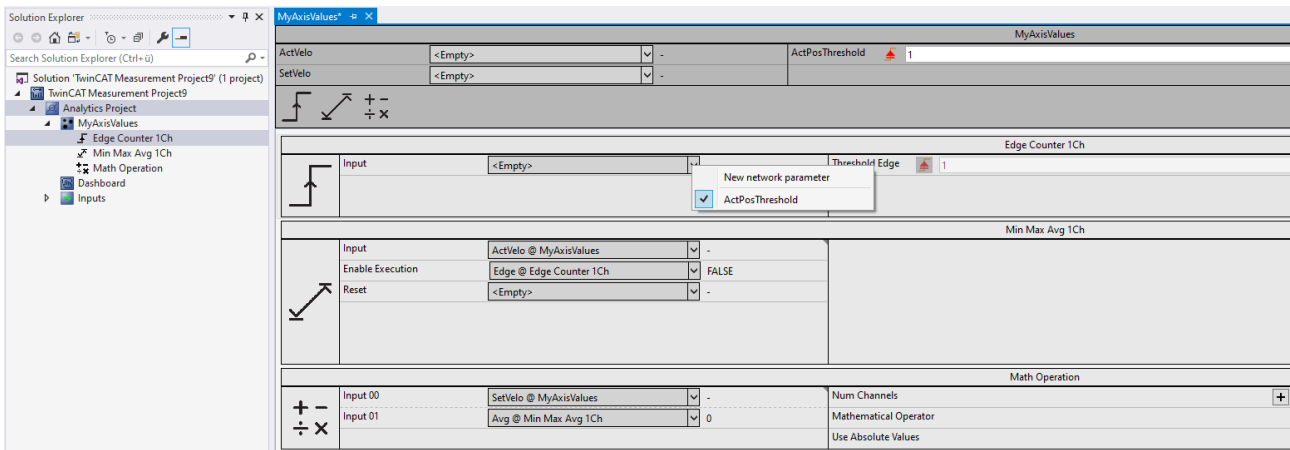
6.3.1.1 Netzwerke als Template

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 instanzieren.

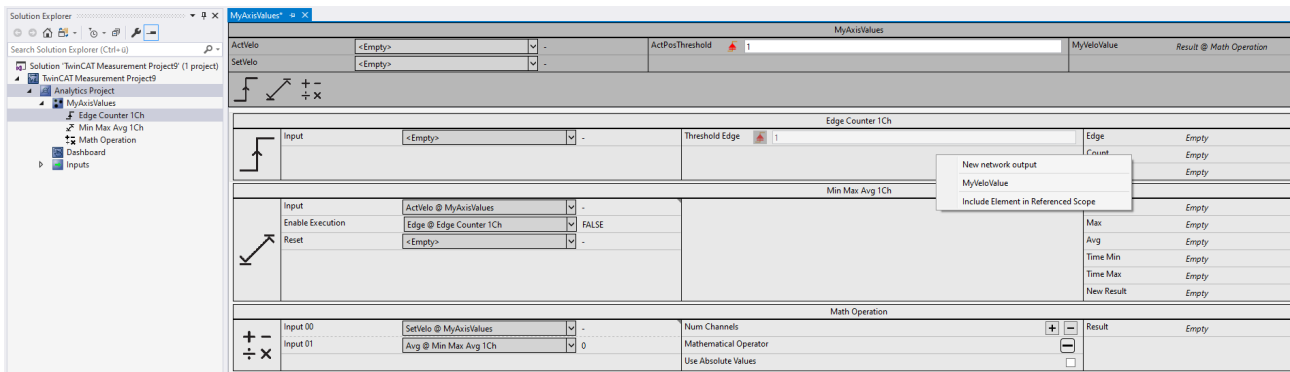
Wenn Sie eine Analyse erstellt haben, müssen Sie zunächst auswählen, welche der verfügbaren Eingänge nach außen am Netzwerk sichtbar sein sollen. Wählen Sie dazu den gewünschten Eingang aus und wechseln per Rechtsklick in das Kontextmenu. Dort können Sie den Eingang mit einem bereits vorhandenen Netzwerkeingang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



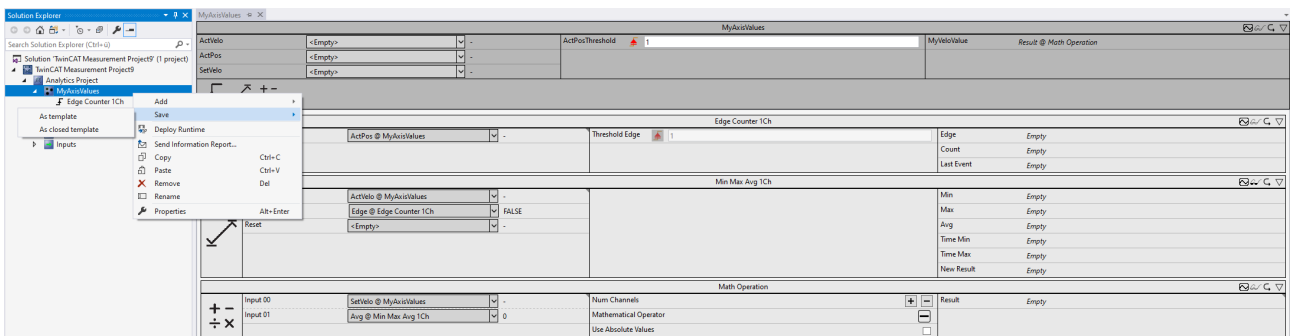
Die gleiche Vorgehensweise gibt es für Parameter und für die Ausgänge.



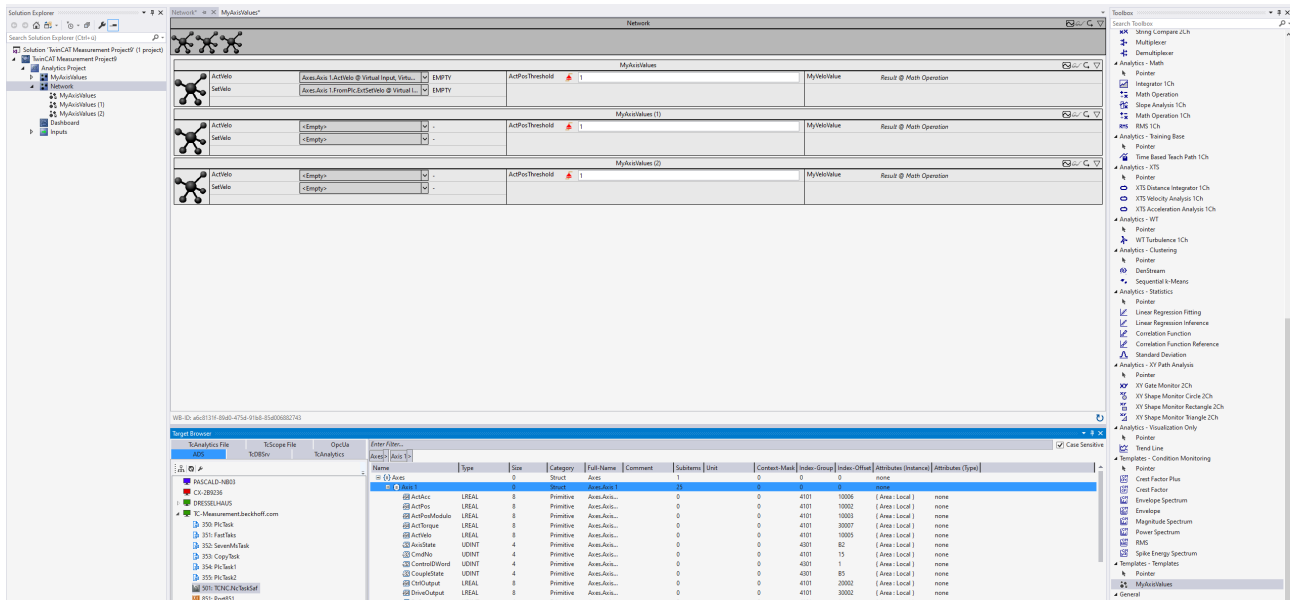
Neben diesen Ein- und Ausgängen gibt es auch die Möglichkeit, sogenannte dynamische Ein- und Ausgänge zu spezifizieren. Diese werden Ihnen automatisch angeboten, sobald einer der gewählten Algorithmen diese Funktion unterstützt. Hintergrund ist dabei die Möglichkeit, die Anzahl der Eingänge oder Ausgänge 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 Netzwerktemple

Eine erstellte Scope-Konfiguration kann zusammen mit dem dazugehörigen Netzwerk in einem Netzwerktemple abgespeichert werden [► 308], um bei einer erneuten Verwendung eines Netzwerks automatisch die gleiche Scope-Konfiguration zu erhalten.

6.3.2 Enable Disable

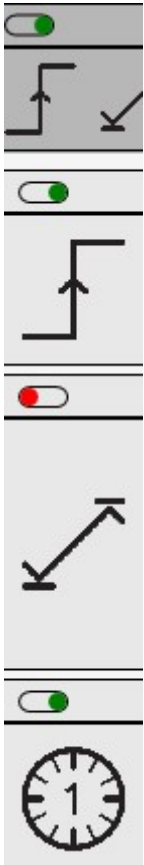
Die TwinCAT Analytics Workbench bietet die Funktionalität einzelne Algorithmen oder auch komplexe Netzwerke in einer Datenanalyse ein- bzw. auszuschalten. In diesem Dokument werden Nutzung und Vorteile dieser Funktion erläutert.

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:

⏻
Zählwerk
🗑️ 📄 ↺ ⌵

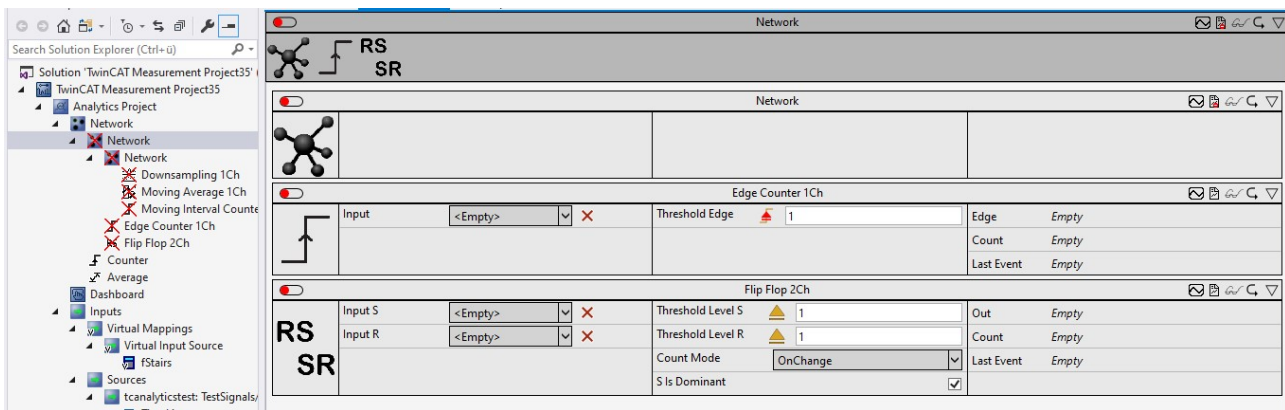
↗	Input	fStairs, Virtual Inp... ✖	Threshold Edge 🔥	1	Edge	Empty
					Count	Empty
					Last Event	Empty

⏻
Produktionsdurchschnitt
🗑️ 📄 ↺ ⌵

↘	Input	Count @ Zählwerk ✖				Min	Empty
						Max	Empty
						Avg	Empty
						Time Min	Empty
						Time Max	Empty

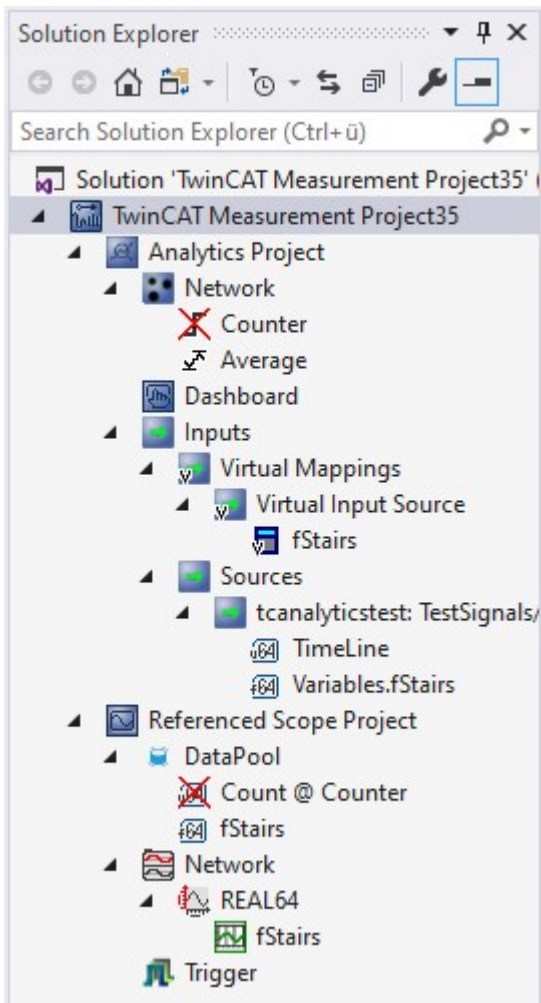
Wird ein Netzwerk deaktiviert so wird die Abarbeitung aller Algorithmen unterhalb des Netzwerkes sowie dessen Unternetzwerke (Rekursiv) ebenfalls deaktiviert.

Deaktivierte Netzelemente mit ihren Unterelementen:



Sollten Sie ein referenziertes Scope verwenden (siehe [Interaktion mit Scope](#) | 303) so werden verknüpfte Elemente des Scope ebenfalls durch das Deaktivieren beeinflusst.

Durch das Deaktivieren des Algorithmus wird die zugehörige Scope Acquisition ebenfalls deaktiviert:



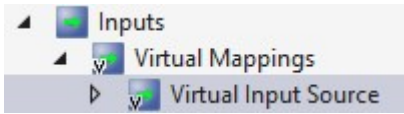
6.3.3 Static Values

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.

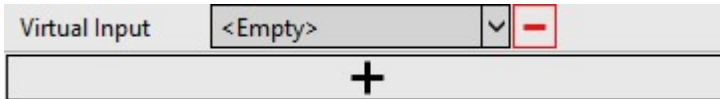
Die Verwendung von statischen Werten in Analysealgorithmen bietet mehrere Vorteile. Zum einen kann die Leistung des Algorithmus verbessert werden, indem Berechnungen vermieden werden, die ansonsten, während jeder Ausführung des Algorithmus durchgeführt werden müssten. Stattdessen kann der

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.

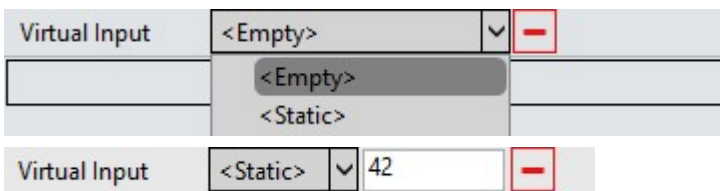
Statische Werte werden in der TwinCAT Analytics Workbench unter den Virtual Inputs einsortiert, um diese an zentraler Stelle zu erzeugen und später zu verwalten.



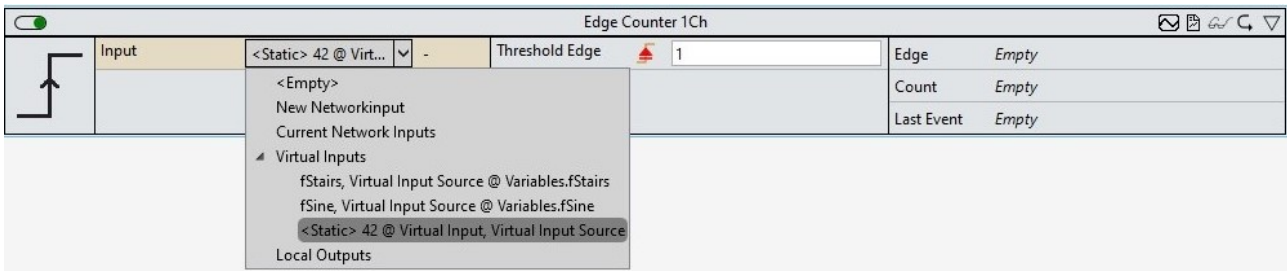
In dem Editor der Virtual Input Source kann über das + ein neuer Virtual Input angelegt, oder ein vorhandener bearbeitet werden.



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.



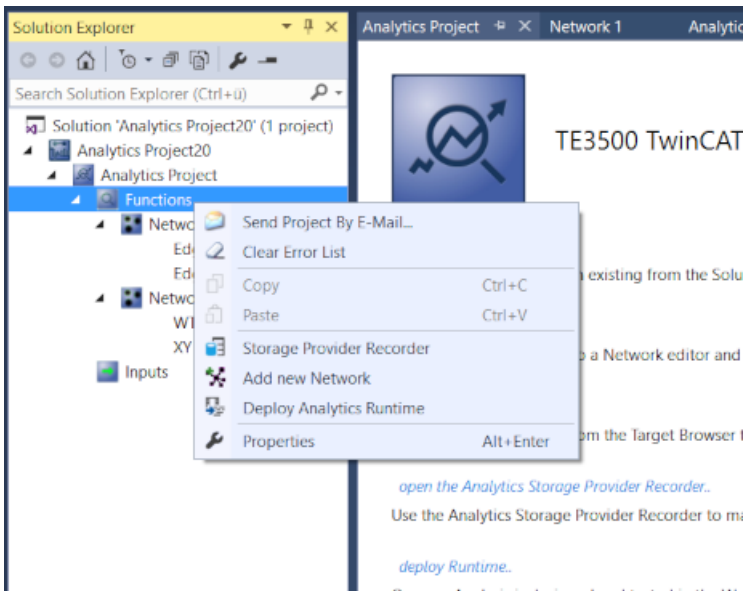
Statische Werte können auch auf Netzwerk Inputs eingesetzt werden. Hier würde die Verknüpfung mit den abhängigen Eingängen der einzelnen Module genau gleich dem Verhalten bei entsprechendem Dateninputs ablaufen.

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.

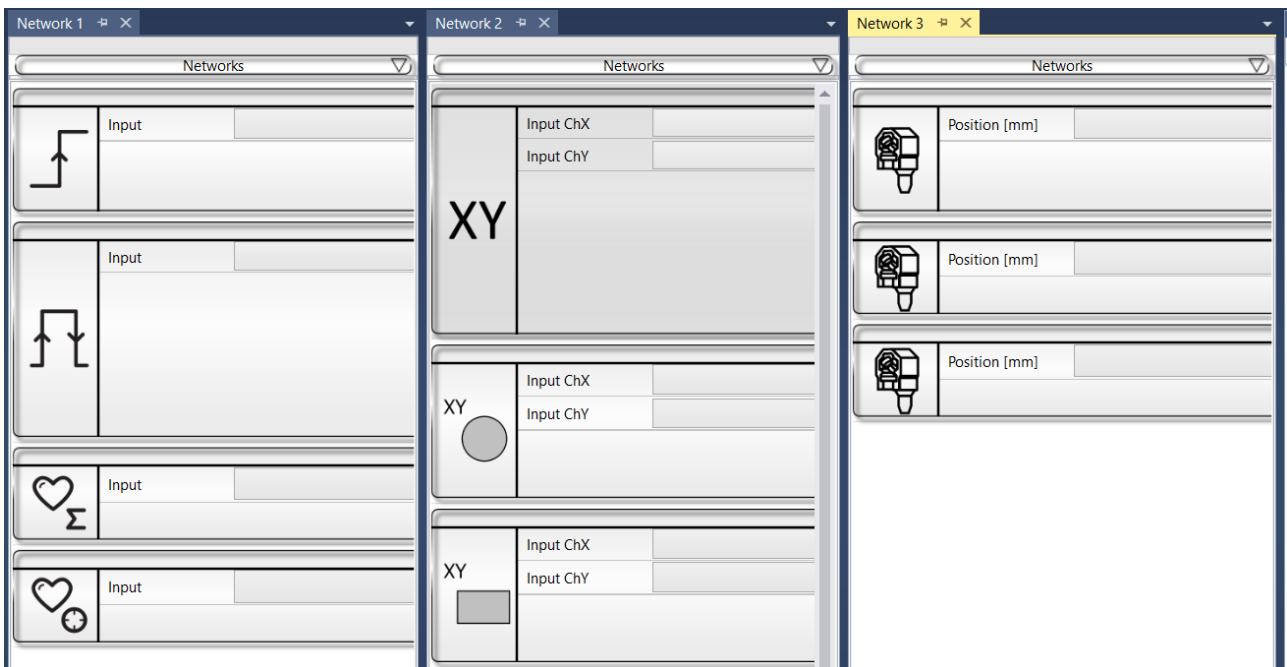
6.4 Netzwerke

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.

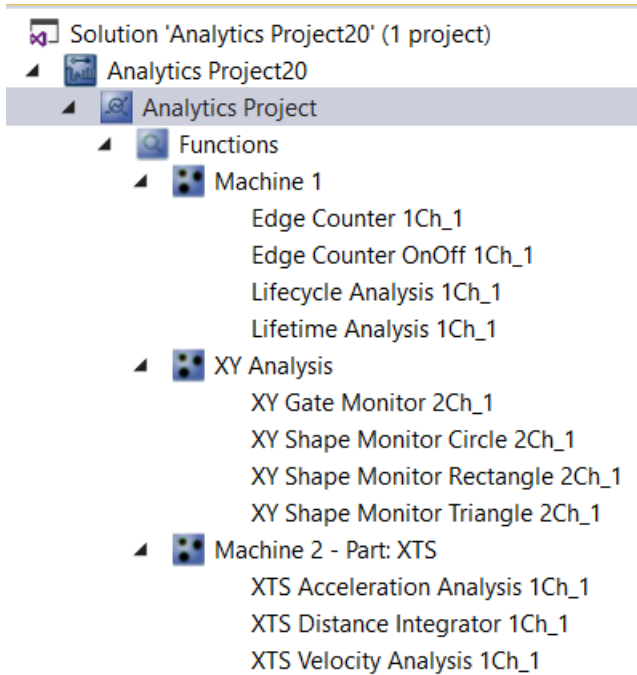
Ein Netzwerk kann direkt von der Analytics Project-Startseite oder über das Kontextmenü hinzugefügt werden.



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



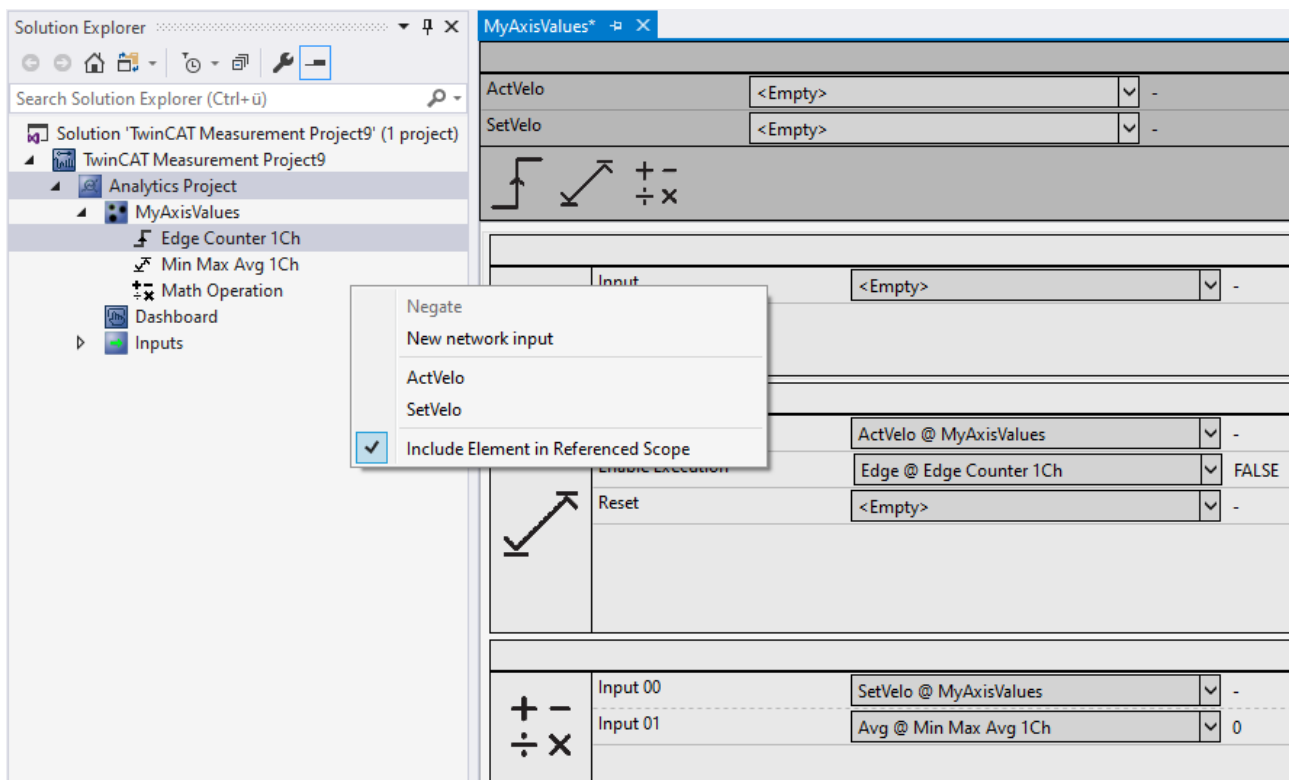
Des Weiteren haben Sie die Möglichkeit, die Netzwerke umzubenennen (F2 auf ausgewähltem Netzwerkelement im Projektmappen-Explorer), zum Beispiel, um Netzwerke für verschiedene Maschinen, Maschinenteile oder andere inhaltliche Verbindungen zu erstellen.



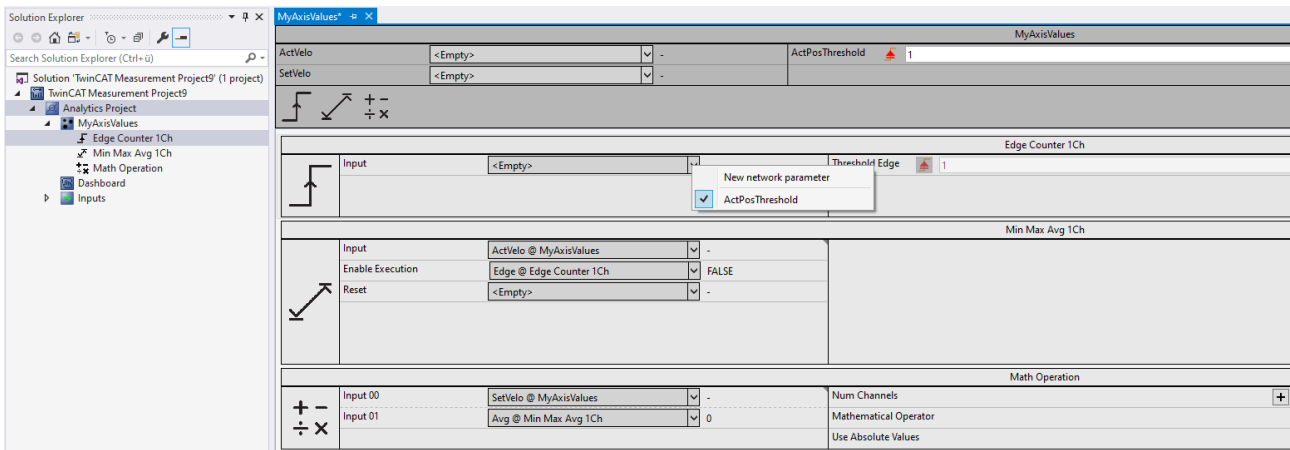
6.4.1 Netzwerke als Template

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 instanzieren.

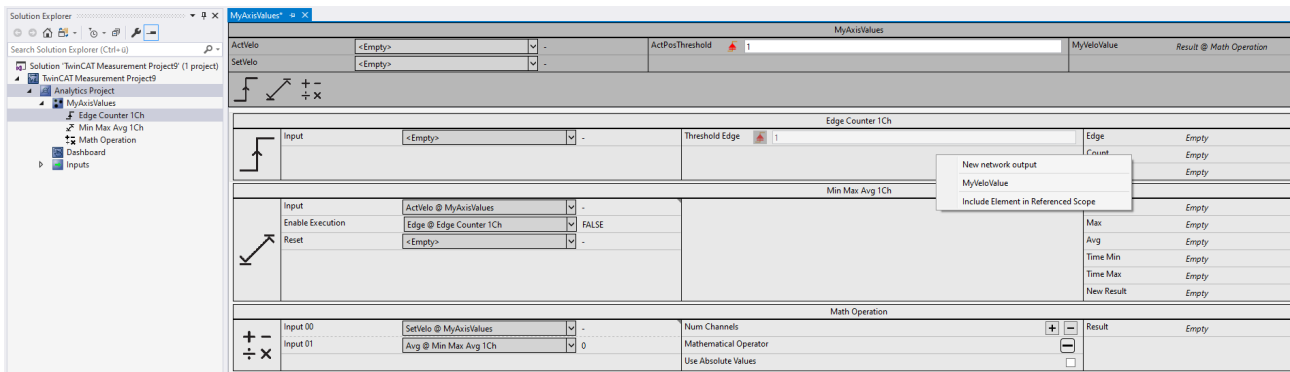
Wenn Sie eine Analyse erstellt haben, müssen Sie zunächst auswählen, welche der verfügbaren Eingänge nach außen am Netzwerk sichtbar sein sollen. Wählen Sie dazu den gewünschten Eingang aus und wechseln per Rechtsklick in das Kontextmenu. Dort können Sie den Eingang mit einem bereits vorhandenen Netzwerkeingang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



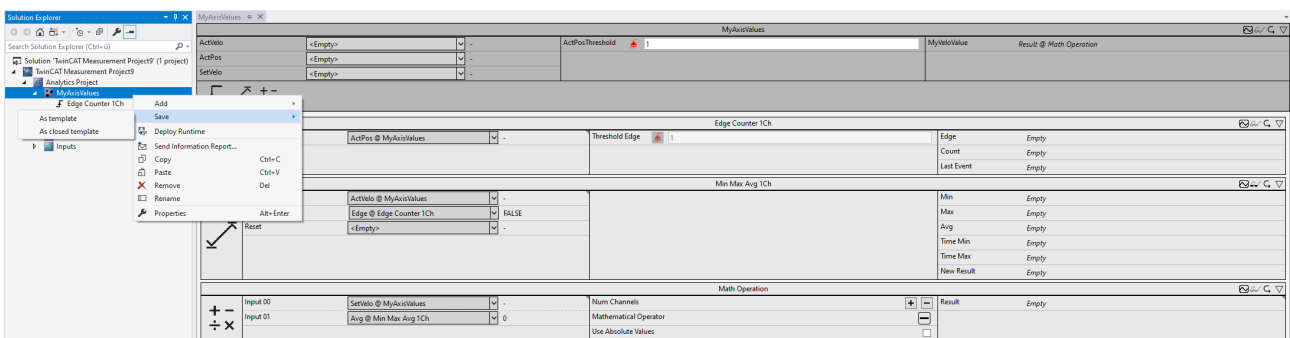
Die gleiche Vorgehensweise gibt es für Parameter und für die Ausgänge.



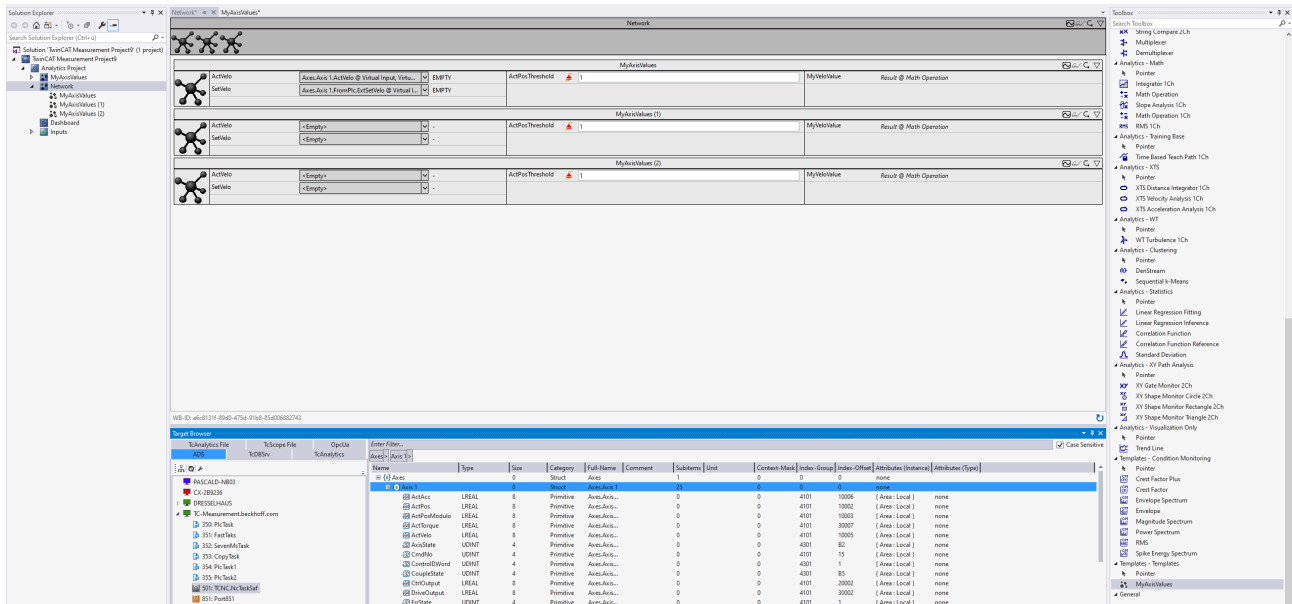
Neben diesen Ein- und Ausgängen gibt es auch die Möglichkeit, sogenannte dynamische Ein- und Ausgänge zu spezifizieren. Diese werden Ihnen automatisch angeboten, sobald einer der gewählten Algorithmen diese Funktion unterstützt. Hintergrund ist dabei die Möglichkeit, die Anzahl der Eingänge oder Ausgänge 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.



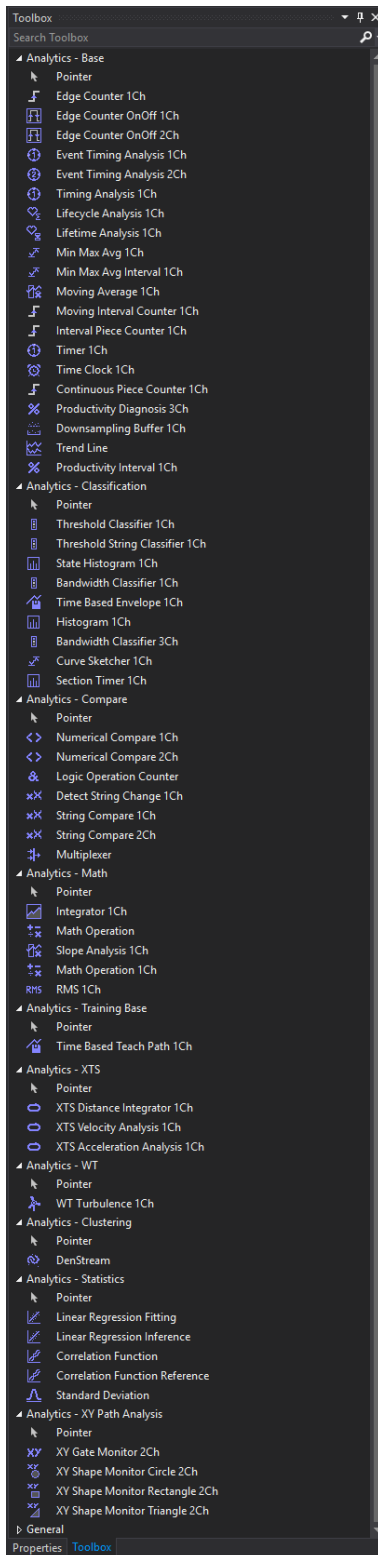
Eine besondere Funktion bietet der Target Browser an. Sind für das Netzwerk die Namen so gewählt worden, dass diese exakt mit Variablen einer Struktur übereinstimmen, können Sie eine vollständige Struktur wie z. B. eine Achsen Struktur direkt auf einen der x-Eingänge eines Netzwerks ziehen. Alle übereinstimmenden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

Scope-Konfiguration gespeichert im Netzwerktemple

Eine erstellte Scope-Konfiguration kann zusammen mit dem dazugehörigen Netzwerk in einem Netzwerktemple abgespeichert werden [► 308], um bei einer erneuten Verwendung eines Netzwerks automatisch die gleiche Scope-Konfiguration zu erhalten.

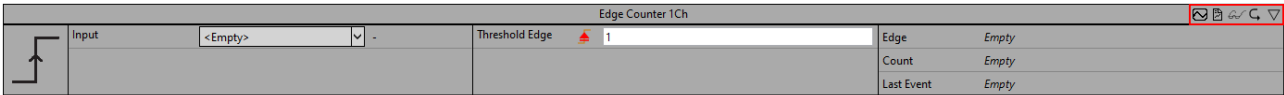
6.5 Algorithmen

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 entwickelt werden (siehe [C++ Lambda Funktionen](#) [► 237]). Zusätzlich besteht die Möglichkeit, Algorithmen aus anderen Bibliotheken zu verwenden, die in die Analytics Toolbox eingebunden sind (siehe [Algorithmen aus weiteren TwinCAT Bibliotheken](#) [► 268]). Durch den [Analytics Custom Toolbox Cleaner](#) [► 267] kann die Toolbox hinsichtlich anwenderspezifischen Elementen, wie zum Beispiel den Templates oder den C++ Lambda Funktionen bereinigt werden.

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



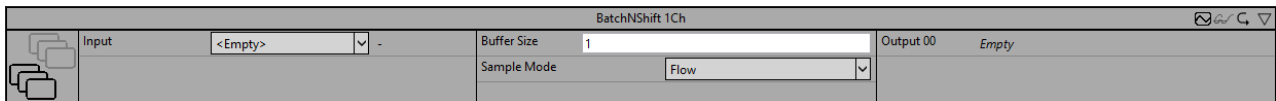
- **Include in Referenced Scope:** Wenn Sie auf das *Included in Referenced Scope* Icon klicken, werden die Algorithmusausgänge zu einem referenzierten Scope-Projekt hinzugefügt.
- **Include in Report:** Wenn Sie auf das *Include in Report* Icon klicken, wird der Algorithmus mit der konfigurierten Visualisierung 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 boolesches 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.

6.5.1 Analytics - Base

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

6.5.1.1 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 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:

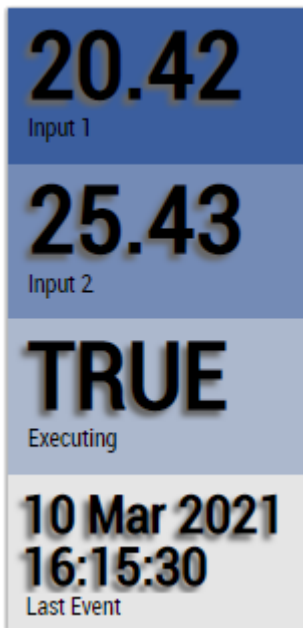
1. Das Table Control oder Multivalue Control visualisiert die Ausgangswerte *Output Value 00..n*

Data Table Vertical

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

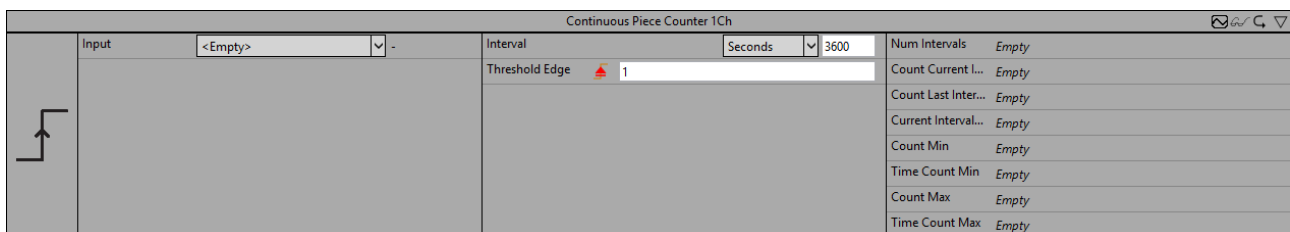
DataTable Horizontal

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



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

6.5.1.2 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 Flankenähler 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

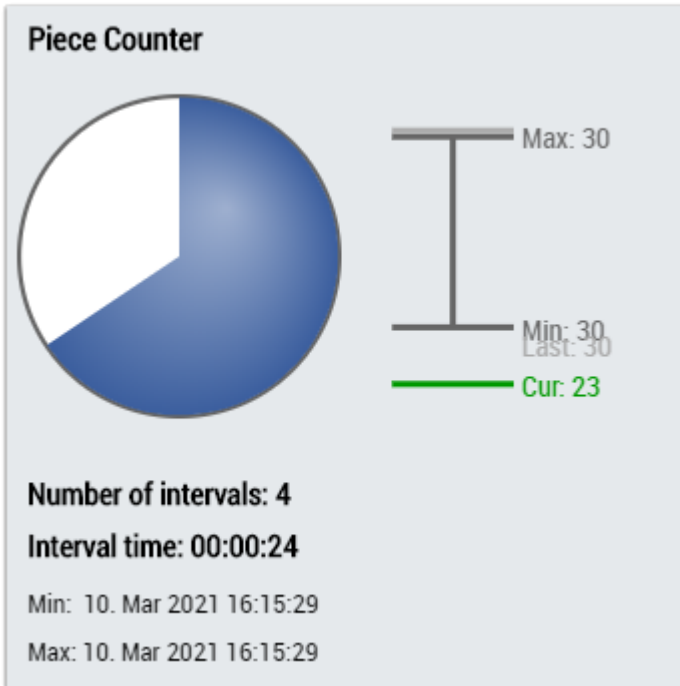
- **Num Interval:** Zeigt die Anzahl der Intervalle an.
- **Count Last Interval:** Zeigt die Anzahl der Teile im letzten Intervall an.
- **Count Current Interval:** Zeigt die Anzahl der Teile im aktuellen Intervall an.
- **Count Min:** Zeigt die Mindestanzahl der Teile in einem Intervall an.
- **Count 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.

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

20.42

Input 1

25.43

Input 2

TRUE

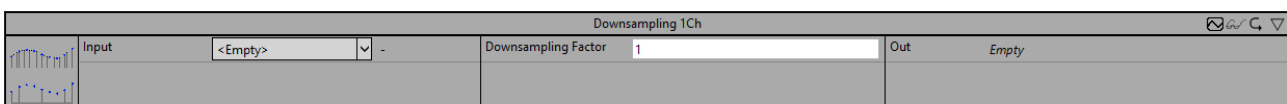
Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.1.3 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

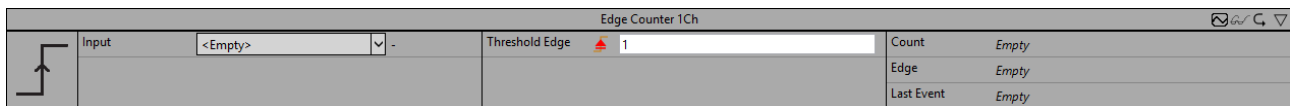
Für den Algorithmus *Downsampling 1Ch* sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

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



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

6.5.1.4 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 Flankenähler 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.

Single Value

8

Last event:
10. Mar 2021 16:24:29

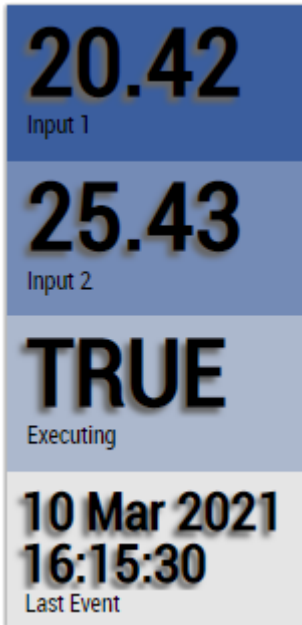
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Edge, Count, Last Event.

Data Table Vertical

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

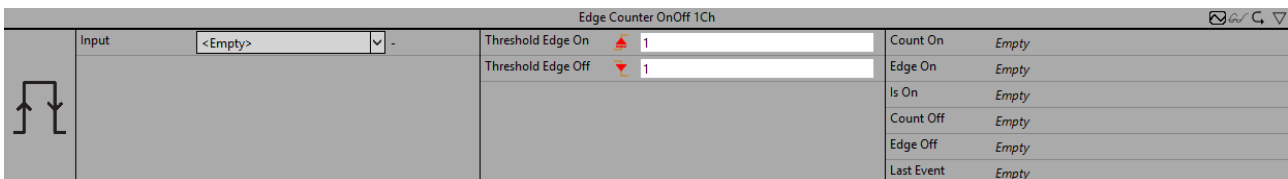
DataTable Horizontal

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



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

6.5.1.5 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.

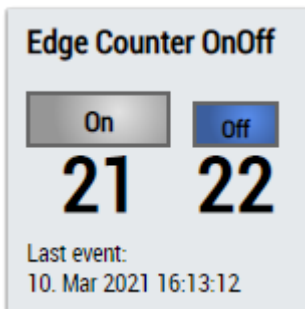
Ausgangswerte

- **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:

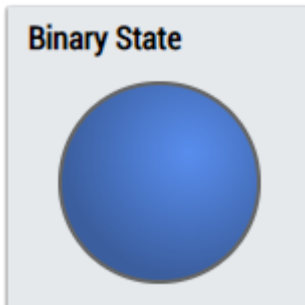
1. Das EdgeCounterOnOff Control visualisiert die Ausgangswerte Is On, Count On, Count Off und Last Event.



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

20.42

Input 1

25.43

Input 2

TRUE

Executing

**10 Mar 2021
16:15:30**

Last Event

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

6.5.1.6 Edge Counter On Off 2Ch

Edge Counter OnOff 2Ch			
Input Ch1	<Empty>	-	Reset On Multiple On <input type="checkbox"/>
Input Ch2	<Empty>	-	Threshold Edge On 1
			Threshold Edge Off 1
			Count On Empty
			Edge On Empty
			Is On Empty
			Count Off Empty
			Edge Off Empty
			Last Event Empty

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.

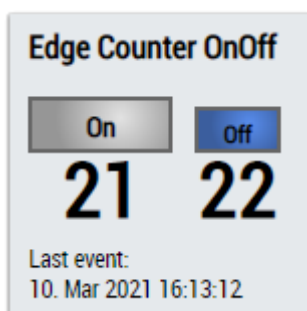
Ausgangswerte

- **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.

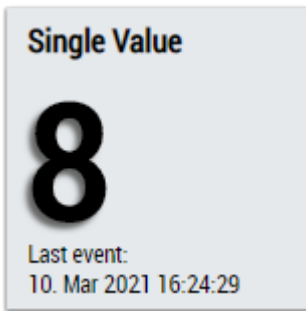
Standard HMI Controls

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

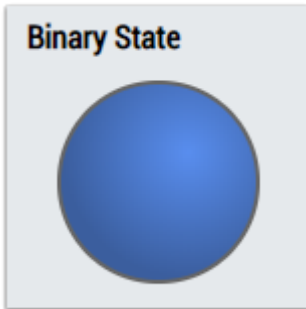
1. Das *EdgeCounterOnOff* Control visualisiert die Ausgangswerte *Is On*, *Count On*, *Count Off* und *Last Event*.



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



3. Das BinaryState Control visualisiert den Ausgangswert Is On.



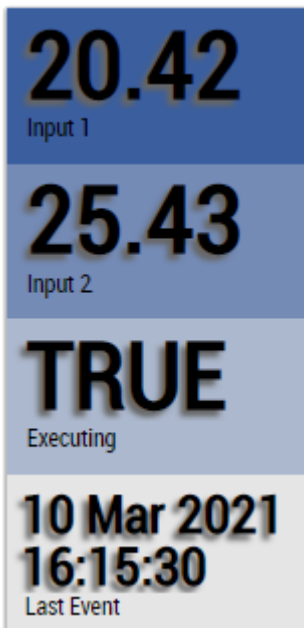
4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Flanken (Is On, Edge On, Edge Off), Count On, Count Off, Last Event.

Data Table Vertical

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

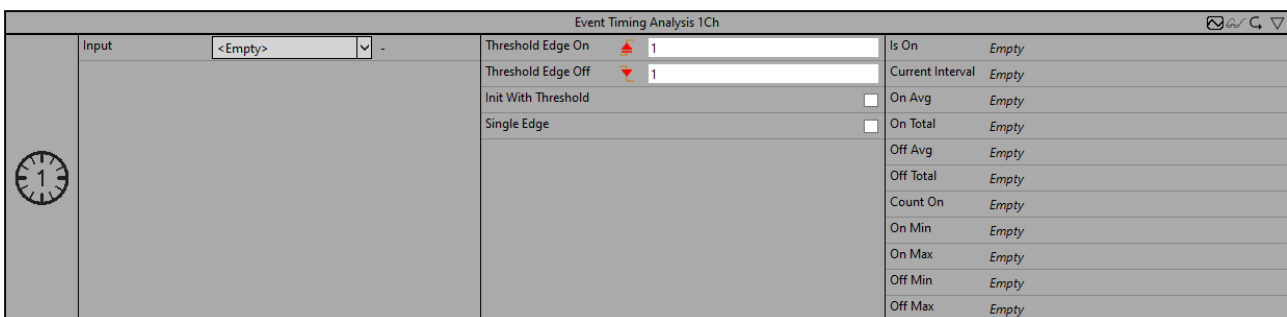
DataTable Horizontal

	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 2Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.1.7 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.

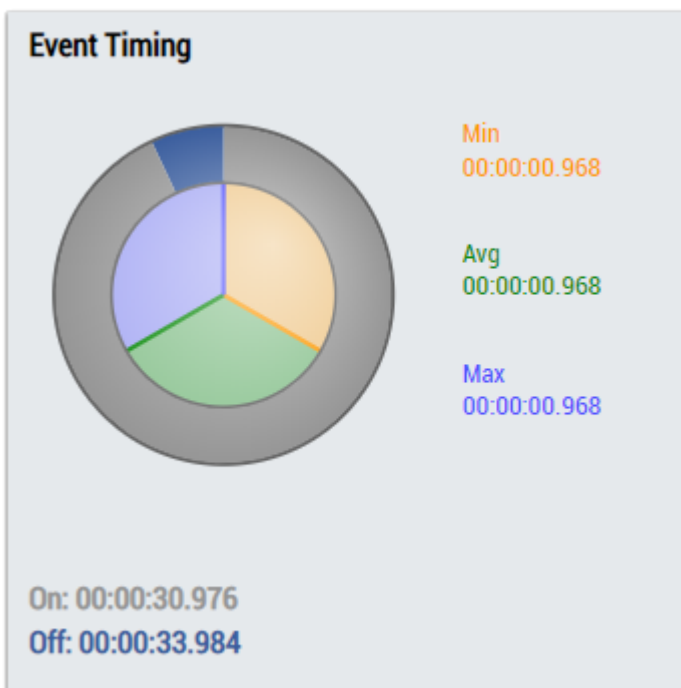
Ausgangswerte

- **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.
- **On Min:** Zeigt die Minimum-Zeit an, für die der „Is On“-Wert *TRUE* ist.
- **On Max:** Zeigt die Maximum-Zeit an, für die der „Is On“-Wert *TRUE* ist.
- **On Avg:** Zeigt die durchschnittliche Zeit an, für die der „Is On“-Wert *TRUE* ist.
- **On Total:** Zeigt die Gesamtzeit an, für die der „Is On“-Wert *TRUE* ist.
- **Off Min:** Zeigt die Minimum-Zeit an, für die der „Is On“-Wert *FALSE* ist.
- **Off Max:** Zeigt die Maximum-Zeit an, für die der „Is On“-Wert *FALSE* ist.
- **Off Avg:** Zeigt die durchschnittliche Zeit an, für die der „Is On“-Wert *FALSE* 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 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.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021
16:15:30

Last Event

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

6.5.1.8 Event Timing Analysis 2Ch

Event Timing Analysis 2Ch			
Input Ch1	<Empty>	-	Reset On Multiple On <input type="checkbox"/>
Input Ch2	<Empty>	-	Threshold Edge On 1
			Threshold Edge Off 1
			Init With Threshold <input type="checkbox"/>
			Count On <i>Empty</i>
			Is On <i>Empty</i>
			On Avg <i>Empty</i>
			On Total <i>Empty</i>
			On Min <i>Empty</i>
			On Max <i>Empty</i>
			Off Avg <i>Empty</i>
			Off Total <i>Empty</i>
			Off Min <i>Empty</i>
			Off Max <i>Empty</i>
			Current Interval <i>Empty</i>

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 Flankenähler 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*.

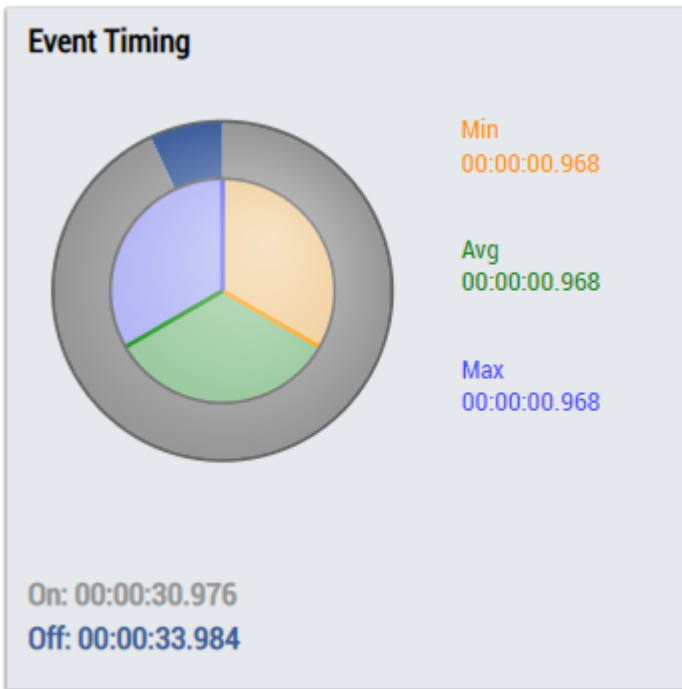
Ausgangswerte

- **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.
- **On Min:** Zeigt die Minimum-Zeit an, für die der „Is On“-Wert *TRUE* ist.
- **On Max:** Zeigt die Maximum-Zeit an, für die der „Is On“-Wert *TRUE* ist.
- **On Avg:** Zeigt die durchschnittliche Zeit an, für die der „Is On“-Wert *TRUE* ist.
- **On Total:** Zeigt die Gesamtzeit an, für die der „Is On“-Wert *TRUE* ist.
- **Off Min:** Zeigt die Minimum-Zeit an, für die der „Is On“-Wert *FALSE* ist.
- **Off Max:** Zeigt die Maximum-Zeit an, für die der „Is On“-Wert *FALSE* ist.
- **Off Avg:** Zeigt die durchschnittliche Zeit an, für die der „Is On“-Wert *FALSE* ist.
- **Off Total:** Zeigt die Gesamtzeit an, für die der „Is On“-Wert *FALSE* ist.
- **Count On:** Zählt die Anzahl der ausgelösten On-Ereignisse.

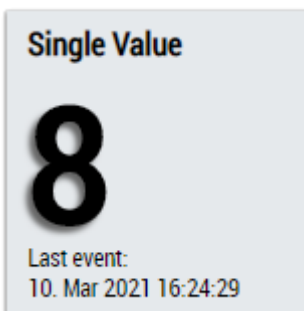
Standard HMI Controls

Für den Algorithmus *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.

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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

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

6.5.1.9 Flip Flop 2Ch

Flip Flop 2Ch							
RS SR	Input S	<Empty>	-	Threshold Level S	▲ 1	Out	Empty
	Input R	<Empty>	-	Threshold Level R	▲ 1	Count	Empty
				Count Mode	OnChange	Last Event	Empty
				S Is Dominant	<input checked="" type="checkbox"/>		

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*.

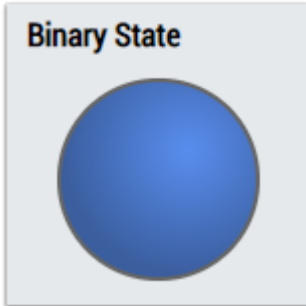
Ausgangswerte

- **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*.



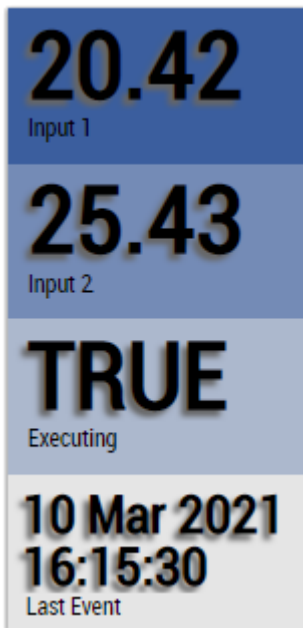
2. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte: *Out*, *Count*, *Last Event*.

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 *Flip Flop 2Ch* mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.1.10 Interval Piece Counter 1Ch

Interval Piece Counter 1Ch			
Input	<Empty>	Interval	Seconds 1
Start Interval	<Empty>	Reset On Multiple Start	<input type="checkbox"/>
		Threshold Edge	▲ 1
		Num Intervals	Empty
		Count Current I...	Empty
		Count Last Inter...	Empty
		Current Interval...	Empty
		Executing Interval	Empty
		Count Min	Empty
		Time Count Min	Empty
		Count Max	Empty
		Time Count Max	Empty

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 Flankenähler 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

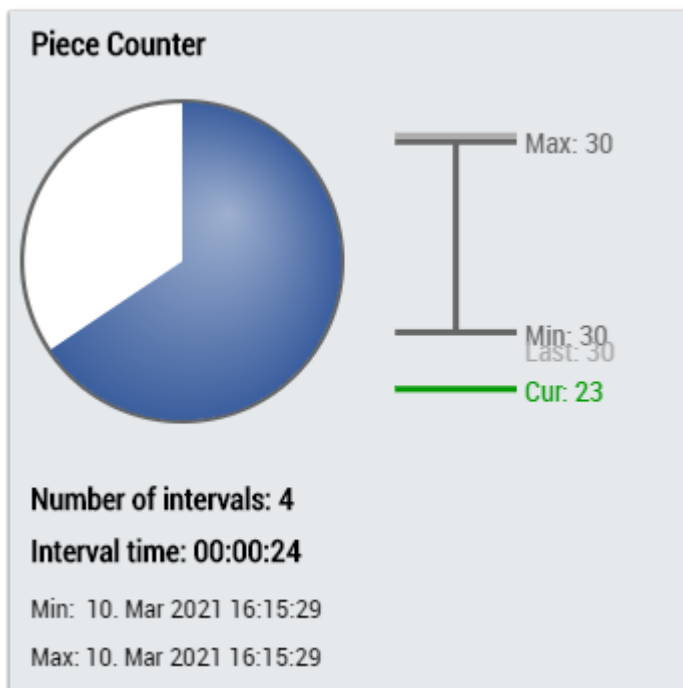
- **Executing Interval:** Zeigt *True* an, wenn die Berechnung aktiv ist und das Intervall läuft, ansonsten *False*.
- **Num Intervals:** Zeigt die Anzahl der Intervalle an.

- **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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.1.11 Lifecycle Analysis 1Ch

 	Input <input type="text" value="<Empty>"/>	Lifecycle Analysis 1Ch Estimated Cycles <input type="text" value="1000"/> Threshold Edge <input type="text" value="1"/>	Cycles Elapsed <i>Empty</i> Cycles Remaining <i>Empty</i>
------	--	---	--

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 Flankenähler 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.

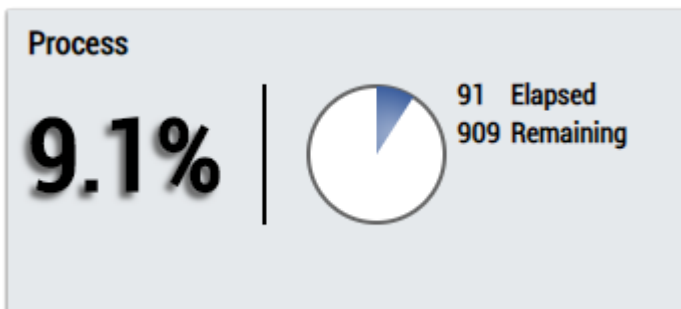
Ausgangswerte

- **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*.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.1.12 Lifetime Analysis 1Ch

Lifetime Analysis 1Ch				
Input	<Empty>	Estimated Lifetime	Seconds	3600
		Threshold Level	=	1
		Time Elapsed		Empty
		Time Remaining		Empty

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.

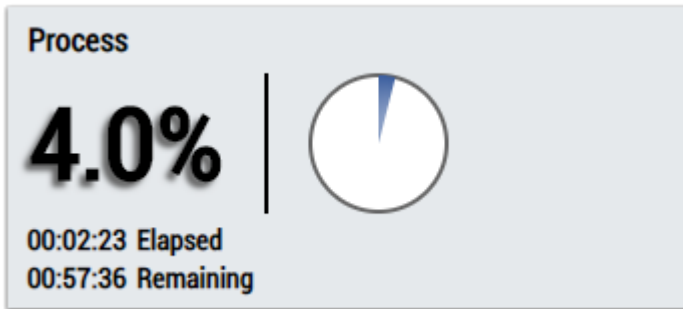
Ausgangswerte

- **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.



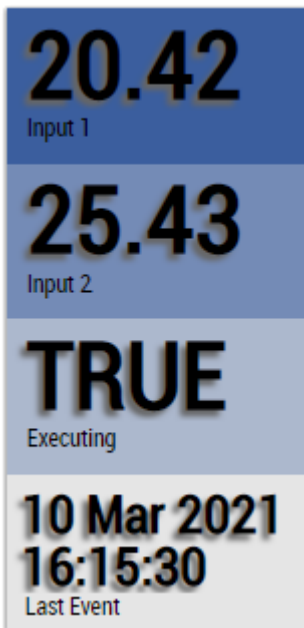
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Elapsed Lifetime, Remaining Lifetime.

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 kundenspezifische HMI Controls gemappt werden.

6.5.1.13 Min Max Avg 1Ch

Min Max Avg 1Ch			
Input	<Empty>	Avg	Empty
		Min	Empty
		Time Min	Empty
		Max	Empty
		Time Max	Empty

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.

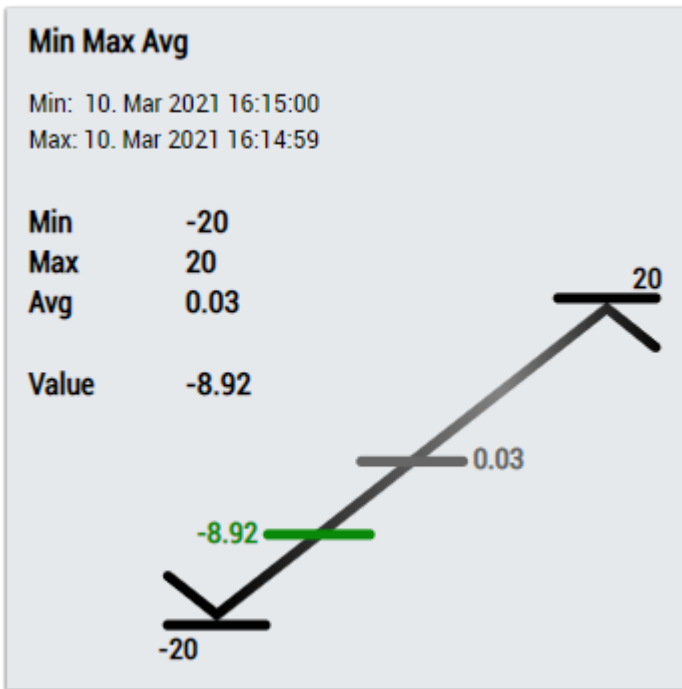
Ausgangswerte

- **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.

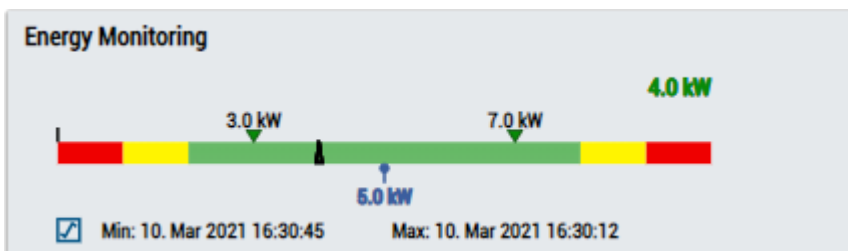
Standard HMI Controls

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

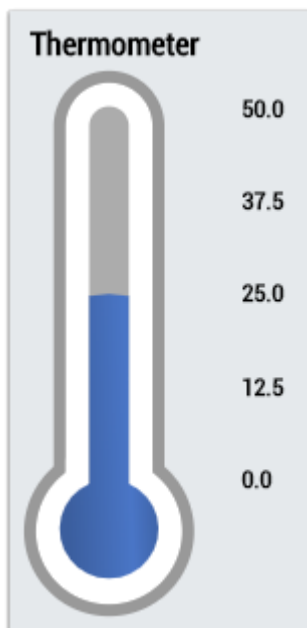
1. Das MinMaxAvg Control visualisiert die Ausgangswerte Min, Max, Avg, Time Min und Time Max sowie den Eingangswert der Daten.



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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021

16:15:30

Last Event

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

6.5.1.14 Min Max Avg Interval 1Ch

Min Max Avg Interval 1Ch			
Input	<Empty>	Interval	Seconds 1
		Avg	Empty
		Current Interval...	Empty
		Min	Empty
		Time Min	Empty
		Max	Empty
		Time Max	Empty

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.

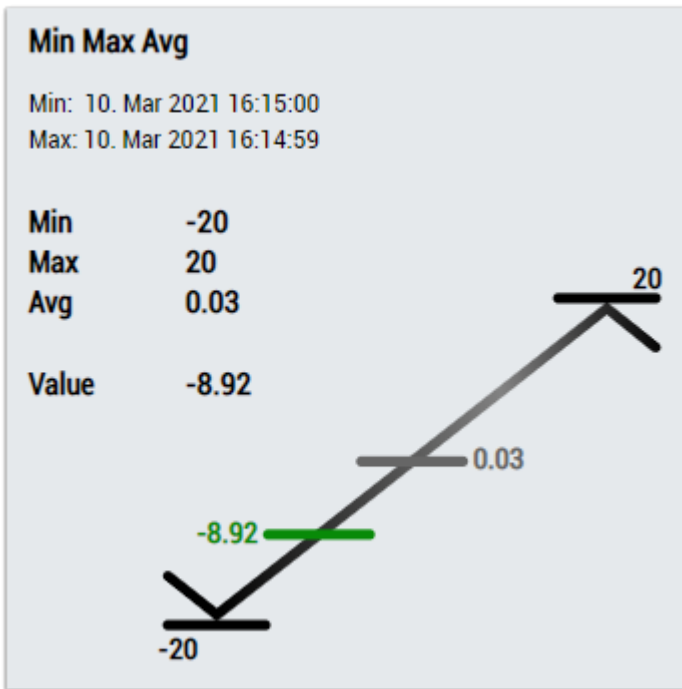
Ausgangswerte

- **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.

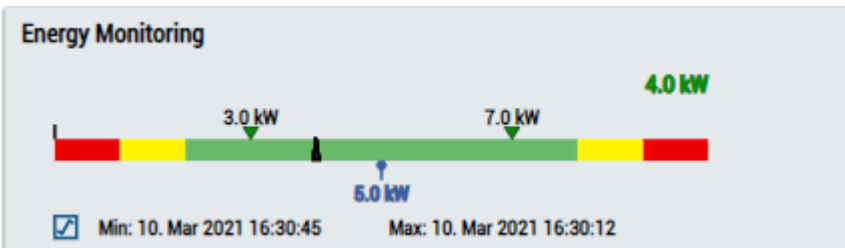
Standard HMI Controls

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

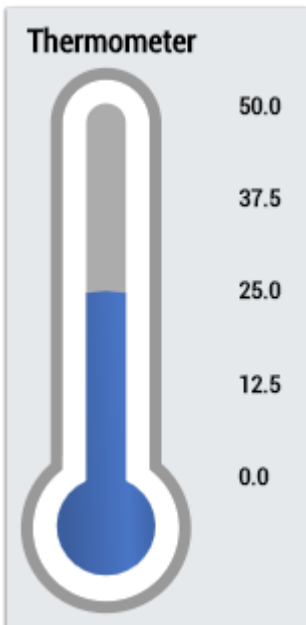
1. Das *MinMaxAvg* Control visualisiert die Ausgangswerte *Min*, *Max*, *Avg*, *Time Min* und *Time Max* sowie den Eingangswert der Daten.



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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.1.15 Moving Average 1Ch

Moving Average 1Ch ☒ ⚙ ⏴ ⏵			
	Input	<Empty>	-
	Num Values	1	Moving Avg <i>Empty</i>
	Startup Behaviour	AvgOverExisting	Moving Min <i>Empty</i>
			Moving Max <i>Empty</i>

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.

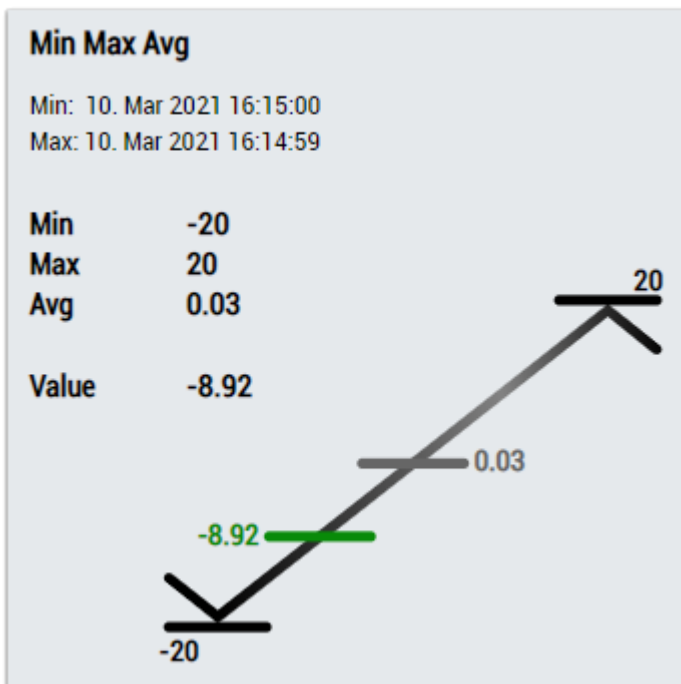
Ausgangswerte

- **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.

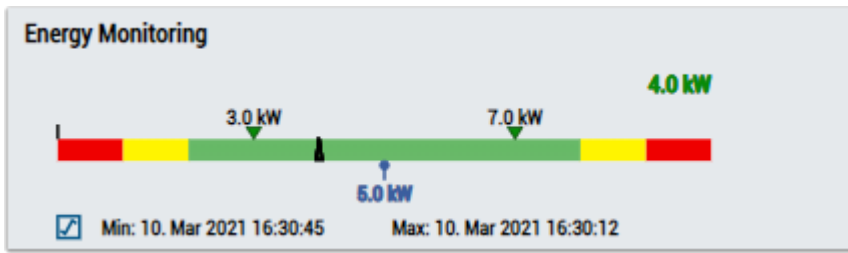
Standard HMI Controls

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

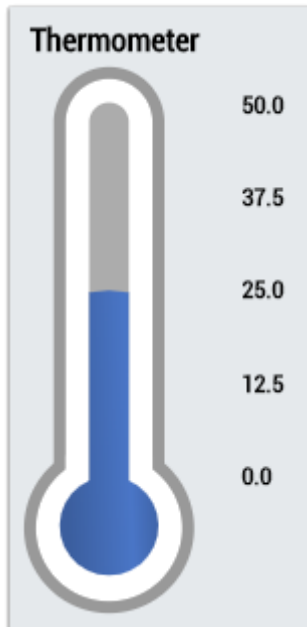
1. Das *MinMaxAvg* Control visualisiert die Ausgangswerte *Moving Min*, *Moving Max* und *Moving Avg* sowie den Eingangswert der Daten.



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 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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

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

6.5.1.16 Moving Interval Counter 1Ch

Moving Interval Counter 1Ch			
Input	<Empty>	Count Limit	20
Interval	Seconds	1	Counts In Interval
Threshold Edge	▲	1	Edge
			Limited
			Time First Count
			Time Last Count

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 Flankenähler 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.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.1.17 Overall Equipment Effectiveness (OEE)

Overall Equipment Effectiveness (OEE)							
%	Scheduled Time	<Empty>	-	Ideal Cycle Time	Seconds 1	OEE [%]	Empty
	Operating Time	<Empty>	-	Level OK / Warning	90	OEE Class	Empty
	Units Produced	<Empty>	-	Level Warning / Alarm	75	OEE Event Warning	Empty
	Defective Units	<Empty>	-			OEE Event Alarm	Empty
						Availability [%]	Empty
					Performance [%]	Empty	
					Quality [%]	Empty	

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 größer 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.

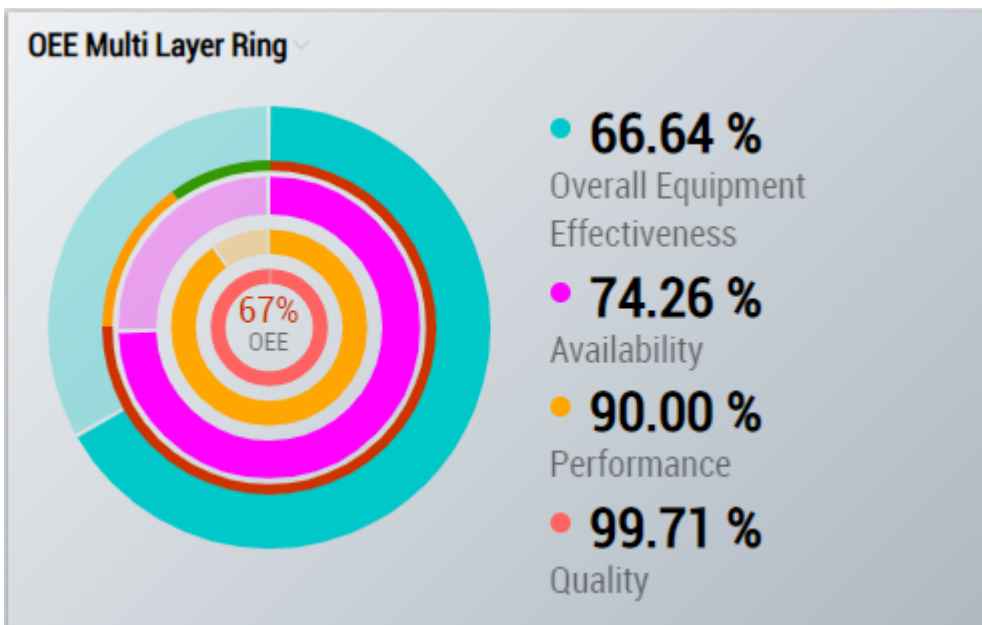
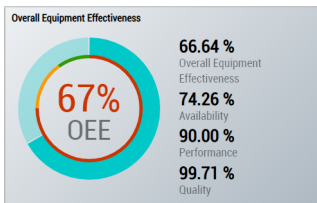
Ausgangswerte

- **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*.



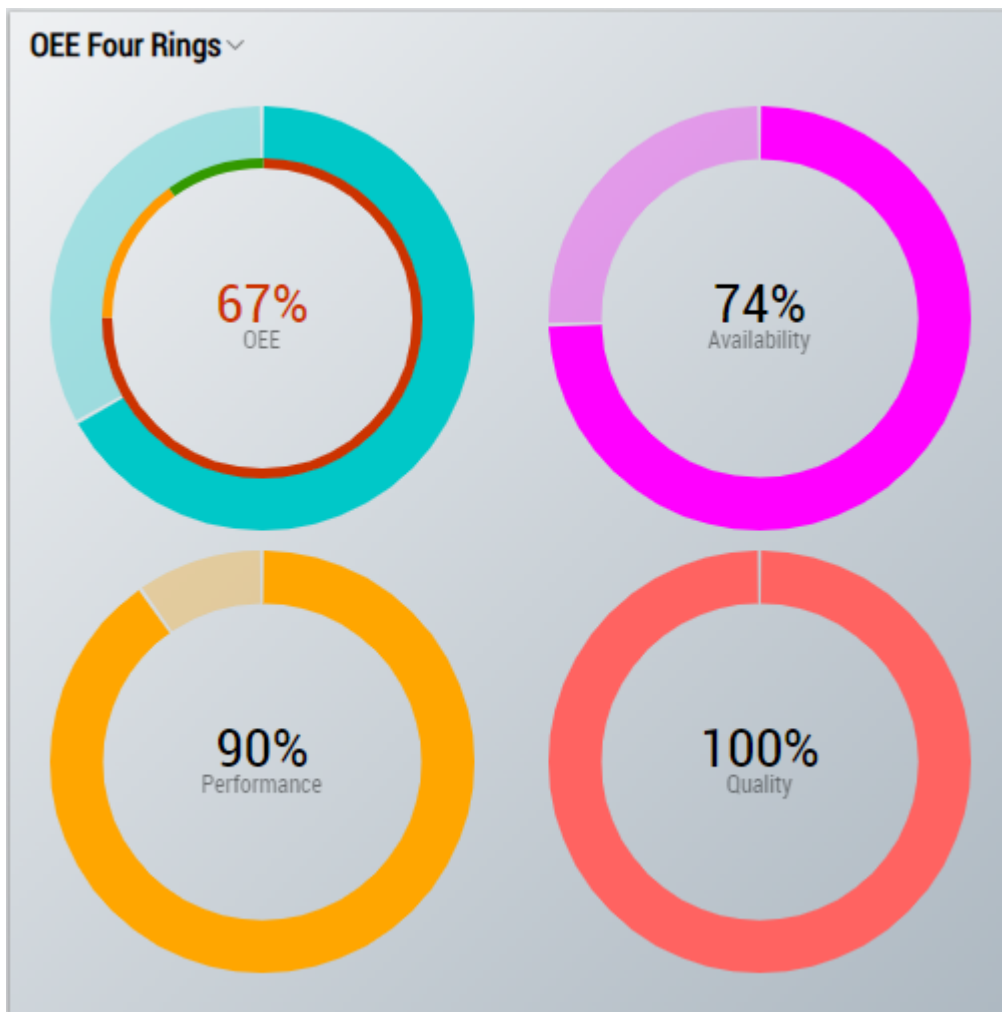


Abb. 1:

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

Data Table 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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

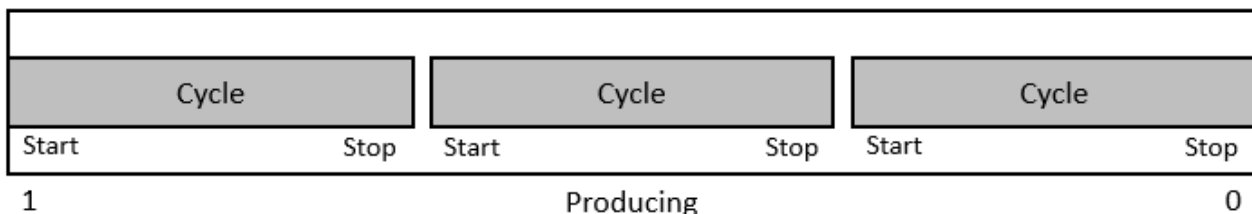
Last Event

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

6.5.1.18 Productivity Diagnosis 3Ch

Productivity Diagnosis 3Ch							
%	Is Producing	<Empty>	-	Threshold Level Producing	1	Producing	Empty
	Start Cycle	<Empty>	-	Threshold Edge Start Cycle	1	Productivity [%]	Empty
	Stop Cycle	<Empty>	-	Threshold Edge Stop Cycle	1	Productivity Last Cycle [%]	Empty
				Produced Pieces	1	Expected Productivity [%]	Empty
				Production Time	Seconds 1200	Elapsed Time	Empty
						Production Cycles	Empty

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.

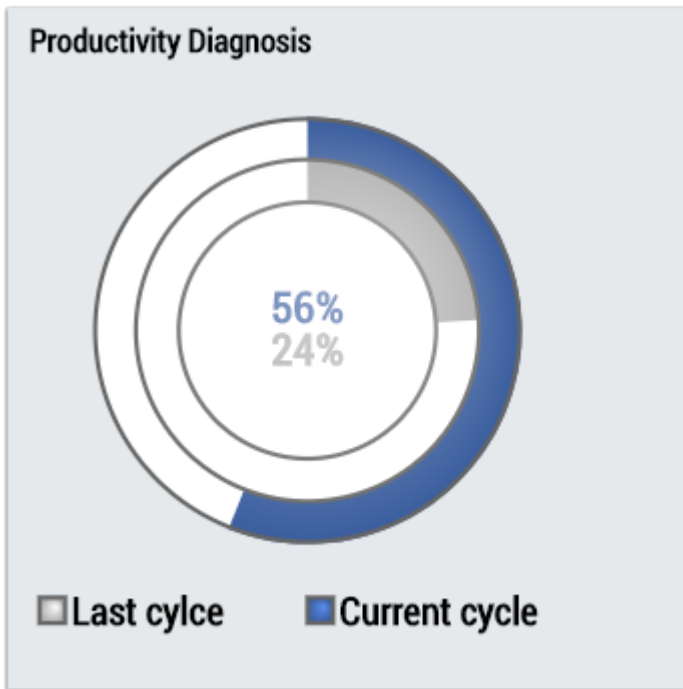
Ausgangswerte

- **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*.



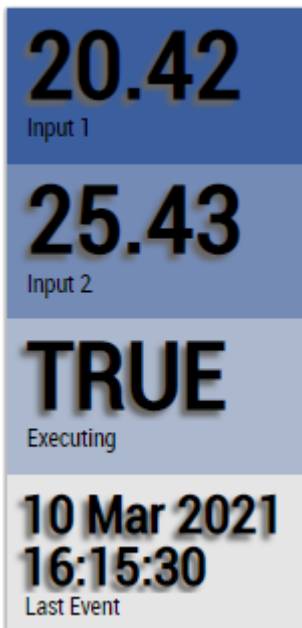
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: New Result, Producing, Cycle Finished, Productivity, Productivity Last Cycle.

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 kundenspezifische HMI Controls gemappt werden.

6.5.1.19 Productivity Interval 1Ch

Productivity Interval 1Ch			
%	Input	<Empty>	-
	Time Start	<Empty>	-
	Time Stop	<Empty>	-
	Expected Pieces	100	
	Threshold Edge	🔥 1	
	Current Product...	Empty	
	Current Timesta...	Empty	
	Elapsed Time	Empty	
	Expected Produ...	Empty	
	Interval Length	Empty	
	Last Full Period...	Empty	
	Produced In Int...	Empty	
	Remaining In In...	Empty	
	Remaining Time	Empty	

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.

Ausgangswerte

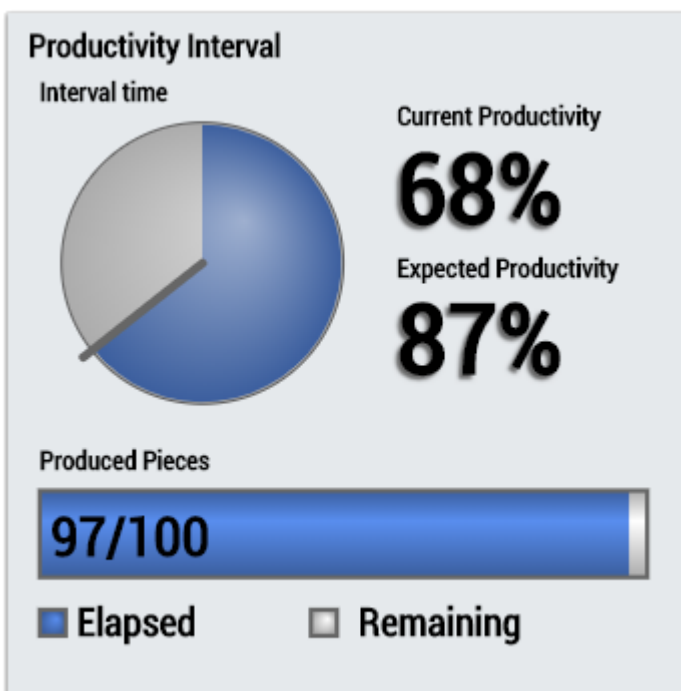
- **Within Interval:** Gibt an, ob sich die aktuelle Uhrzeit innerhalb des Intervalls befindet.
- **Current Timestamp:** Aktueller Zeitstempel.
- **Interval Length:** Länge des Intervalls.
- **Elapsed Time:** Verstrichene Zeit innerhalb des Intervalls.
- **Remaining Time:** Verbleibende Zeit innerhalb 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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.1.20 Signal Generator 1Ch

Signal Generator 1Ch		Signal <i>Empty</i>
	Timestamp	Amplitude <input type="text" value="1"/>
		Frequency <input type="text" value="50"/>
		Function Type <input type="text" value="Sine"/>
		Offset <input type="text" value="0"/>
		Sample Rate <input type="text" value="1000"/>

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

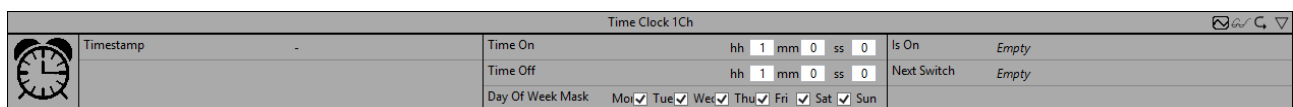
Für den Signal Generator 1Ch sind die folgenden HMI Controls für die Generierung eines Analytics Dashboards verfügbar:

1. Das SingleValue Control visualisiert den Ausgangswert Signal.



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

6.5.1.21 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.

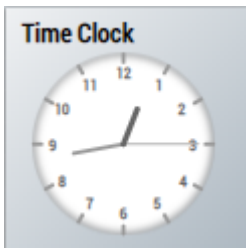
Ausgangswerte

- **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.



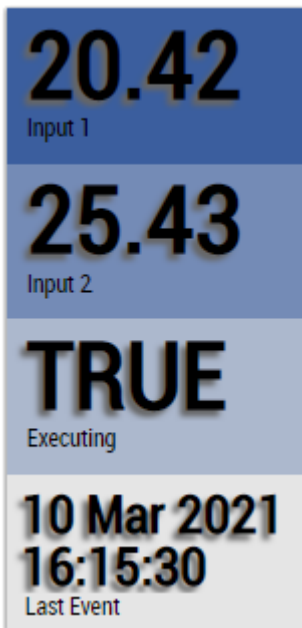
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Output, Next Switch.

Data Table Vertical

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

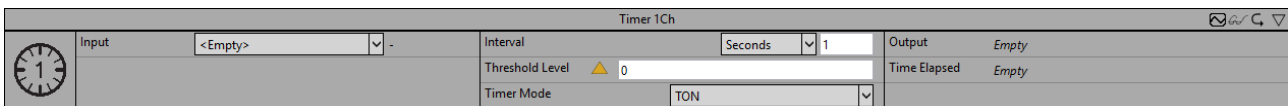
DataTable Horizontal

	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 Clock 1Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.1.22 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.

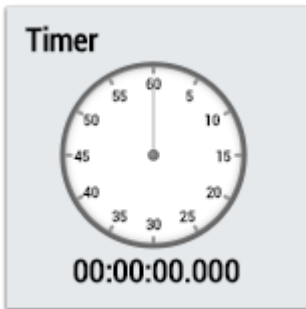
Ausgangswerte

- **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.



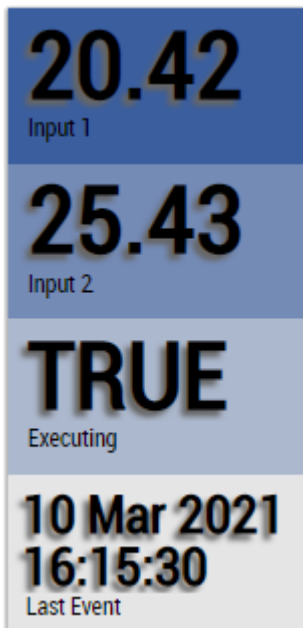
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Output, Elapsed Time.

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 kundenspezifische HMI Controls gemappt werden.

6.5.1.23 Timing Analysis 1Ch

Timing Analysis 1Ch			
	Input	<Empty>	Threshold LevelOn 1
			Count On Empty
			Is On Empty
			On Total Empty
			Off Total Empty
			Current Interval Empty

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.

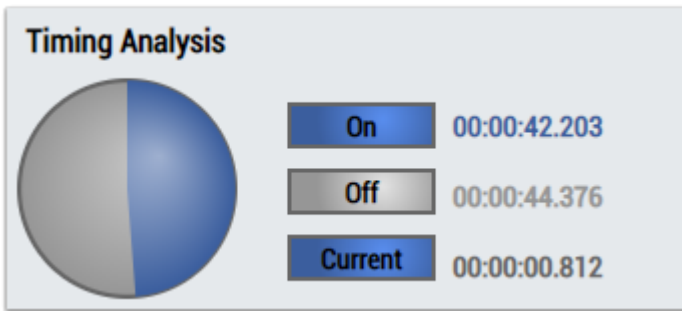
Ausgangswerte

- **Is On:** Zeigt innerhalb der Zeitspanne der On-Periode *TRUE* an, ansonsten *FALSE*.
- **Current Interval:** Zeigt die Zeit des aktuellen Intervalls an.
- **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.

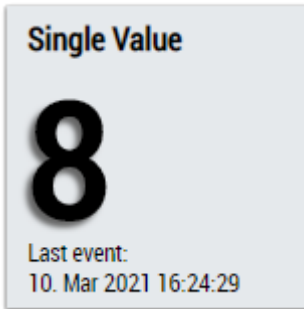
Standard HMI Controls

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.



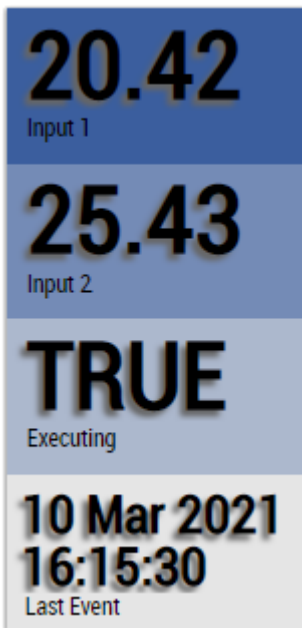
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Is On, Count On, Current Interval, On Total, Off Total.

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 kundenspezifische HMI Controls gemappt werden.

6.5.1.24 Latching Switch 1Ch

Latching Switch 1Ch			
	Input	<Empty>	-
	Threshold Edge	▲	1
	Out	Empty	
	Count	Empty	
	Last Event	Empty	

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.

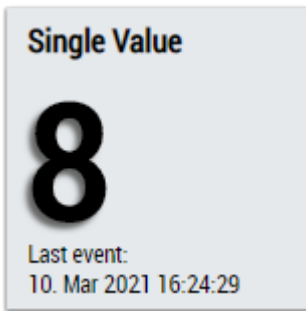
Ausgangswerte

- **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.

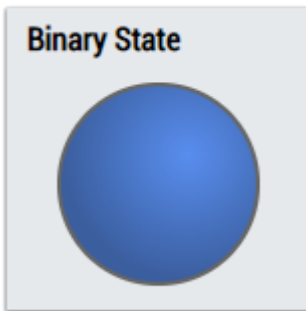
Standard HMI Controls

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*.



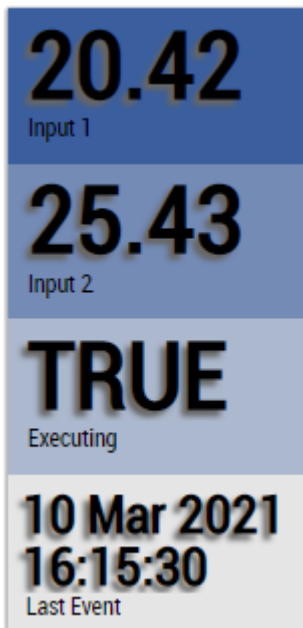
3. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte.

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 kundenspezifische HMI Controls gemappt werden.

6.5.2 Analytics - Classification

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

6.5.2.1 Bandwidth Classifier 1Ch

Bandwidth Classifier 1Ch			
Input	<Empty>	Lower Bound	0
		Upper Bound	10
		Class	Empty
		Last Event	Empty

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.

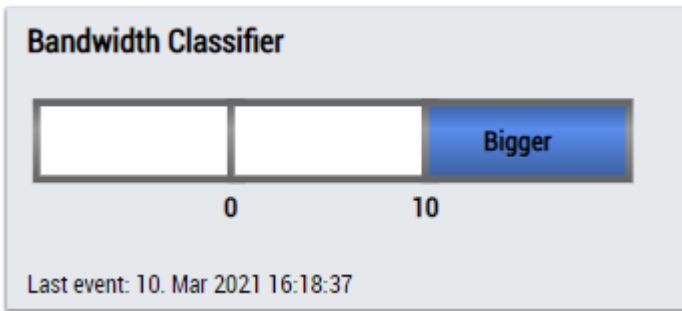
Ausgangswerte

- **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.

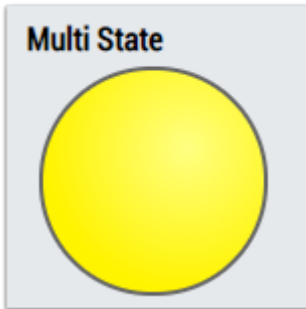
Standard HMI Controls

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.



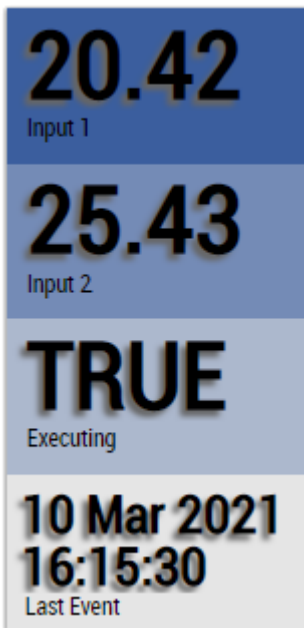
3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Class, Last Event.

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 1Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.2.2 Bandwidth Classifier 3 Ch

Bandwidth Classifier 3Ch			
Input	<Empty>	-	Class Empty
Lower Bound	<Empty>	-	Last Event Empty
Upper Bound	<Empty>	-	

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.

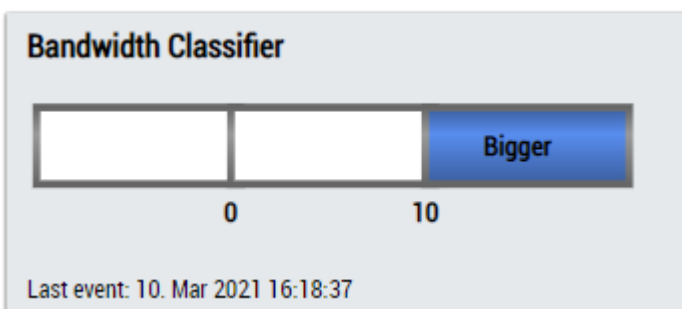
Ausgangswerte

- **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.

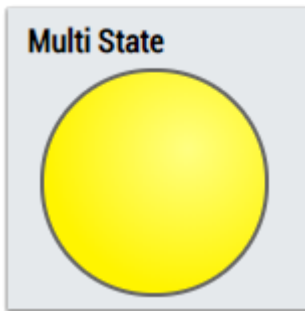
Standard HMI Controls

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.



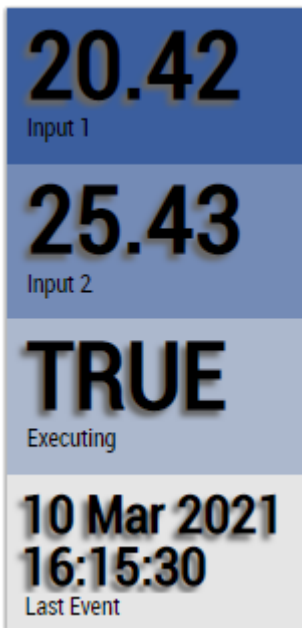
3. Das Table Control visualisiert alle Ausgangswerte: Class, Last Event.

Data Table Vertical

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

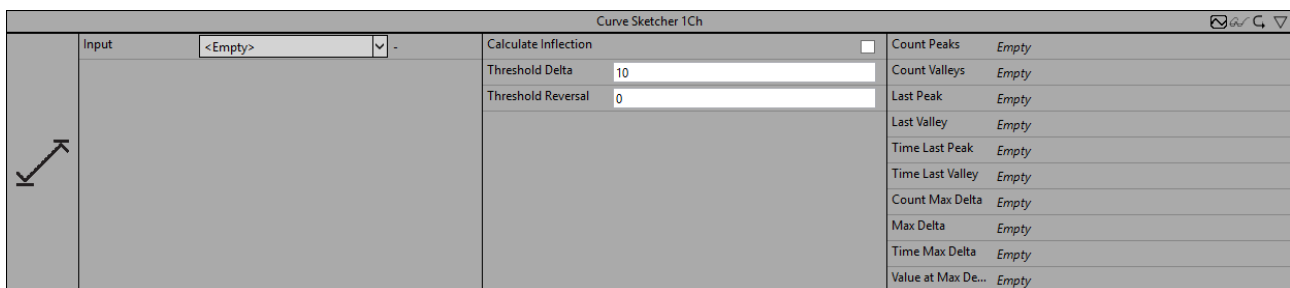
DataTable Horizontal

	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 kundenspezifische HMI Controls gemappt werden.

6.5.2.3 Curve Sketcher 1Ch

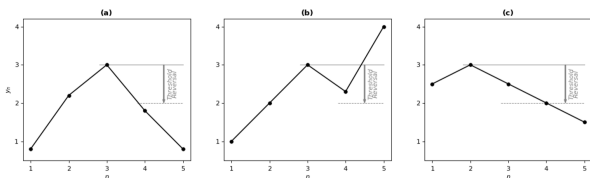


Curve Sketcher 1Ch identifiziert Umkehrungen (*Peaks* und *Täler*) in einem Eingangsdatenstrom. Des Weiteren können lokale Maxima der absoluten Differenz zwischen zwei aufeinanderfolgenden Werten (bezeichnet als *Delta*) identifiziert werden. Analog zu einer kontinuierlichen Kurve entsprechen die identifizierten 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.

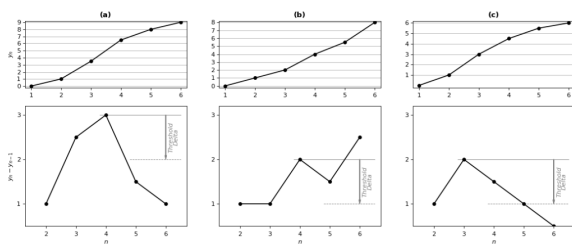
Optional kann ein boolesches Signal für den Eingang *Enable Execution* ausgewählt werden, sodass der Algorithmus nur aktiv ist, wenn der Wert des ausgewählten Signals *TRUE* ist.

Konfigurationsoptionen

- **Calculate Inflection:** Boolesches 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.
 - Der Wert y_4 wird nach der Verarbeitung des Werts y_5 als Maximum identifiziert, da die Differenz der zwei Deltas *Threshold Delta* überschreitet.
 - Es wird kein Maximum identifiziert, da die Differenz zwischen den Deltas kleiner ist als *Threshold Delta*.
 - 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.

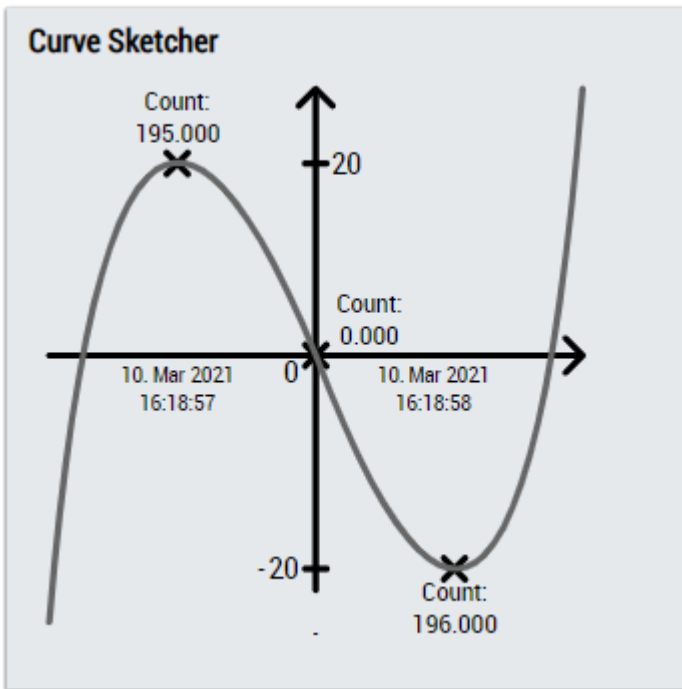
Ausgangswerte

- **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.



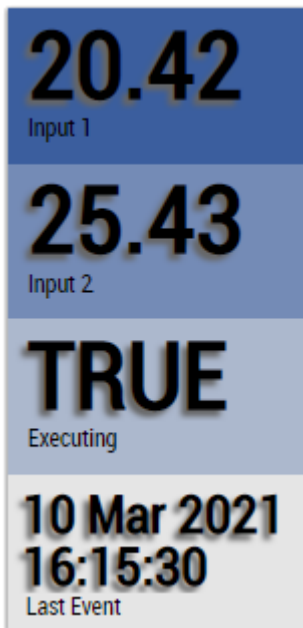
2. Das Table Control visualisiert alle Ausgangswerte: Last Peak, Time Last Peak, Count Peaks, Last Valley, Time Last Valley, Count Valley, Value at Max Delta, Last Delta, Time Last Delta und Count Delta.

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 kundenspezifische HMI Controls gemappt werden.

6.5.2.4 Histogram 1Ch

Histogram 1Ch			
Input	<Empty>	Histogram Mode	Absolute
		Min Binned	1
		Max Binned	10
		Bins	9
		Num Values	Empty
		Histogram	

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 Histogramm 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.

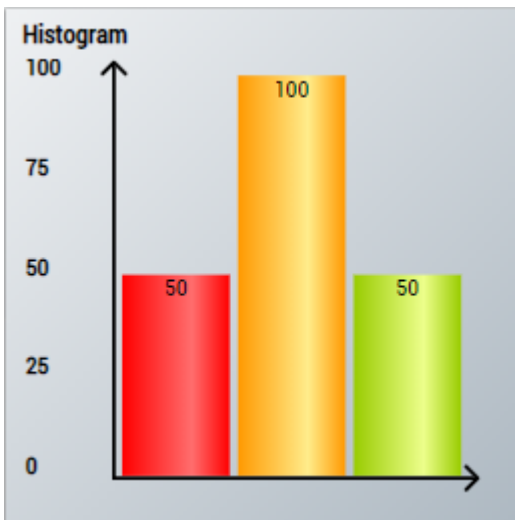
Ausgangswerte

- **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.

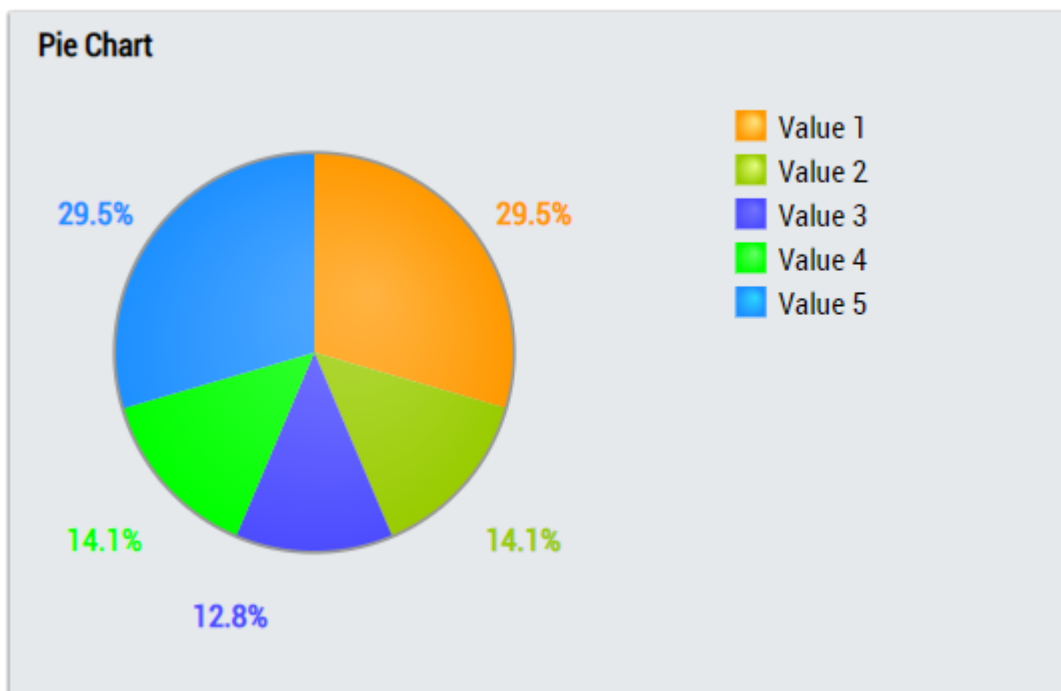
Standard HMI Controls

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.2.5 Section Timer 1 Ch

Section Timer 1Ch			
	Input <input type="text" value="<Empty>"/>	Num Sections <input type="text" value="1"/> + -	Section <input type="text" value="Empty"/>
		First Lower Border <input type="text" value="1"/>	Sections <input type="text" value="Empty"/>
		Upper Border 00 <input type="text" value="2"/>	
		Upper Border 01 <input type="text" value="3"/>	
		Upper Border 02 <input type="text" value="4"/>	

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 n0:** 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.

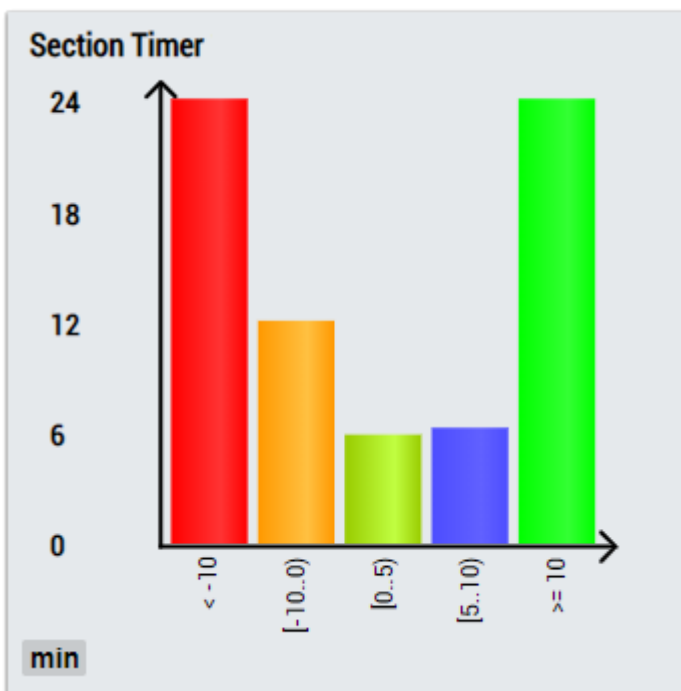
Ausgangswerte

- **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*, *Upper Border 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.

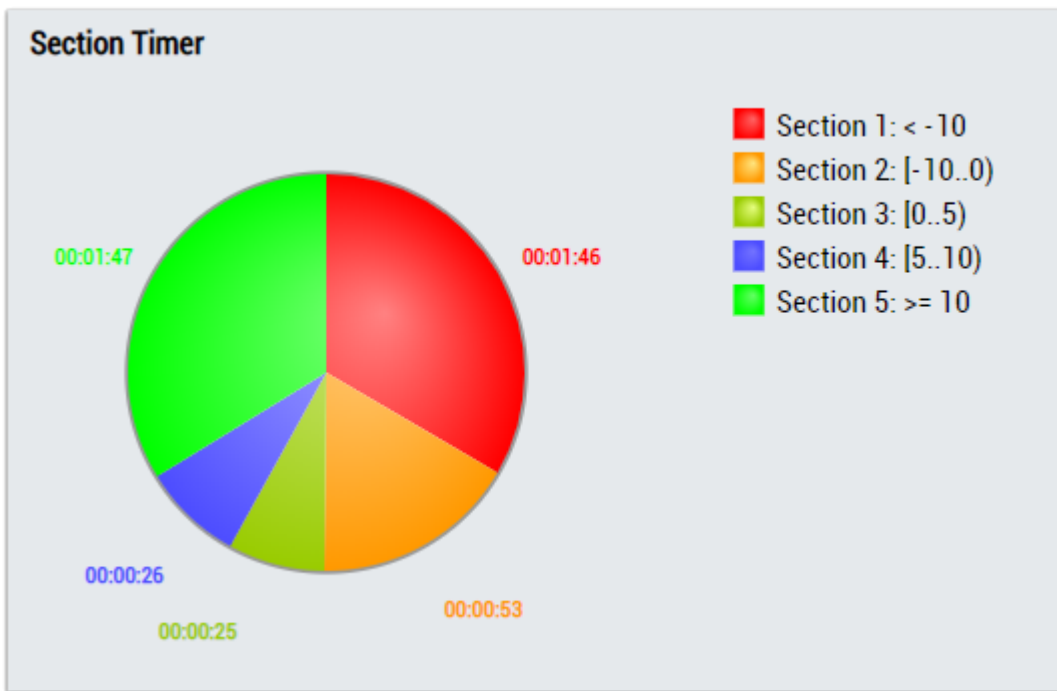
Standard HMI Controls

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*.

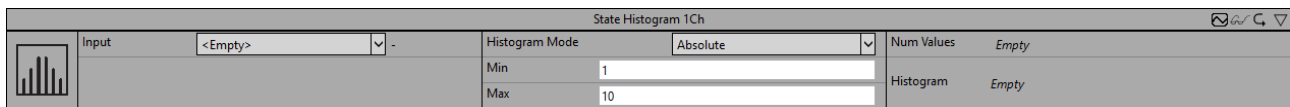


2. Das *PieChart Control* visualisiert den Eingangswert *Array of Timespans* und die Konfigurationsoptionen *First Lower Border* und *Upper Border*.



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

6.5.2.6 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.

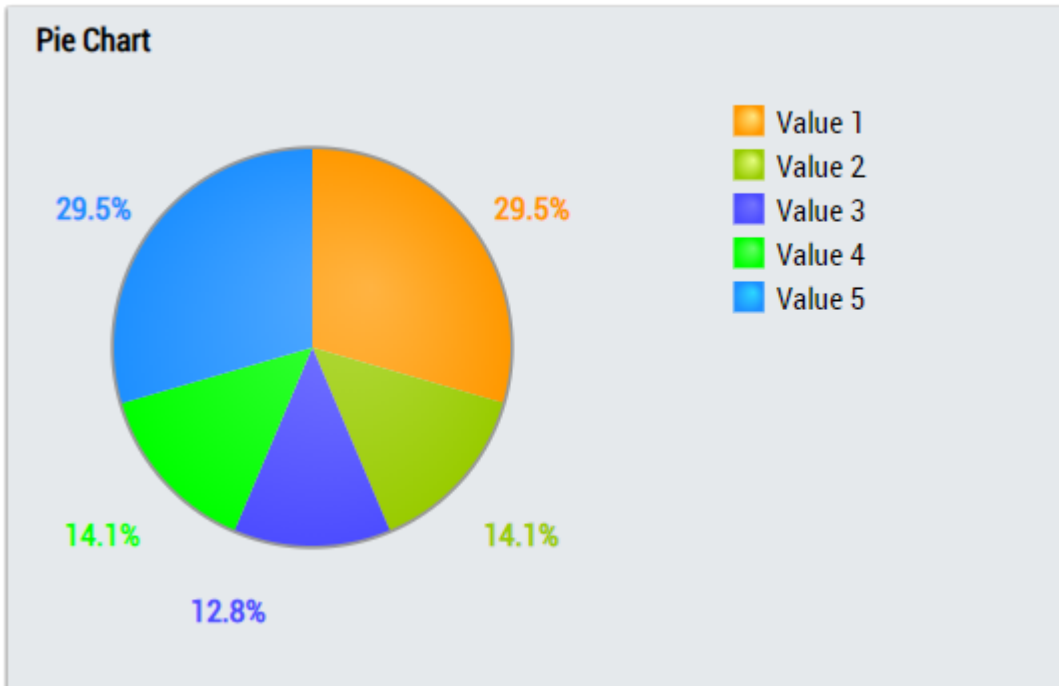
Ausgangswerte

- **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.

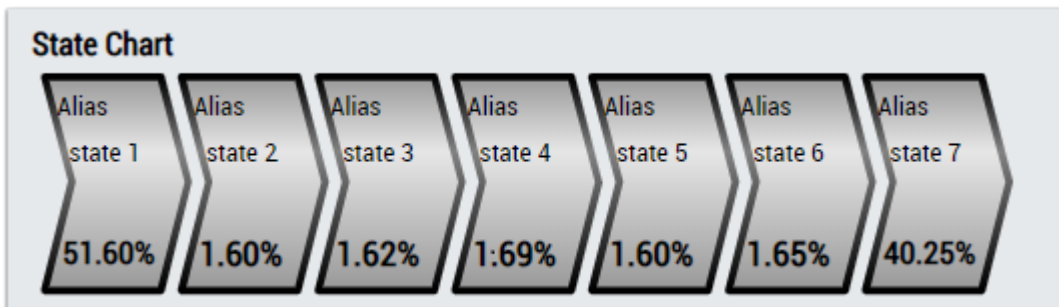
Standard HMI Controls

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

20.42

Input 1

25.43

Input 2

TRUE

Executing


10 Mar 2021

16:15:30

Last Event

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

6.5.2.7 Threshold Classifier 1Ch

Threshold Classifier 1Ch			
	Input	<Empty>	
	Level OK / Warning	10	Class <i>Empty</i>
	Level Warning / Alarm	20	Last Event Alarm <i>Empty</i>
			Last Event Warn... <i>Empty</i>

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.

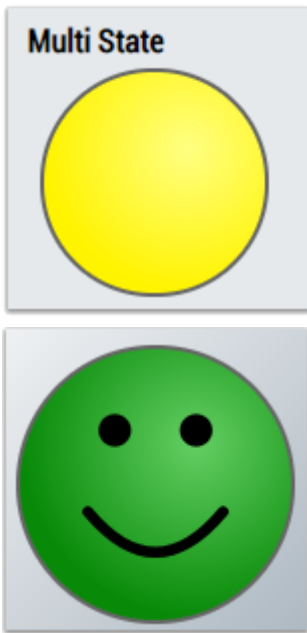
Standard HMI Controls

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

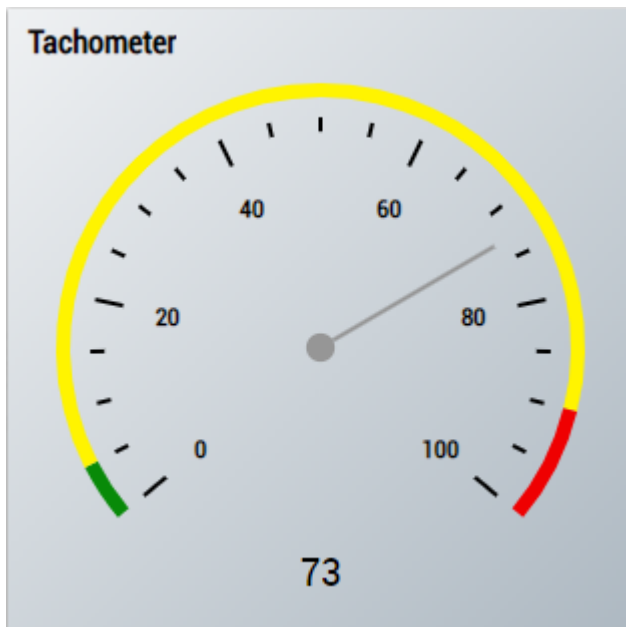
1. Das *TrafficLight Control* visualisiert die Ausgangswerte *Class*, *Last Event Warning* und *Last Event Alarm* sowie den Eingangswert der Daten.



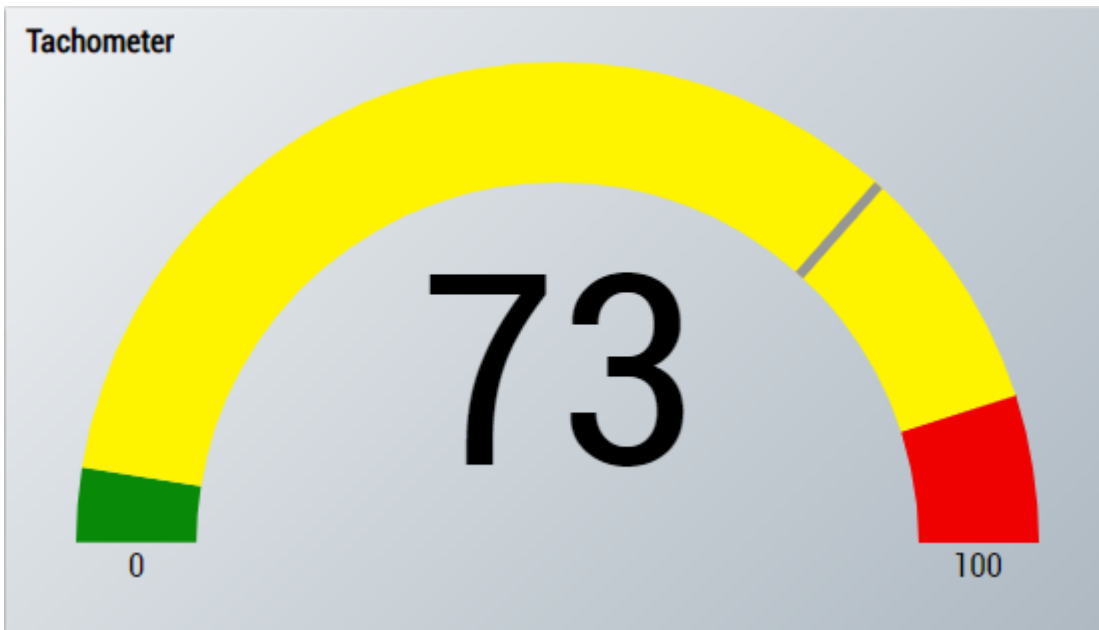
2. Das *MultiState Control* visualisiert den Ausgangswert *Class*, optional mit einem Smiley.



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



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



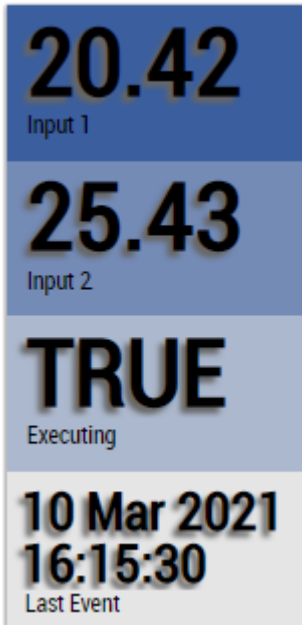
5. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Class, Last Event Warning Last Event Alarm.

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 kundenspezifische HMI Controls gemappt werden.

6.5.2.8 Threshold String Classifier 1Ch

Threshold String Classifier 1Ch			
	Input	<Empty>	
	String Level 1	String 1	Last Event
	Level 1 / 2	10	Output String
	String Level 2	String 2	Empty
	Level 2 / 3	20	
String Level 3	String 3		

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.

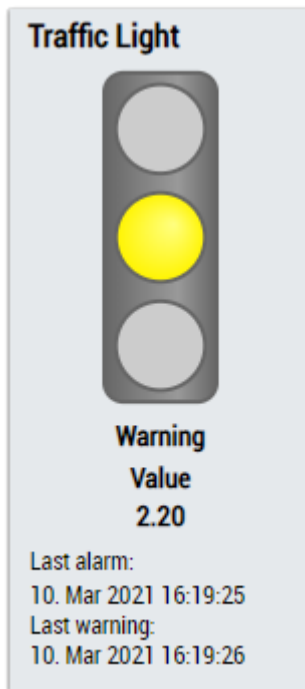
Ausgangswerte

- **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:

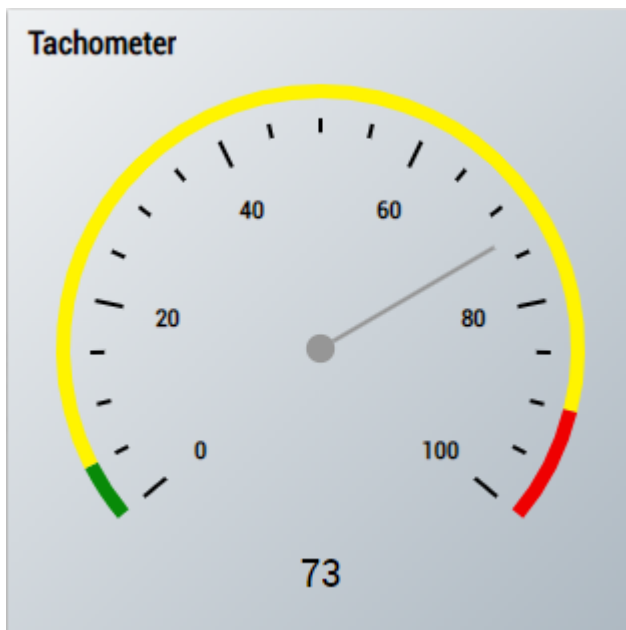
1. Das TrafficLight Control visualisiert die Ausgangswerte Output String und Last Event sowie den Eingangswert der Daten.



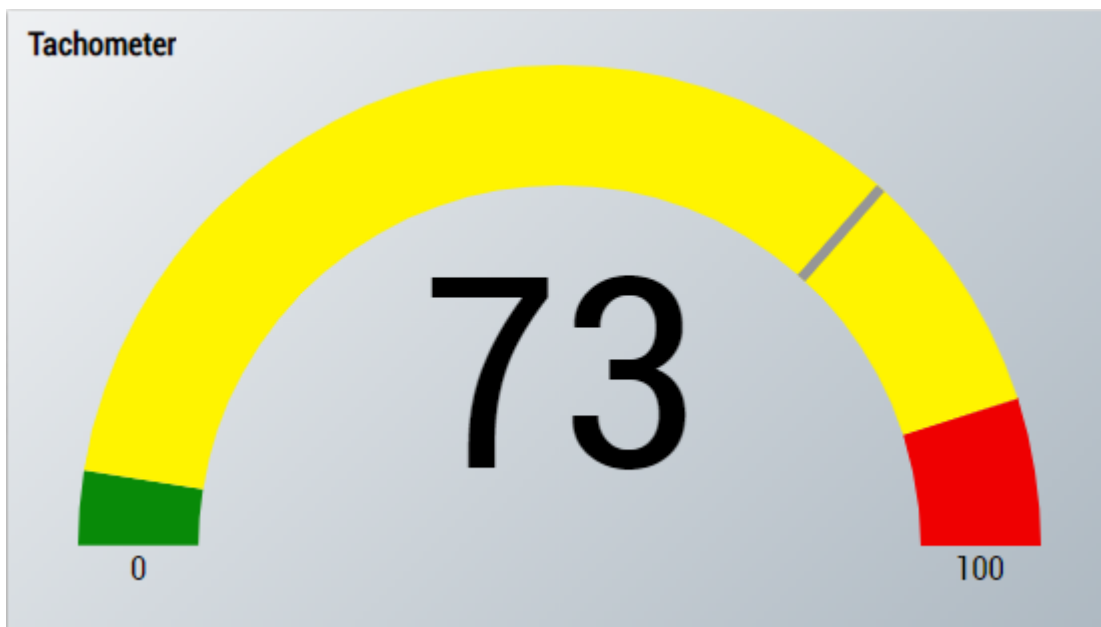
2. Das SingleValue Control visualisiert den Ausgangswert Output String.



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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

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

6.5.2.9 Time Based Envelope 1Ch

Time Based Envelope 1Ch			
Input	<Empty>	Band	10
Start Period	<Empty>	Band Mode	Absolute
		File Path	C:\TwinCAT\3.1\Boot\Teach.tas
		Compare Result	Empty
		Count Bigger	Empty
		Count Smaller	Empty
		Count Within B...	Empty
		Current Compa...	Empty
		Executing Com...	Empty
		Lower Band	Empty
		Upper Band	Empty
		Value Number	Empty
		Value Read	Empty
		Within Band	Empty

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 \[► 191\]](#) 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 \[► 191\]](#) zu verwenden. Stattdessen sollte zunächst ein Referenzsignal mit dem [Time Based Teach Path 1Ch \[► 191\]](#) eingelesen 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 eingelesenen Datendatei.
- **Band Mode:** Modus des Bandbetriebs (absolute oder relative Werte verwenden).
- **Band:** Bandbreite des Bandbetriebs.

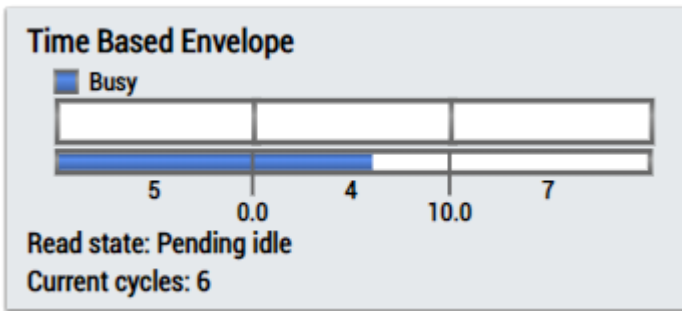
Ausgangswerte

- **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*.



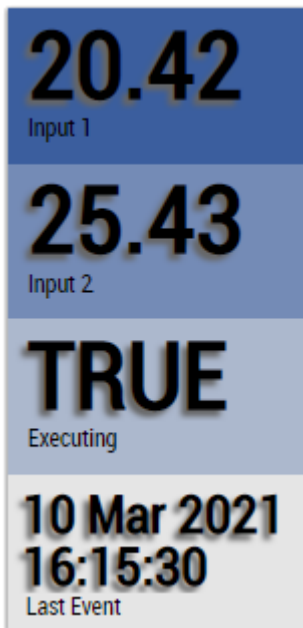
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Executing Comparison, State, Lower Band, Upper Band, Within Band, Compare Result, Current Compared Cycles, Count Within Band, Count Smaller, Count Bigger, Value Number.

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

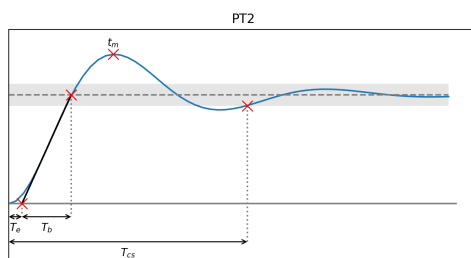


Alternativ können im Algorithmus Time Based Envelope mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.2.10 Step Response 1Ch

Step Response 1Ch						
Input	<Empty>	-	Threshold Reversal	0.01	Executing	Empty
Setpoint	<Empty>	-	Relative Tolerance	<input checked="" type="checkbox"/>	Equivalent Dea...	Empty
			Error Tolerance	5	Equivalent Tim...	Empty
					Settling Time	Empty
					Time Max	Empty
					Error	Empty

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.

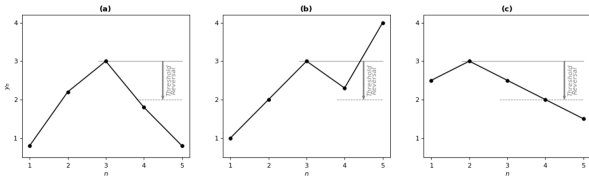
Konfigurationsoptionen

- 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.
 - 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:** Boolesches 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 Schnittpunkt der Wendetangenten mit dem Startwert.
- **Equivalent Time Constant:** Ausgleichszeit der Strecke. Zeitspanne zwischen dem Schnittpunkt der Wendetangenten mit dem Startwert und dem Schnittpunkt der Wendetangenten mit dem Sollwert.
- **Settling Time:** Ausregelzeit der Strecke.
- **Time Max:** Zeitpunkt des maximalen Überschingers.
- **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.

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

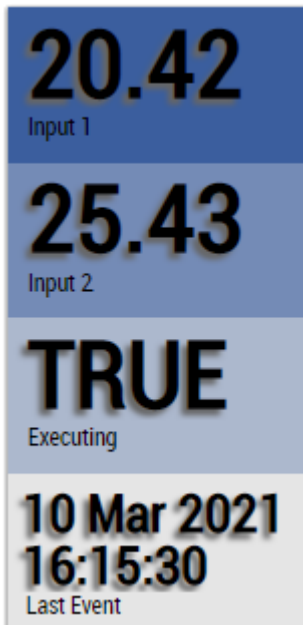


Abb. 2:

Alternativ können mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.3 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*.

6.5.3.1 DenStream

DenStream						
Update Micro Cl...	<Empty>	-	Num Channels	+ -	Number of Clus...	Empty
Input 00	<Empty>	-	Epsilon	0,15	Number of out...	Empty
			Epsilon (DBSCAN)	0,8	Number of pot...	Empty
			Lambda	1E-06	Cluster Index	Empty
			Mu x Beta	2	Last Event	Empty
			outMCs Buffer Size	100	Last Switch	Empty
			potMCs Buffer Size	100	MC Buffer Overf...	Empty
			Min Weight (DBSCAN)	2		

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 dichte-basierten 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. Farblich 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 $Beta \times Mu$ 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 $Beta \times Mu = 1$ 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 $Mu \times Beta$ und $Lambda$), wird er aus dem Speicher gelöscht.

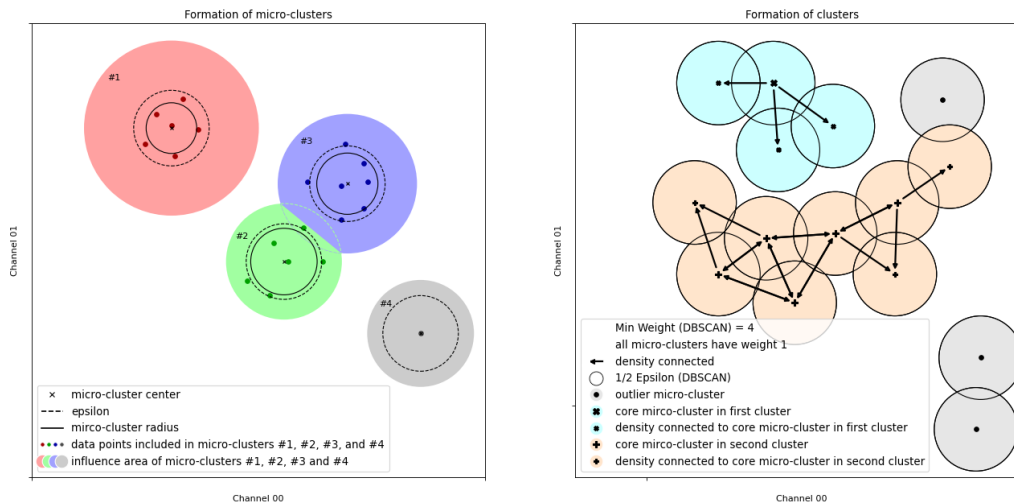
Die Parameter $Epsilon$ (DBSCAN) und $Min Weight$ (DBSCAN) betreffen die zweite Phase. Diese Parameter wurden vom DBSCAN-Algorithmus [3] übernommen.

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 größeren 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 wird 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 Knowledge 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 $2^{(-\text{Lambda})}$ 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.

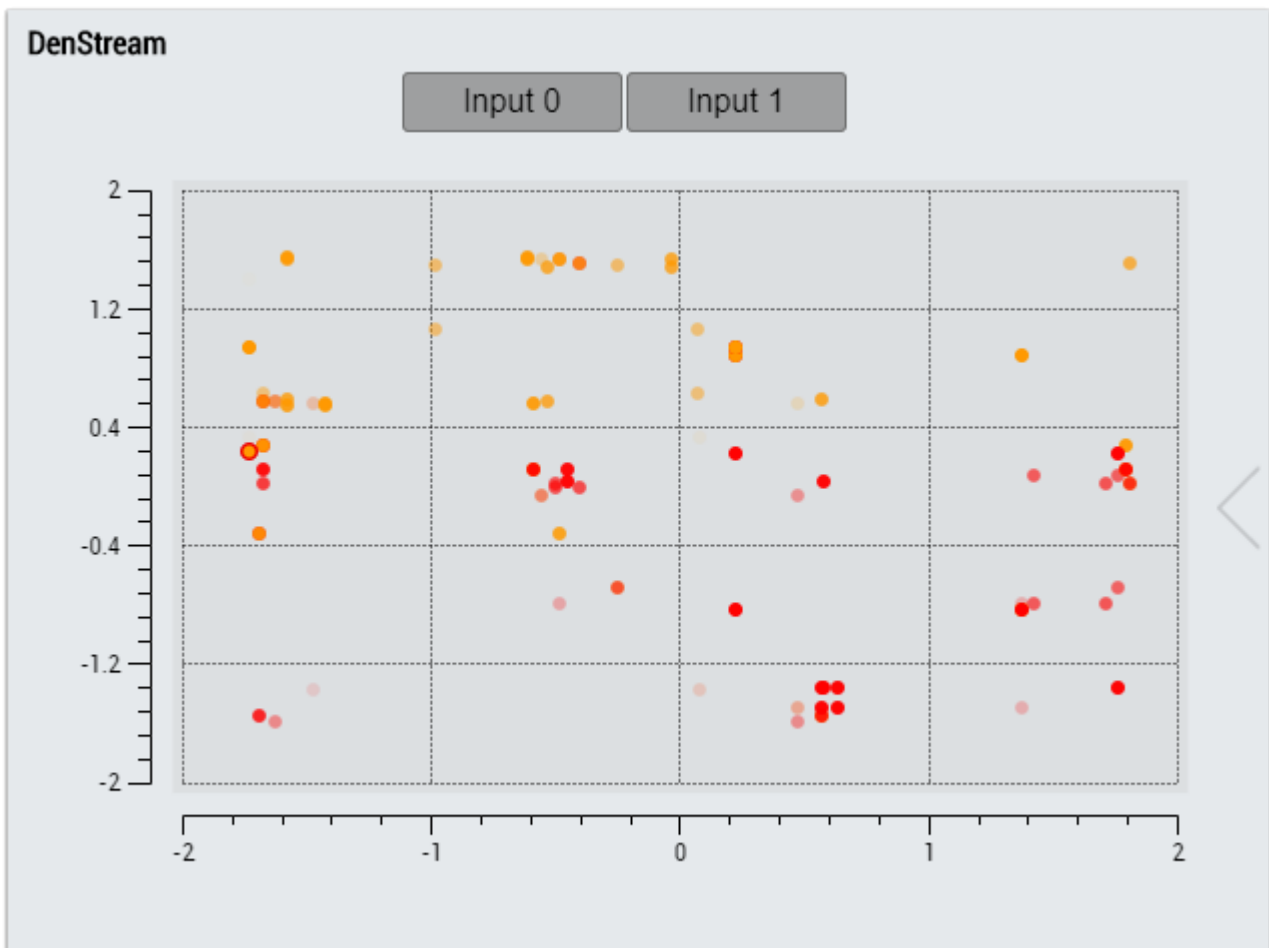
Ausgangswerte

- **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:

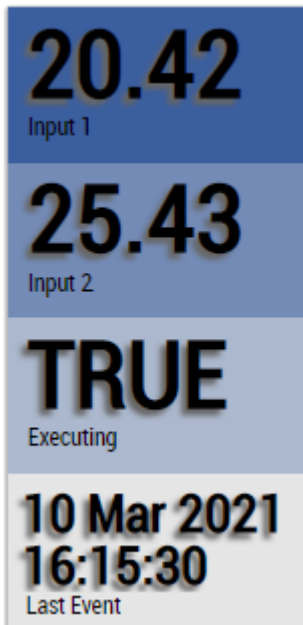
1. Das DenStream Control visualisiert im Chart die Inputs und farblich deren jeweilige Klassifikation zu einem Cluster (Cluster-Index). Über die Buttons können die anzuzeigenden Inputs ausgewählt werden, maximal zwei gleichzeitig sind möglich. Der Slider auf der rechten Seite führt alle Ausgangswerte 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 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 DenStream mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.3.2 Sequential k-Means

Sequential k-Means			
Update Cluster...	<Empty>	-	Num Channels <input type="text" value="1"/> + -
Input 00	<Empty>	-	Number of Clusters <input type="text" value="1"/>
			Aggregation Buffer Size <input type="text" value="10"/>
			Initial Cluster Centers <input type="text" value="1"/>
			Initialization Mode <input type="text" value="Equidistant"/>
			Max Iterations <input type="text" value="1"/>
			Lower Bounds 00 <input type="text" value="-1000"/>
			Upper Bounds 00 <input type="text" value="1000"/>
			Cluster Index <input type="text" value="Empty"/>
			Distance <input type="text" value="Empty"/>
			Cluster Center 00 <input type="text" value="Empty"/>

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 zwischenspeichernd. 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.
- **Equidistant:** 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.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30


Last Event

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

6.5.4 Analytics - Compare

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

6.5.4.1 Demultiplexer

Demultiplexer				
	Input	<Empty>	-	Num Channels
				+ -
				Count <i>Empty</i>
				Current Channel <i>Empty</i>
				Last Event <i>Empty</i>
				Out 00 <i>Empty</i>
				Out 01 <i>Empty</i>

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:

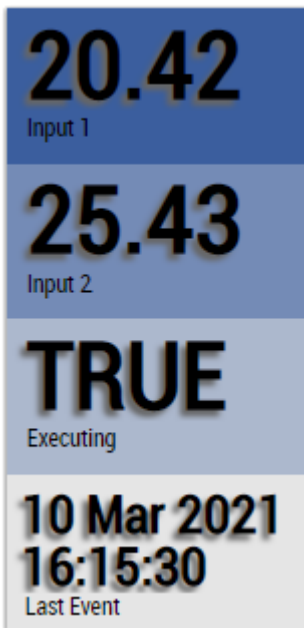
1. Das Table Control oder Multivalue Control visualisiert die Ausgangswerte Count, Current Channel und Last Event und Out.

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 kundenspezifische HMI Controls gemappt werden.

6.5.4.2 Detect String Change 1Ch

Detect String Change 1Ch			
a→A	Input	<Empty>	-
	Case Sensitive	<input checked="" type="checkbox"/>	Count
			Empty
			Last Event
			Empty
			String Changed
			Empty

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.

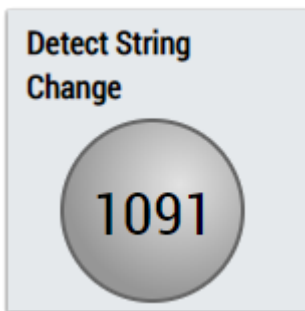
Ausgangswerte

- **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:

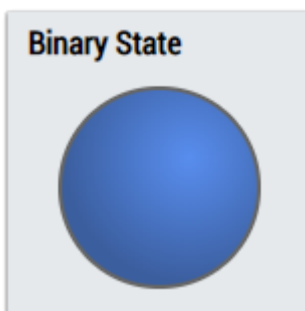
1. Das Comparison Control visualisiert die Ausgangswerte Boolean Switch und Count.



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.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.4.3 Dynamic Time Warping

Dynamic Time Warping					
	Input	<Empty>	-	Num Templates	+ - Executing C... Empty
	Start Period	<Empty>	-	File Path 01	ect1\TwinCAT Measurement Project1\Teach\Template1.t ...
	Stop Period	<Empty>	-	File Path 02	ect1\TwinCAT Measurement Project1\Teach\Template2.t ...
					Best Match Idx Empty
					Distance 01 Empty
					Distance 02 Empty

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 \[▶ 191\]](#) zu verwenden. Stattdessen sollte zunächst ein Referenzsignal mit dem [Time Based Teach Path 1Ch \[▶ 191\]](#) 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 \[▶ 191\]](#) 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 \[▶ 75\]](#) 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:

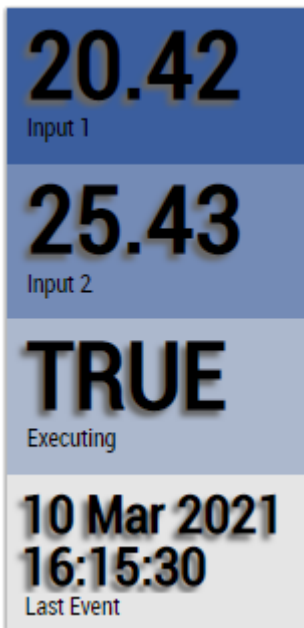
1. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte: *Executing Compare*, *Best Match Idx*, *Distance*.

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 *Dynamic Time Warping* mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.4.4 Dynamic Time Warping Interval

Dynamic Time Warping Interval					
	Reference Input	<Empty>	-	Num Channels	+ -
	Input 01	<Empty>	-	Window Size	100
	Input 02	<Empty>	-		
				Best Match Idx	Empty
				Distance 01	Empty
				Distance 02	Empty

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*.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.4.5 Logic Operation Counter

Logic Operation Counter			
&	Input 00	<Empty>	-
	Input 01	<Empty>	-
	Num Channels		+ -
	Count Mode		OnChange
	Logic Operator		≥1
Threshold 00		▲ 1	
Threshold 01		▲ 1	
Count		Empty	
Last Event		Empty	
Operation Out		Empty	

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:



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*.

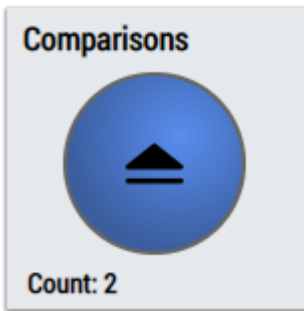
Ausgangswerte

- **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:

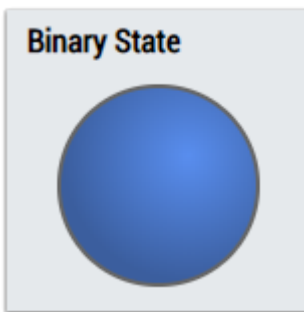
1. Das *Comparison Control* visualisiert die Ausgangswerte *Operation Out* und *Count* sowie die Konfigurationsoption *Logic Operator*.



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.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.4.6 Multiplexer

Multiplexer						
	Default Result	<Empty>	-	Num Channels + -	Count	Empty
	Condition 00	<Empty>	-		Current Channel	Empty
	Input 00	<Empty>	-		Last Event	Empty
					Result	Empty

Der Multiplexer wählt einen Kanal aus einem oder mehreren Eingangskanälen aus. Für jeden Eingangskanal muss zusätzlich ein boolescher 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:

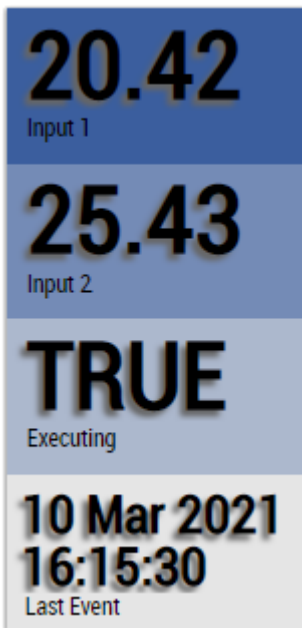
1. Das Table Control oder Multivalue Control visualisiert die Ausgangswerte Count, Result, Current Channel und Last Event.

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 Multiplexer mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.4.7 Numerical Compare 1Ch

Numerical Compare 1Ch			
< >	Input: <Empty>	Count Mode: OnChange	Count: Empty
	Reference: 0	Use Absolute Values: <input type="checkbox"/>	Last Event: Empty
			Operation Out: Empty

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*.

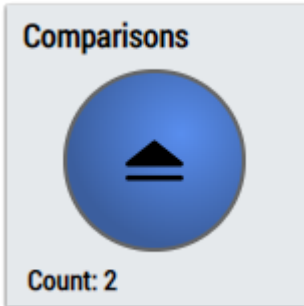
Ausgangswerte

- **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

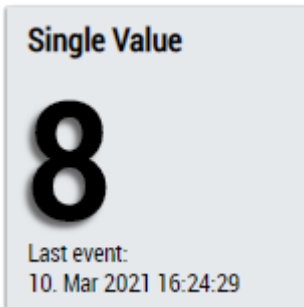
Standard HMI Controls

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

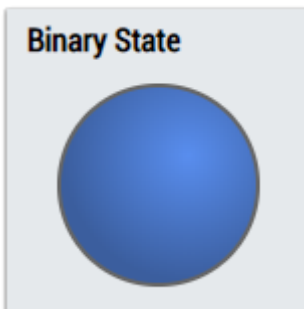
1. Das Comparison Control oder Multivalue Control visualisiert die Ausgangswerte Operation Out und Count sowie die Konfigurationsoption Operator.



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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.4.8 Numerical Compare 2Ch

Numerical Compare 2Ch					
< >	Input Ch1	<Empty>	-	Count Mode	OnChange
	Input Ch2	<Empty>	-	Operator (Ch1 Operator Ch2)	greaterThanOrEqualTo
				Use Absolute Values	<input type="checkbox"/>
				Count	Empty
				Last Event	Empty
				Operation Out	Empty

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

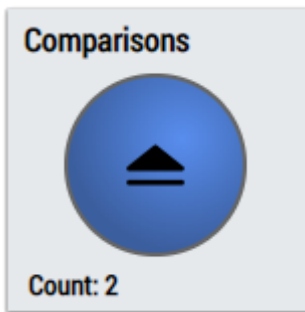
- **Operation Out:** Ergebnis der Vergleichsoperation.

- **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:

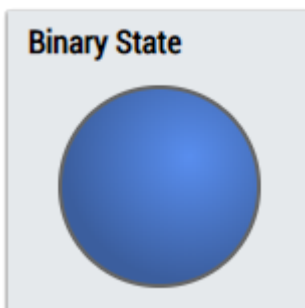
1. Das Comparison Control oder Multivalue Control visualisiert die Ausgangswerte Operation Out und Count sowie die Konfigurationsoption Operator.



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 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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.4.9 String Compare 1Ch

	Input <input type="text" value="<Empty>"/>	Case Sensitive <input checked="" type="checkbox"/>	Count <i>Empty</i>
a b c < > a c b	Count Mode <input type="text" value="OnChange"/>	Last Event <i>Empty</i>	String Match <i>Empty</i>
	Reference String <input type="text" value="EmptyString"/>	String Compare Mode <input type="text" value="Equals"/>	

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.

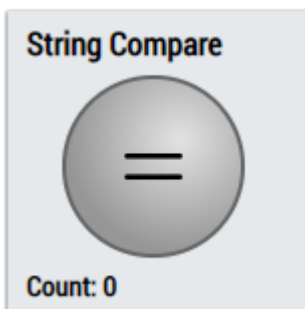
Ausgangswerte

- **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:

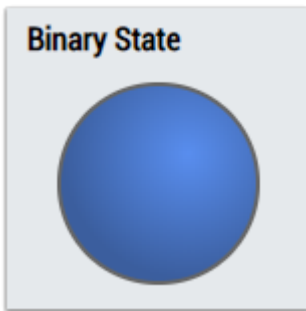
1. Das *Comparison Control* visualisiert die Ausgangswerte *String Match* und *Count* sowie die Konfigurationsoption *String Compare Mode*.



2. Das *Single Value Control* visualisiert die Ausgangswerte *Count* und *Last Event*.



3. Das *BinaryState Control* visualisiert den Ausgangswert *String Match*.



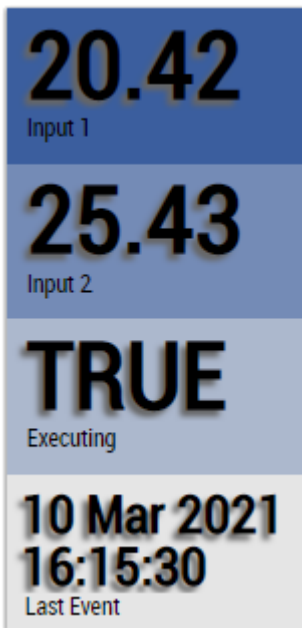
4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: String Match, Count, Last Event.

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 kundenspezifische HMI Controls gemappt werden.

6.5.4.10 String Compare 2Ch

String Compare 2Ch					
a b c	Input Ch1	<Empty>	-	Case Sensitive	<input checked="" type="checkbox"/>
< >	Input Ch2	<Empty>	-	Count Mode	OnChange
a c b				String Compare Mode	Equals
				Count	Empty
				Last Event	Empty
				String Match	Empty

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.

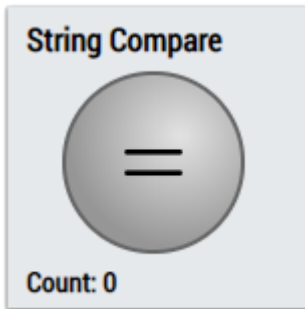
Ausgangswerte

- **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

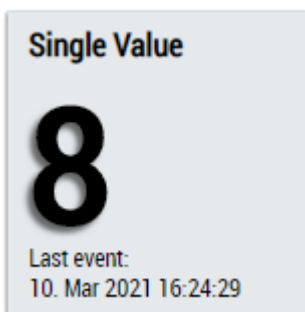
Standard HMI Controls

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

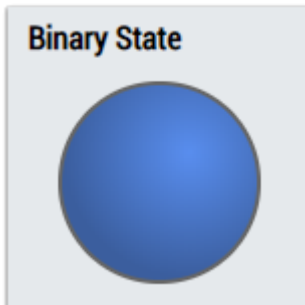
1. Das Comparison Control visualisiert die Ausgangswerte String Match und Count sowie die Konfigurationsoption String Compare Mode.



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.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

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

6.5.4.11 Detect Value Change 1Ch

Detect Value Change 1Ch			
<>	Input	Tolerance	Value Changed
	<Empty>	0	Empty
			Count
			Empty
			Last Event
			Empty

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, wird eine Wertänderung erkannt.

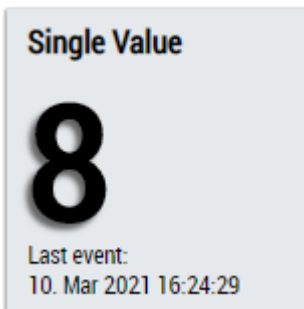
Ausgangswerte

- **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.

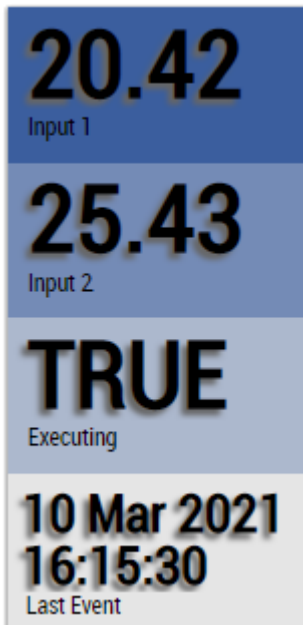


2. Das Table Control und Multivalue Control visualisieren alle Ausgangswerte.

The image shows a rectangular HMI control titled "Data Table Vertical". It contains a table with two columns. The first column lists labels, and the second column lists their corresponding values.

	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

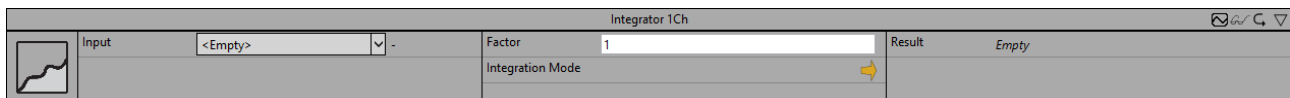


Alternativ können mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.5 Analytics - Math

Die Algorithmen der Kategorie *Analytics-Math* bieten Funktionen für mathematische Operationen, wie Grundrechenarten, Integration oder Steigungsanalyse.

6.5.5.1 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.

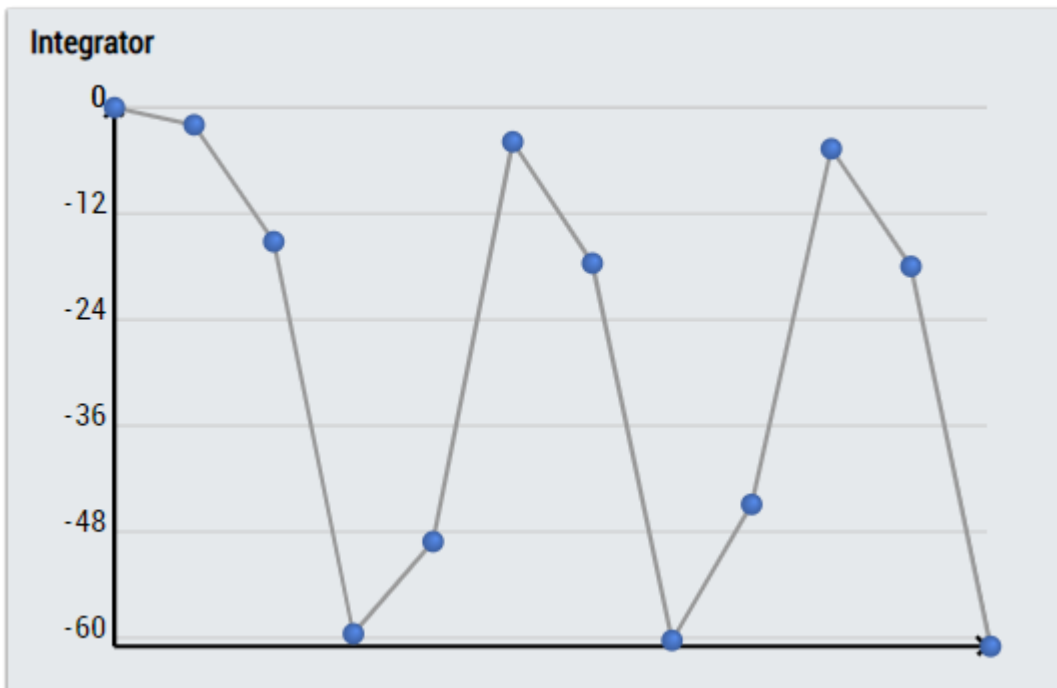
Ausgangswerte

- **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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

Alternativ können im Algorithmus Integrator 1Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.5.2 Math Operation

Math Operation			
+ -	Input 00	<Empty>	-
÷ ×	Input 01	<Empty>	-
	Num Channels	+ -	Result <i>Empty</i>
	Mathematical Operator	+	
	Use Absolute Values	<input type="checkbox"/>	

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 und 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

- **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.

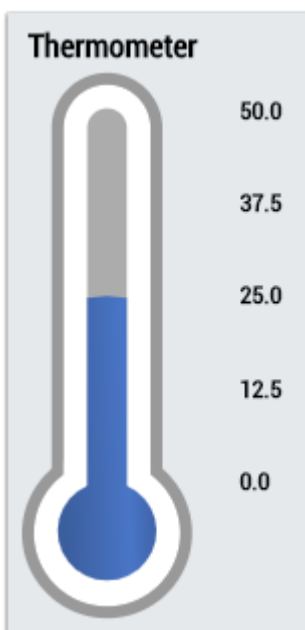
Standard HMI Controls

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*.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

Alternativ können im Algorithmus Math Operation mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.5.3 Math Operation 1Ch

Math Operation 1Ch			
+ - ÷ ×	Input	<input type="text" value="<Empty>"/>	Mathematical Operand <input type="text" value="0"/> Use Absolute Values <input type="checkbox"/>
		Result	Empty

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, wird der absolute Wert des Eingangssignals verwendet.

Ausgangswerte

- **Result:** Ergebnis der mathematischen Operation.

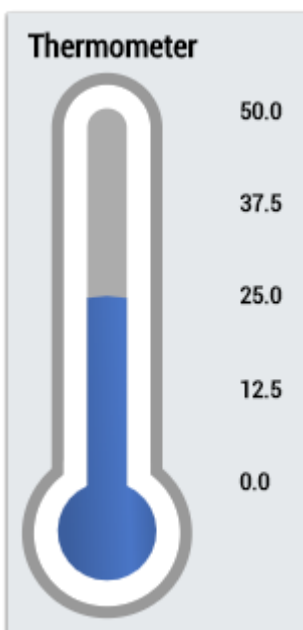
Standard HMI Controls

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.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

Alternativ können im Algorithmus Math Operation mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.5.4 RMS 1Ch

	RMS 1Ch	RMS 00	Empty
RMS	Input <input type="text" value="<Empty>"/>	Num Cascades <input type="text" value="1"/>	Cascades <input type="text" value="Cascade 00: 0:00:00:01.000"/>
		Sample Rate <input type="text" value="1000"/>	Startup Behaviour <input type="text" value="UsePreviousCascadeValue"/>

Der *RMS 1Ch* berechnet das quadratische Mittel (root mean square) über die Eingangswerte nach der Formel

$$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^N x[n]^2}$$

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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

Alternativ können im Algorithmus RMS mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.5.5 Slope Analysis 1Ch

Slope Analysis 1Ch			
Input	<Empty>	Num Values	2
		Delta Time	Empty
		Delta Value	Empty
		Slope	Empty
		Time Slope	Empty
		Slope Min	Empty
		Time Slope Min	Empty
		Slope Max	Empty
		Time Slope Max	Empty

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.

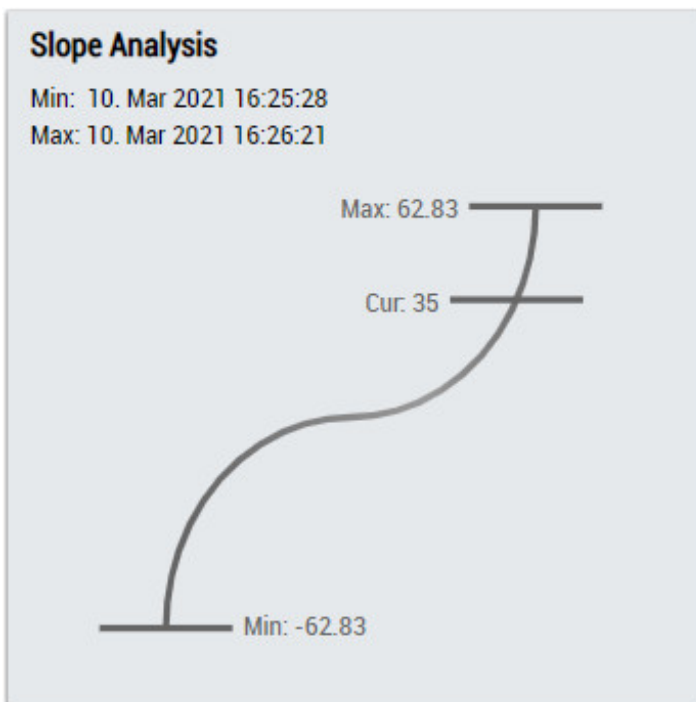
Ausgangswerte

- **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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

Alternativ können im Algorithmus mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.6 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.

6.5.6.1 Time Based Teach Path 1Ch

Time Based Teach Path 1Ch			
Input	<Empty>	Number Of Teaches	0
Start Period	<Empty>	File Path	C:\TwinCAT\3.1\Boot\Teach.tas
Stop Period	<Empty>	Involve Existing File	<input checked="" type="checkbox"/>
		Teach mode	Maximum
		Current Teach C...	Empty
		Executing Teach	Empty
		Teaching Done	Empty
		Values in File	Empty
		Written Values	Empty

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 (boolesche 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* [► 144] 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* [► 144] zu verwenden. Stattdessen sollte zunächst das Referenzsignal erlernt werden und erst im Anschluss eine Auswertung mit dem *Time Based Envelope 1Ch* [► 144] 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 boolesche Eingangssignal von *Start Period* negiert.
- **Negate Stop Period:** Wenn die Checkbox aktiviert ist, wird das boolesche Eingangssignal von *Stop Period* negiert.

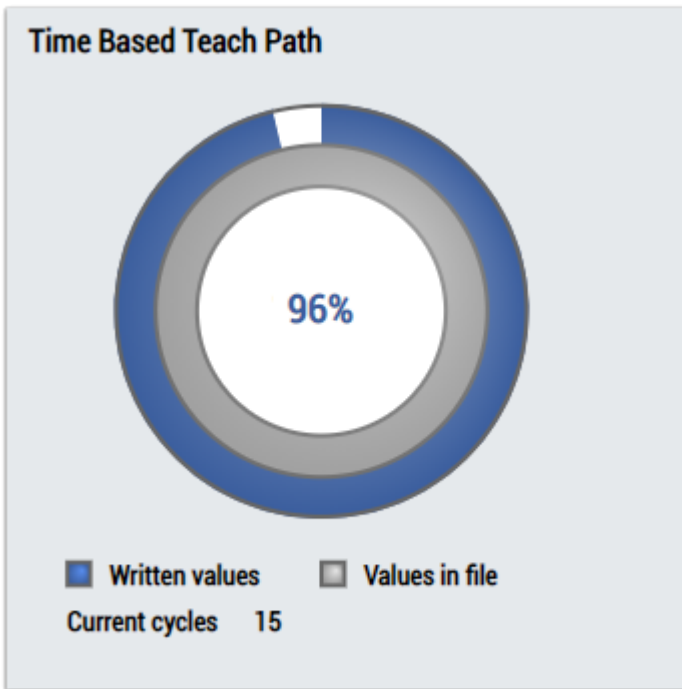
Ausgangswerte

- **Executing Teach:** Zeigt an, ob die Einlernung aktiv ist (Zeitbereich zwischen Start- und Stopflag).
- **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*.



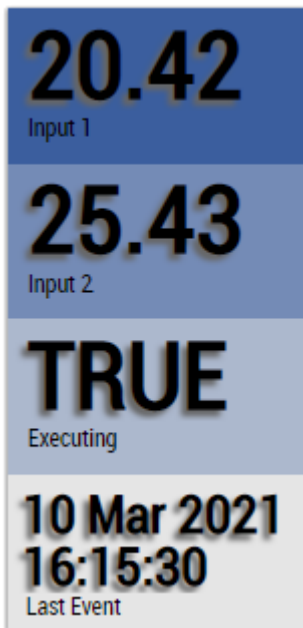
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Written Values, Values in File, Current Teach Cycles.

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 kundenspezifische HMI Controls gemappt werden.

6.5.7 Analytics - XTS

Die Algorithmen der Kategorie *Analytics-XTS* bieten spezielle Funktionen für das Beckhoff XTS-System, zum Beispiel die Analyse von Entfernung, Geschwindigkeit und Beschleunigung.

6.5.7.1 XTS Acceleration Analysis 1Ch

XTS Acceleration Analysis 1Ch		
Position [mm]	<Empty>	XTS Length [mm] 4000
		Acceleration [m/s²] Empty

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.

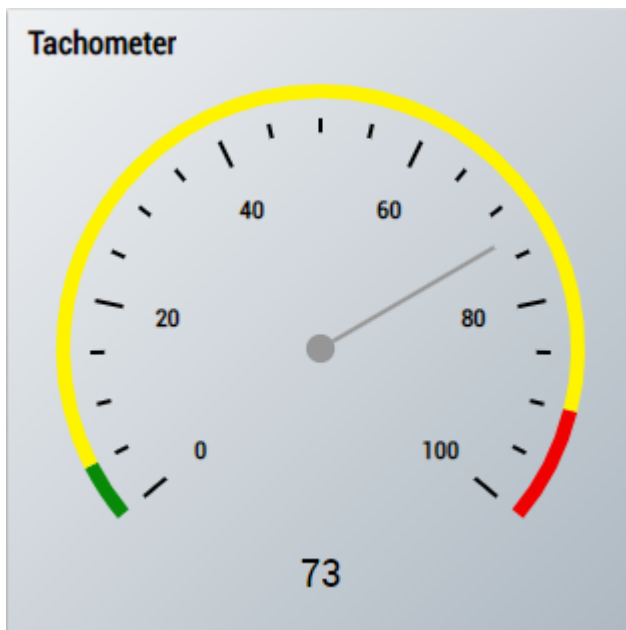
Ausgangswerte

- **Acceleration:** Aktuelle Beschleunigung des XTS-Movers.

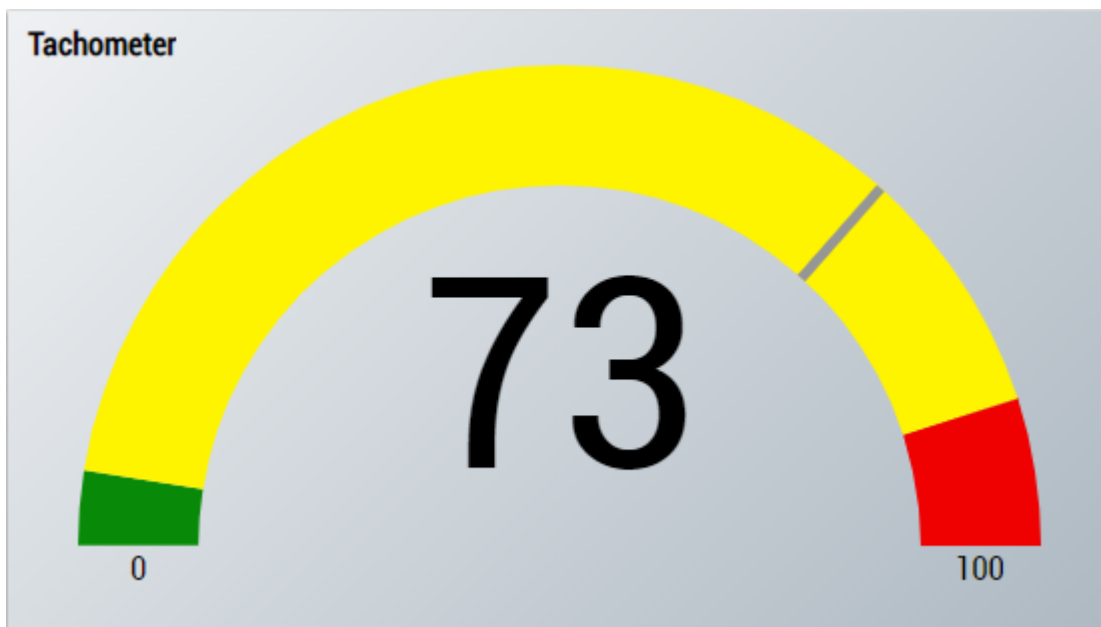
Standard HMI Controls

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

Alternativ können im Algorithmus XTS Acceleration Analysis 1Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.7.2 XTS Distance Integrator 1Ch

XTS Distance Integrator 1Ch				
	Position [mm]	<Empty>	XTS Length [mm]	4000
			Distance [m]	Empty
			Distance Negati...	Empty
			Distance Positiv...	Empty

Der *XTS Distance Integrator 1Ch* berechnet die zurückgelegte Entfernung eines XTS-Movers. Der Algorithmus liefert die Gesamtdistanz, 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.

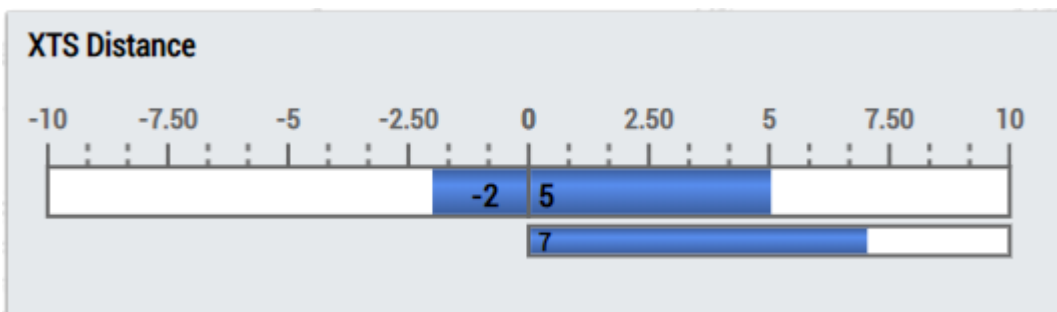
Ausgangswerte

- **Distance**: Gesamtdistanz, 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).

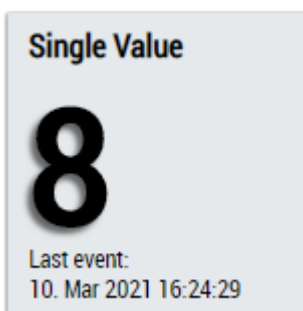
Standard HMI Controls

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: *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

20.42

Input 1

25.43

Input 2

TRUE


Executing

10 Mar 2021 16:15:30

Last Event

Alternativ können im Algorithmus XTS Distance Integrator 1Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.7.3 XTS Velocity Analysis 1Ch

XTS Velocity Analysis 1Ch			
	Position [mm] <input style="width: 80%;" type="text" value=" <Empty> "/>	XTS Length [mm] <input style="width: 90%;" type="text" value=" 4000 "/>	Velocity [m/s] <i>Empty</i>

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.

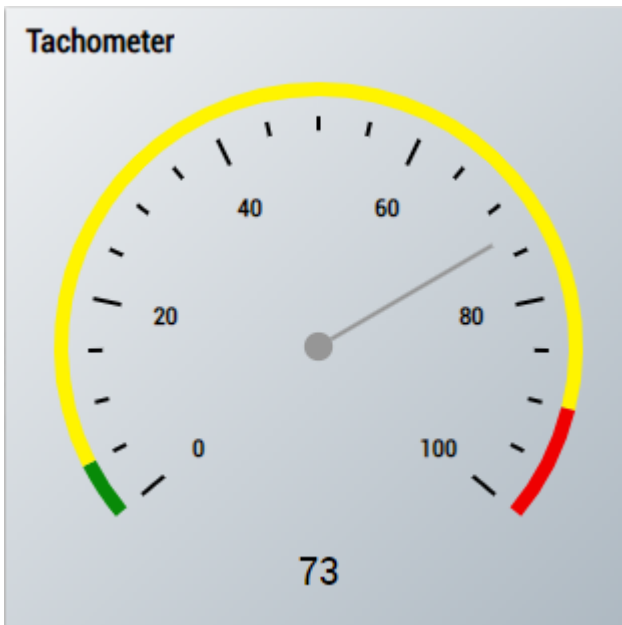
Ausgangswerte

- **Velocity**: Aktuelle Geschwindigkeit des XTS-Movers.

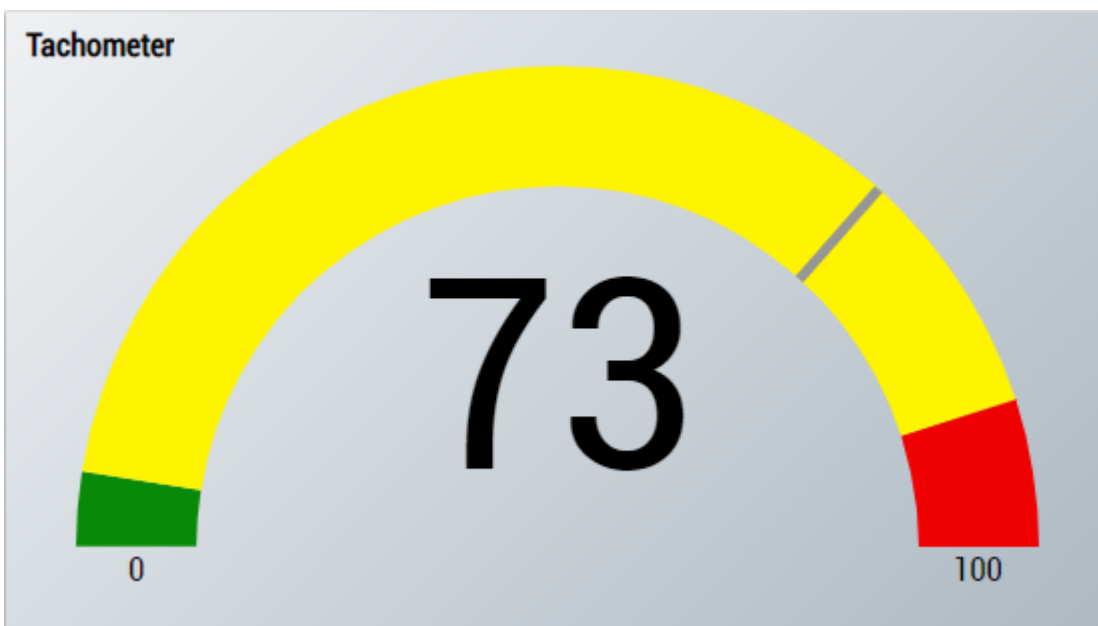
Standard HMI Controls

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.

Single Value

8

Last event:
10. Mar 2021 16:24:29

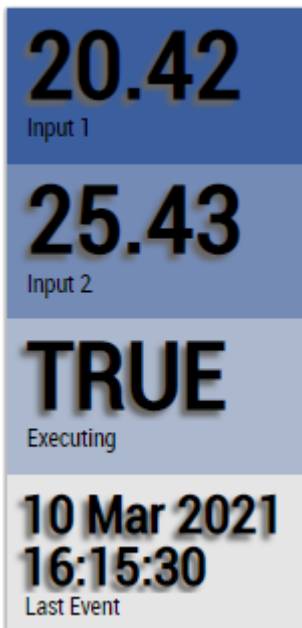
4. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Velocity.

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 Velocity Analysis 1Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.8 Analytics - WT

Die Algorithmen der Kategorie *Analytics-WT* bieten spezielle Funktionen für die Windbranche, zum Beispiel die Analyse von mittlerer Windgeschwindigkeit, Turbulenz und Turbulenzintensität.

6.5.8.1 WT Turbulence 1Ch

WT Turbulence 1Ch			
	Input	<Empty>	Num Cycles
			1
		Mean	Empty
		Turbulence	Empty
		Turbulence Inte...	Empty

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.

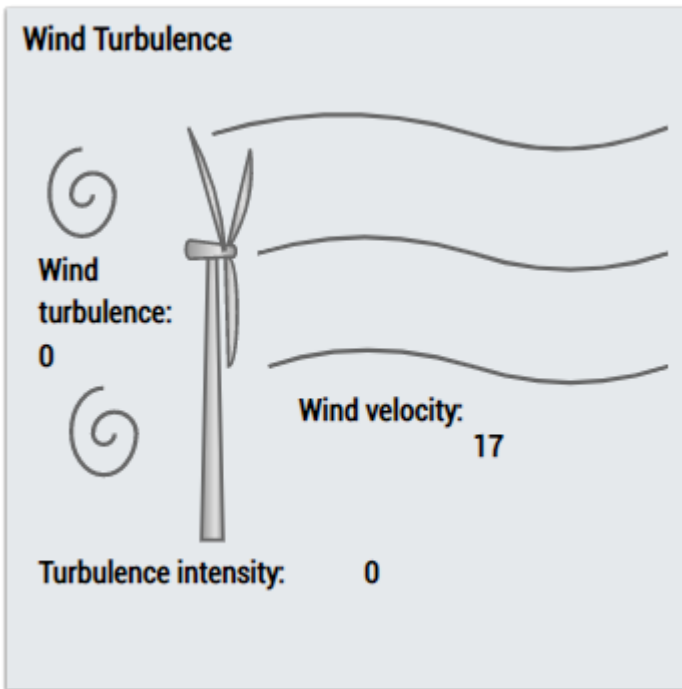
Ausgangswerte

- **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.



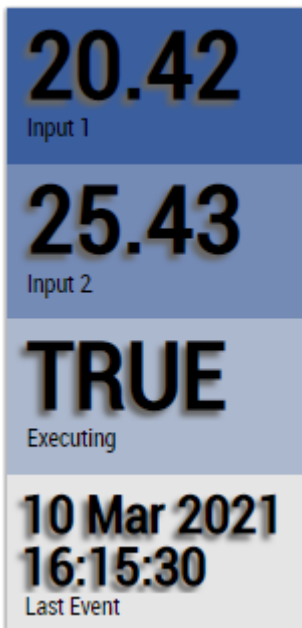
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Mean, Turbulence, Turbulence Intensity.

Data Table 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 WT Turbulence 1Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.9 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.

6.5.9.1 XY Gate Monitor 2Ch

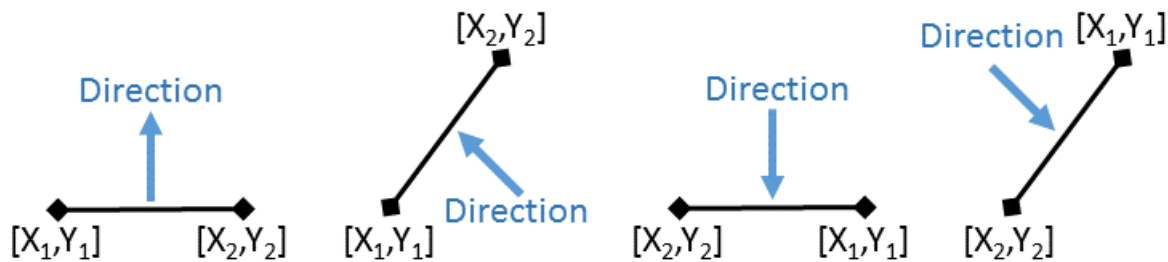
XY Gate Monitor 2Ch							
XY	Input ChX	<Empty>	-	Gate 1 X	1	Count Gate Inte...	Empty
	Input ChY	<Empty>	-	Gate 1 Y	1	Count Outlier l...	Empty
				Gate 2 X	2	Gate Intersection	Empty
				Gate 2 Y	2	Last Intersection	Empty
						Outlier Intersect...	Empty
						Position Interse...	Empty
					Position Interse...	Empty	

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.

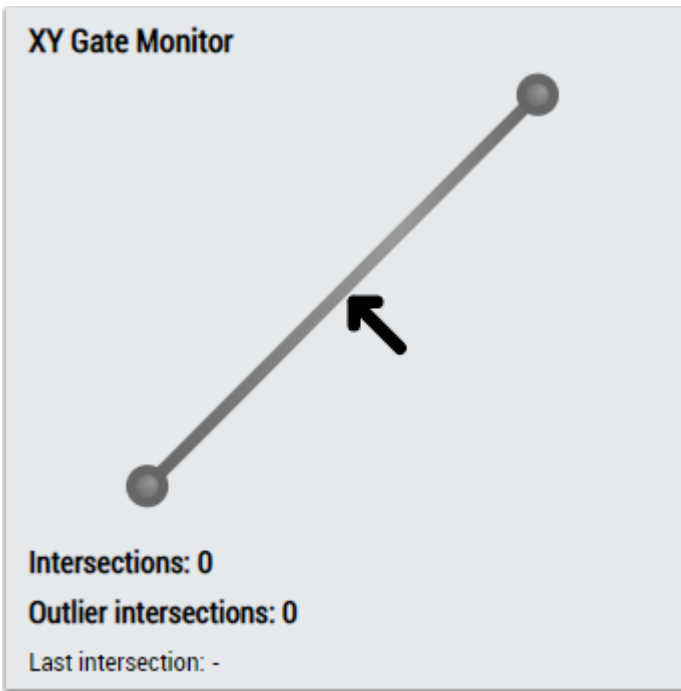
Ausgangswerte

- **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.



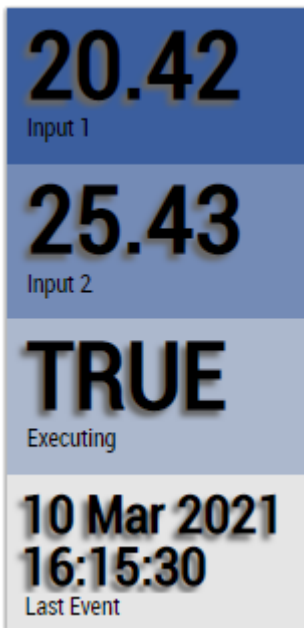
2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Executing, Gate Intersection, Outlier Intersection, Position Intersection X, Position Intersection Y, Count Gate Intersections, Count Outlier Intersections, Last Intersection, Classification.

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 kundenspezifische HMI Controls gemappt werden.

6.5.9.2 XY Shape Monitor Circle 2Ch

XY Shape Monitor Circle 2Ch						
XY 	Input ChX	<Empty>	Centre X	0	Count Intersecti...	Empty
	Input ChY	<Empty>	Centre Y	0	Intersection	Empty
			Radius	1	Last Intersection	Empty
					Within Shape	Empty

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.

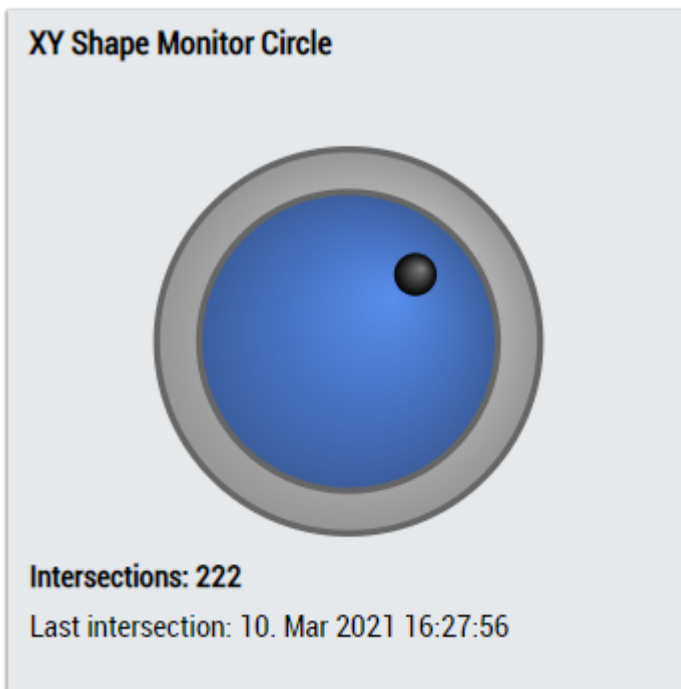
Ausgangswerte

- **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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

Alternativ können im Algorithmus XY Shape Monitor Circle 2Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.9.3 XY Shape Monitor Rectangle 2Ch

XY Shape Monitor Rectangle 2Ch							
XY 	Input ChX	<Empty>	-	Length X	2	Count Intersecti...	Empty
	Input ChY	<Empty>	-	Length Y	2	Intersection	Empty
				Lower Left Corner X	0	Last Intersection	Empty
				Lower Left Corner Y	0	Within Shape	Empty

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

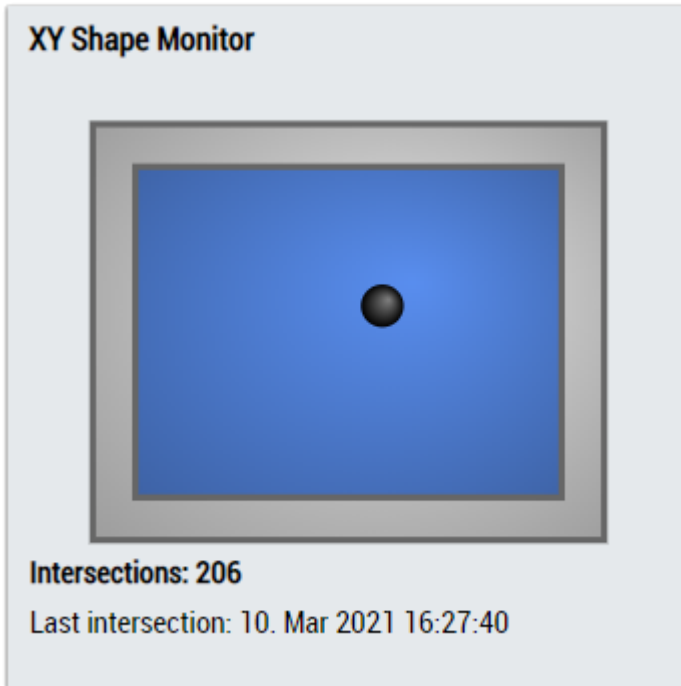
- **Within Shape:** Zeigt an, ob sich das Eingangssignal gegenwärtig innerhalb der bestimmten Form befindet.
- **Intersection:** Zeigt an, ob das Eingangssignal gegenwärtig die bestimmte Form kreuzt.
- **Count Intersections:** Zählt die Gesamtanzahl der Schnittpunkte von Eingangssignal und Form.

- **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.

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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

Alternativ können im Algorithmus XY Shape Monitor Rectangle 2Ch mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.9.4 XY Shape Monitor Triangle 2Ch

XY Shape Monitor Triangle 2Ch							
XY 	Input ChX	<Empty>	-	Corner 1 X	1	Count Intersecti...	Empty
	Input ChY	<Empty>	-	Corner 1 Y	1	Intersection	Empty
				Corner 2 X	2	Last Intersection	Empty
				Corner 2 Y	2	Within Shape	Empty
				Corner 3 X	3		
				Corner 3 Y	3		

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.

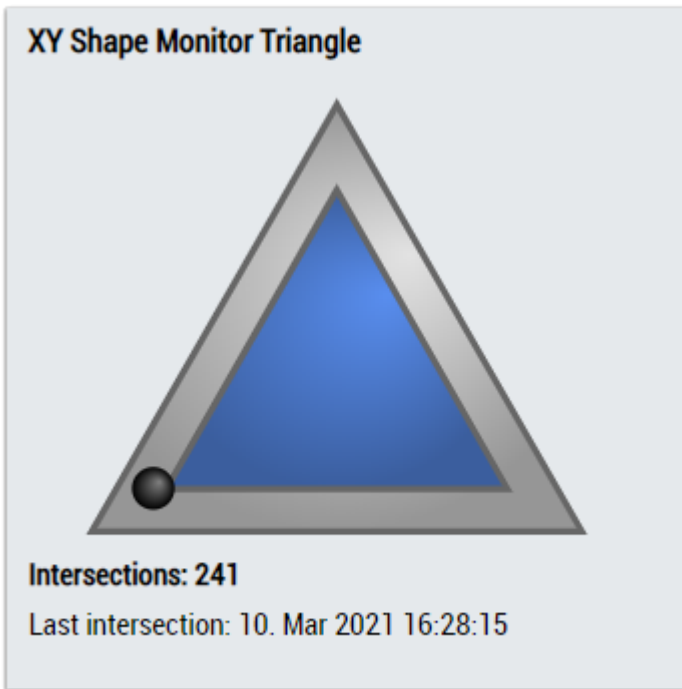
Ausgangswerte

- **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.

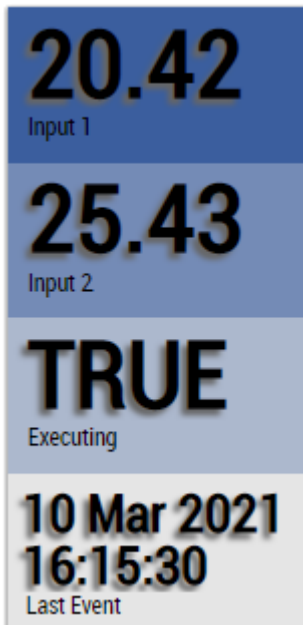


3. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Within Shape, Intersection, Count Intersections, Last Intersection.

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 kundenspezifische HMI Controls gemappt werden.

6.5.10 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.

6.5.10.1 Array Statistics

Array In		Array Statistics		Min	Empty
<Empty>	-	Use Bessel Correction	<input checked="" type="checkbox"/>	Idx Min	Empty
		Threshold Reversal	0.5	Max	Empty
		Threshold Delta	0.5	Idx Max	Empty
				Max Delta	Empty
				Idx Max Delta	Empty
				Count Peaks	Empty
				Count Valleys	Empty
				Sum	Empty
				Mean	Empty
				Standard Deviat...	Empty

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 erwartungstreu 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

$$s' = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x[n] - \bar{x})^2}$$

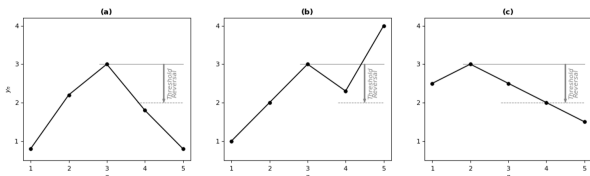
Die empirische Standardabweichung, mit Bessel'scher Korrektur

$$s = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} (x[n] - \bar{x})^2}$$

- **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.

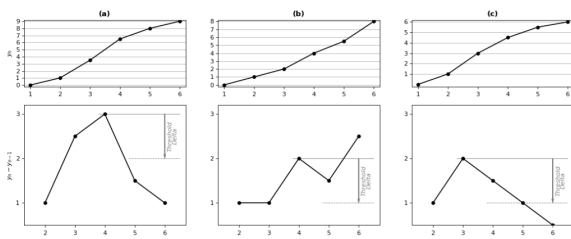
- (a) Der Wert y_3 wird direkt nach der Verarbeitung des Werts y_4 als Peak identifiziert, da die Differenz zwischen y_3 und y_4 größer ist als *Threshold Reversal*.
- (b) Der Wert y_3 wird nicht als Peak identifiziert, da die Differenz zwischen y_3 und y_4 kleiner ist als *Threshold Reversal* und die 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.

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.

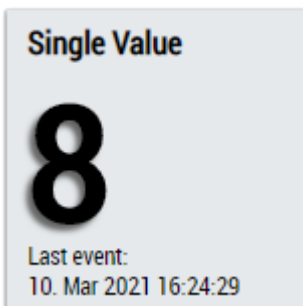
Standard HMI Controls

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.

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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

Alternativ können im Algorithmus Standard Deviation mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.10.2 Correlation Function

Correlation Function						
Channel Ref	<Empty>	-	Num Channels	+ -	Output 00	Empty
Channel 00	<Empty>	-	Correlation Mode	Normed	Minimizing Lag...	Empty
			Step Size	1	Minimum Coef...	Empty
			Window Mode	FixWindow	Maximizing Lag...	Empty
			Window Size	400	Maximum Coef...	Empty
			Minimum Lag	0		
			Maximum Lag	10		

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 jedem *WindowSize* Zyklen neu und die Ausgangswerte werden nur beim Durchlaufen des letzten Zyklus eines Fensters aktualisiert.

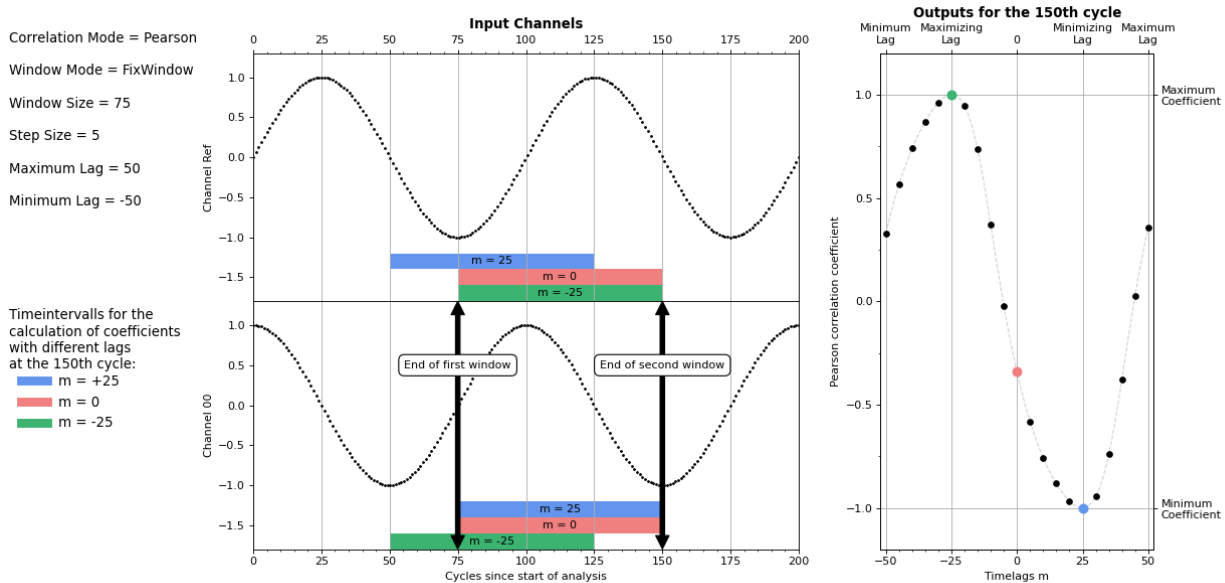
Für m ungleich Null verschiebt sich das Fenster für das entsprechende Signal um m Zyklen. Damit bleibt die Anzahl von Werten, welche in die Berechnung der Koeffizienten mit einbezogen werden, für alle Werte von m gleich. Eine Ausnahme hierzu bilden nur die Ergebnisse aus den ersten Zyklen nach Start 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 0n 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:*

$$\rho_{xy}[m, t_n] = \frac{cov(x, y)[m, t_n]}{\sigma_x[m, t_n] \sigma_y[m, t_n]}, \text{ where } \sigma_x^2[m, t_n] = cov_{xx}[m, t_n]$$

- **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 $m = \text{Minimum Lag}$, $m = \text{Minimum Lag} + \text{Step Size}$, ..., $m = -\text{Step Size}$, $m = 0$, $m = +\text{Step Size}$, ..., $m = \text{Maximum Lag}$ 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:

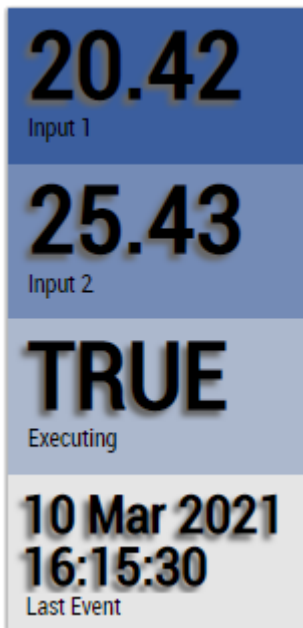
1. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Minimaler Koeffizient, maximaler Koeffizient, minimaler lag, maximaler lag und den Koeffizienten Array.

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 kundenspezifische HMI Controls gemappt werden.

6.5.10.3 Correlation Function Reference

Correlation Function Reference			
Channel 00	<Empty>	Num Channels	+ -
		Correlation Mode	Normed
		File Path	3:\Boot\CorrelationFunctionReference.ta
		Step Size	1
		Window Mode	FixWindow
		Minimum Lag	0
		Maximum Lag	10
		Value Read	Empty
		Output 00	Empty
		Minimizing Lag...	Empty
		Minimum Coef...	Empty
		Maximizing Lag...	Empty
		Maximum Coef...	Empty

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.

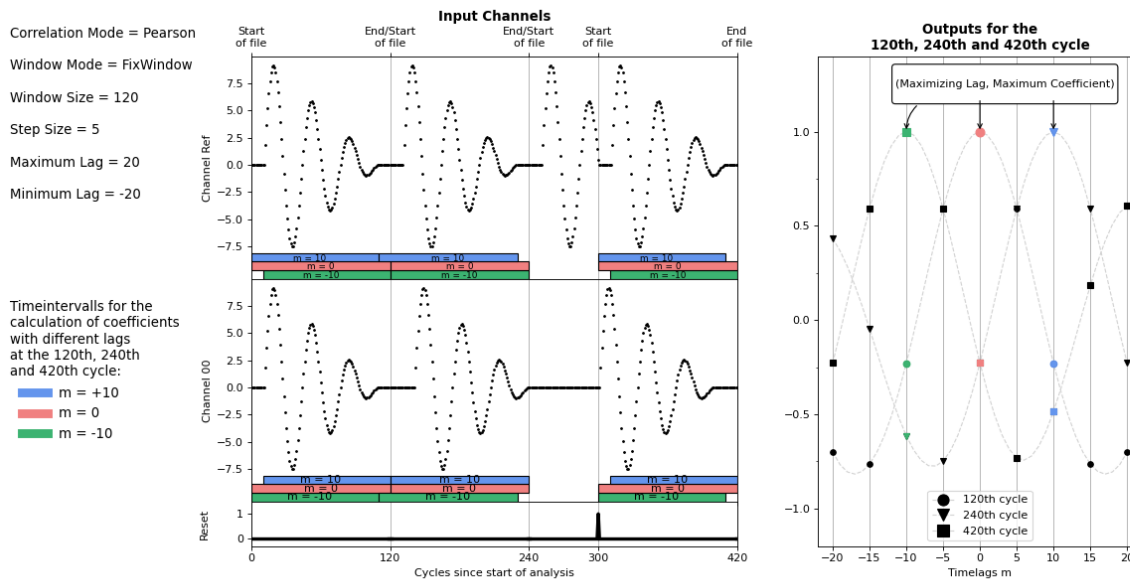
Die Korrelationskoeffizienten können über verschiedene Zeitfenster berechnet werden. Diese werden durch den Parameter *Window Mode* gesetzt. Im *Continuous* Modus werden alle Werte seit Beginn der Analyse in die Berechnungen miteinbezogen. Im *SlidingWindow* Modus läuft die Berechnung kontinuierlich 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 Wert 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 0n 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*:

$$\rho_{xy}[m, t_n] = \frac{cov(x, y)[m, t_n]}{\sigma_x[m, t_n] \sigma_y[m, t_n]}, \text{ where } \sigma_x^2[m, t_n] = cov_{xx}[m, t_n]$$

- **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 $m = \text{Minimum Lag}$, $m = \text{Minimum Lag} + \text{Step Size}$, ..., $m = -\text{Step Size}$, $m = 0$, $m = +\text{Step Size}$, ..., $m = \text{Maximum Lag}$ 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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

Alternativ können im Algorithmus Correlation Function Reference mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.10.4 Linear Regression Fitting

Linear Regression Fitting			
bFitting	<Empty>	Num Channels	MSE
Dependent	<Empty>	File Path	Result
Input 00	<Empty>	Involve Existing File	Output 00
		Step Size	Output 01

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:

$$y = \beta_0 + \sum_{i=1}^n \beta_i \times \text{Input } O_i$$

In jedem Zyklus werden die Werte für β_0 bis β_n mit folgender Vorschrift neu berechnet:

$$\beta_i = \begin{cases} \beta_i - \gamma \times \text{Input } O_i \times (y - \text{Dependent}), & i = 1, 2, \dots, n \\ \beta_i - \gamma \times (y - \text{Dependent}), & i = 0 \end{cases}$$

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.
- **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.

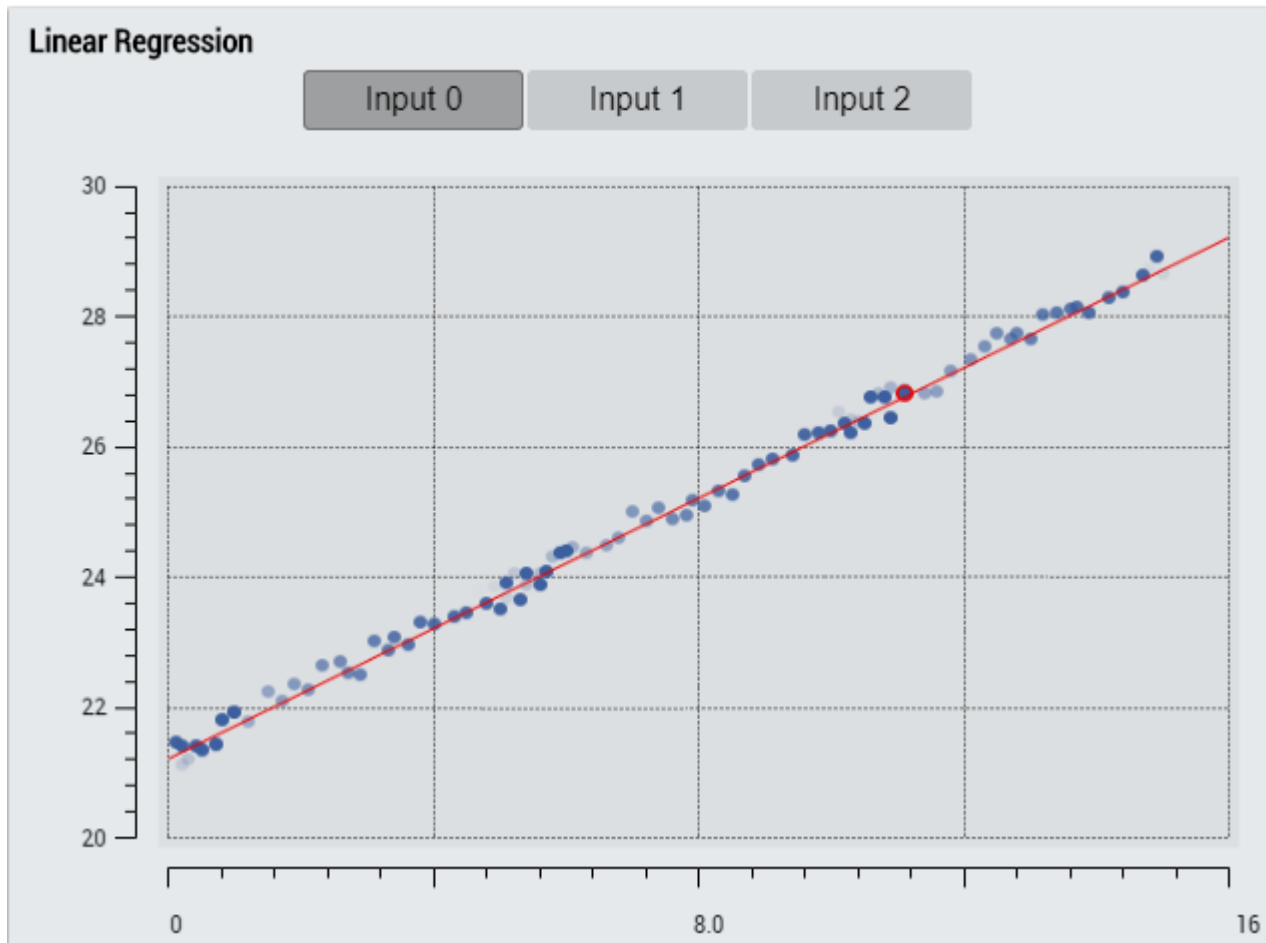
Ausgangswerte

- **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.

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

20.42

Input 1

25.43

Input 2

TRUE

Executing

10 Mar 2021 16:15:30

Last Event

Alternativ können im Algorithmus Linear Regression Curve Fitting mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.10.5 Linear Regression Inference

Linear Regression Inference ⊗ ⌂ ⏴ ⏵

	Input 00 <input type="text" value="<Empty>"/>	-	Num Channels <input type="text" value="0"/>	+ -	Result <i>Empty</i>
			Weights 00 <input type="text" value="0"/>		
			Weights 01 <input type="text" value="0"/>		

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.

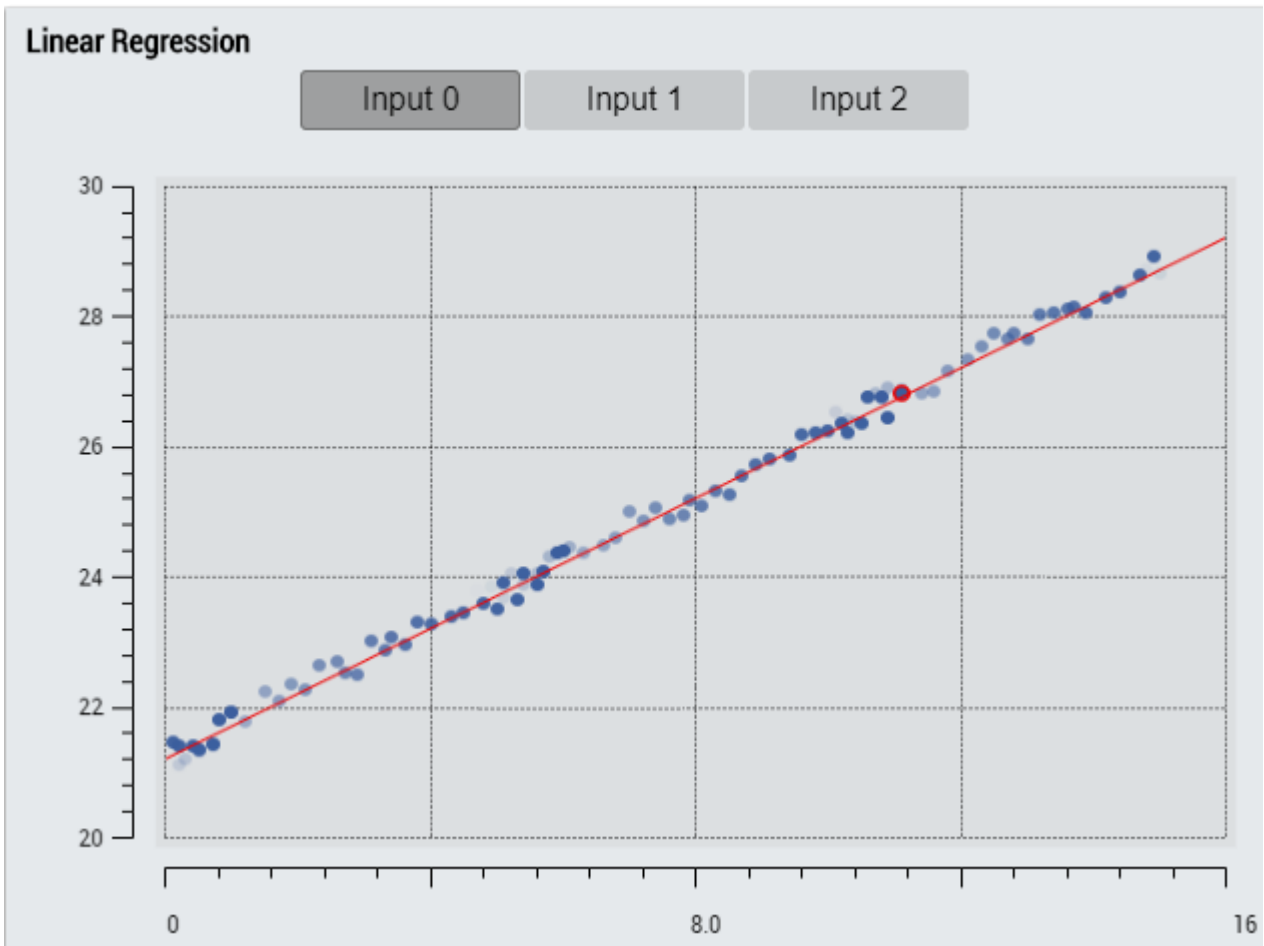
Ausgangswerte

- **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.

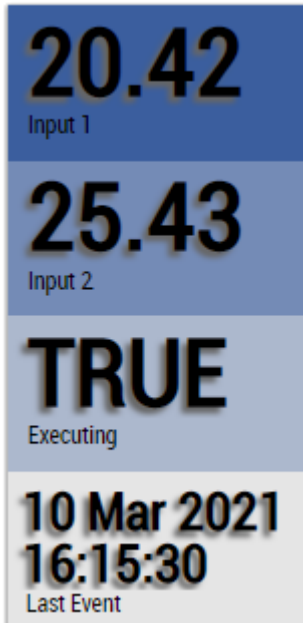


2. Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: MSE, Ergebnis, Output-Koeffizienten.

Data Table Vertical

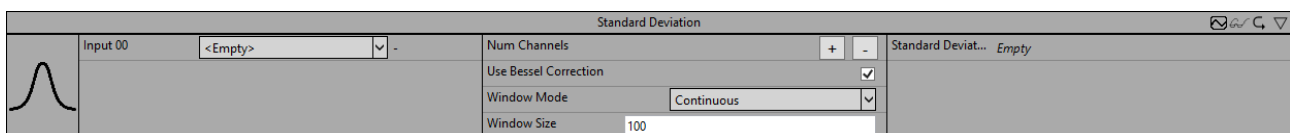
	Data Table
Input 1	20.42
Input 2	25.43
Executing	TRUE
Last Event	10 Mar 2021 16:15:29

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 Linear Regression Inference mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.10.6 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 erwartungstreuere Ergebnis zu erhalten, ist dieser Parameter zu aktivieren.

Die empirische Standardabweichung, ohne Bessel'sche Korrektur

$$s' = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x[n] - \bar{x})^2}$$

Die empirische Standardabweichung, mit Bessel'scher Korrektur

$$s = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} (x[n] - \bar{x})^2}$$

- Window Mode:** Verwendeter Fenstermodus. Beeinflusst die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogen 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 einbezogen. 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:

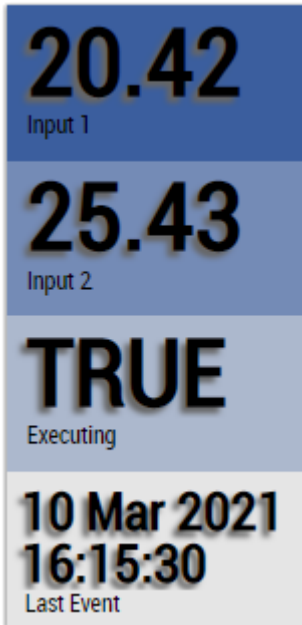
- Das Table Control oder Multivalue Control visualisiert alle Ausgangswerte: Standard Deviation Results.

Data Table Vertical

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

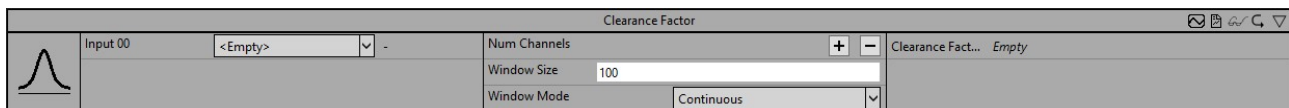
DataTable Horizontal

	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 kundenspezifische HMI Controls gemappt werden.

6.5.10.7 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 einbezogen 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 einbezogen 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 einbezogen. 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.

Data Table Vertical

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

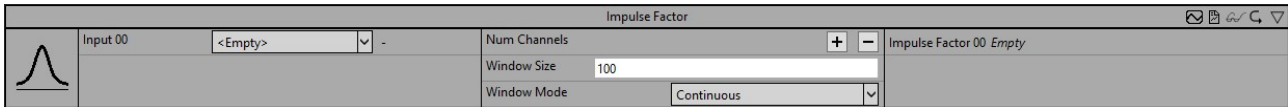
DataTable Horizontal

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

20.42 Input 1
25.43 Input 2
TRUE Executing
10 Mar 2021 16:15:30 Last Event

Alternativ können mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.10.8 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 einbezogen 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 einbezogen 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 einbezogen. 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

20.42
Input 1

25.43
Input 2

TRUE
Executing

10 Mar 2021
16:15:30
Last Event

Abb. 3:

Alternativ können mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.10.9 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.

$$\text{Shape Factor} = \frac{x_{\text{rms}}}{\frac{1}{N} \sum_{n=1}^N |x[n]|}$$

Die Anzahl der Eingangsdaten, die mit in die Berechnung einbezogen 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 einbezogen 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 einbezogen. 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.

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

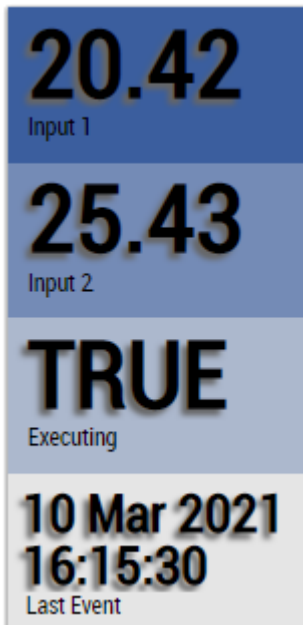


Abb. 4:

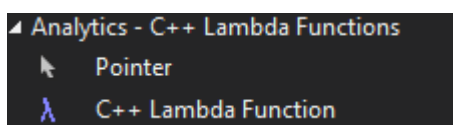
Alternativ können mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.11 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.

6.5.11.1 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.

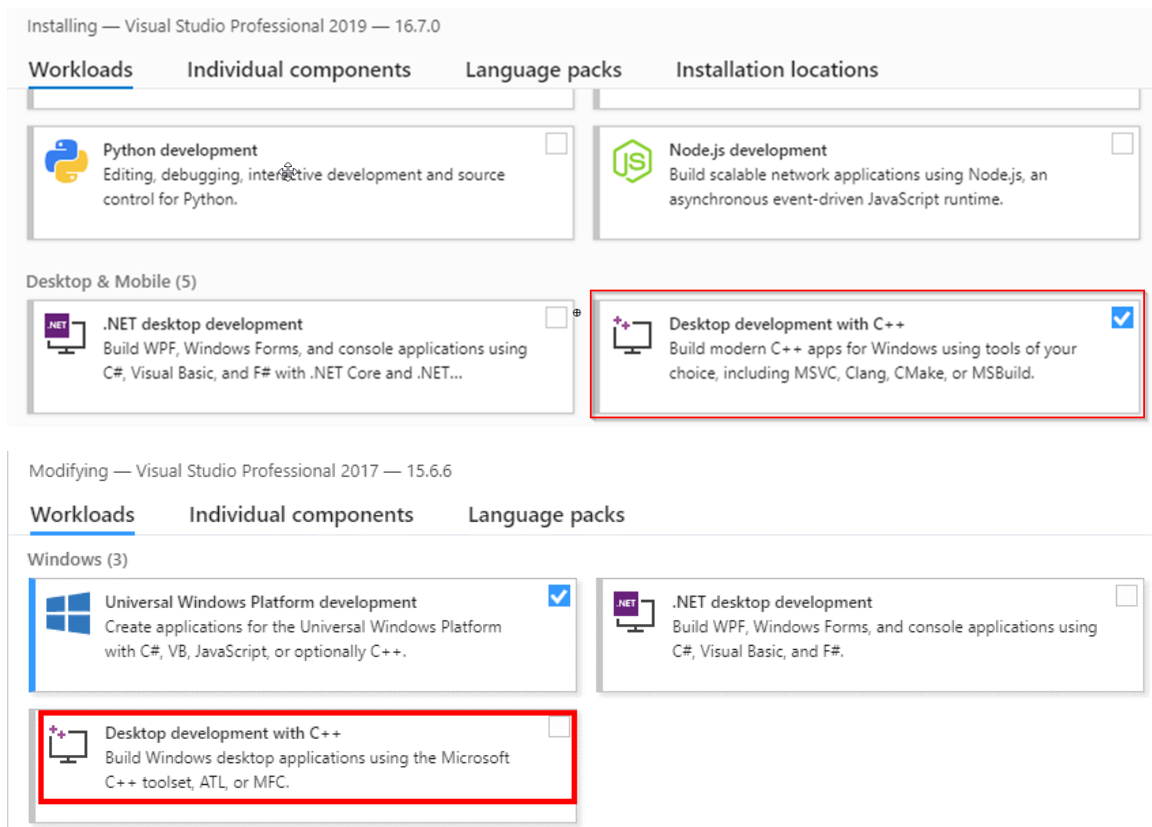
6.5.11.1.1 Anforderungen

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)

6.5.11.1.2 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-Symboleiste.
- 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.
    
```

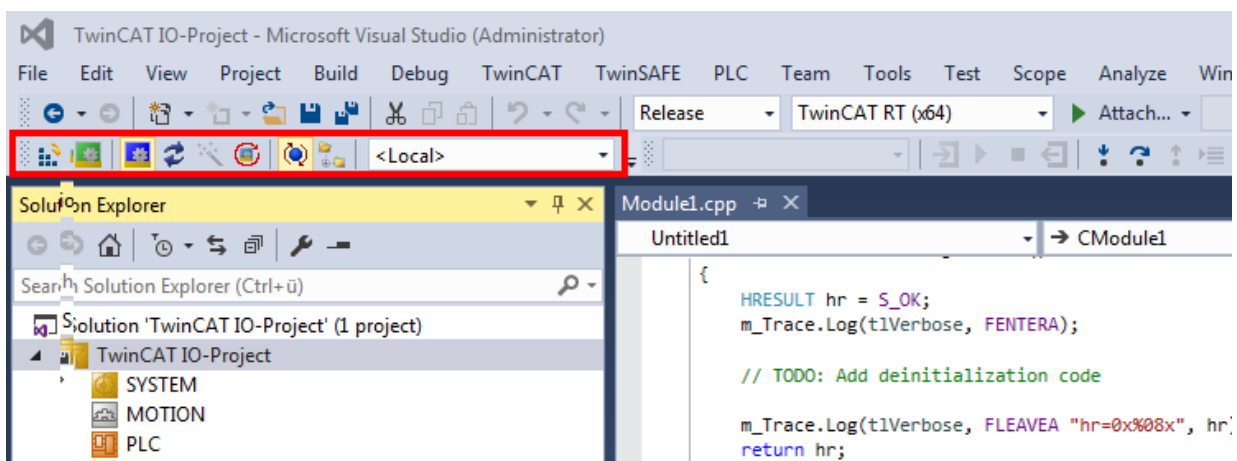
6.5.11.1.2.1 Visual Studio - TwinCAT XAE Base-Symbolleiste

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.



6.5.11.1.2.2 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

i 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.

6.5.11.1.2.2.1 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 [▶ 240], welches entsprechend im Projekt zur Signierung konfiguriert wird.

Testsignierung

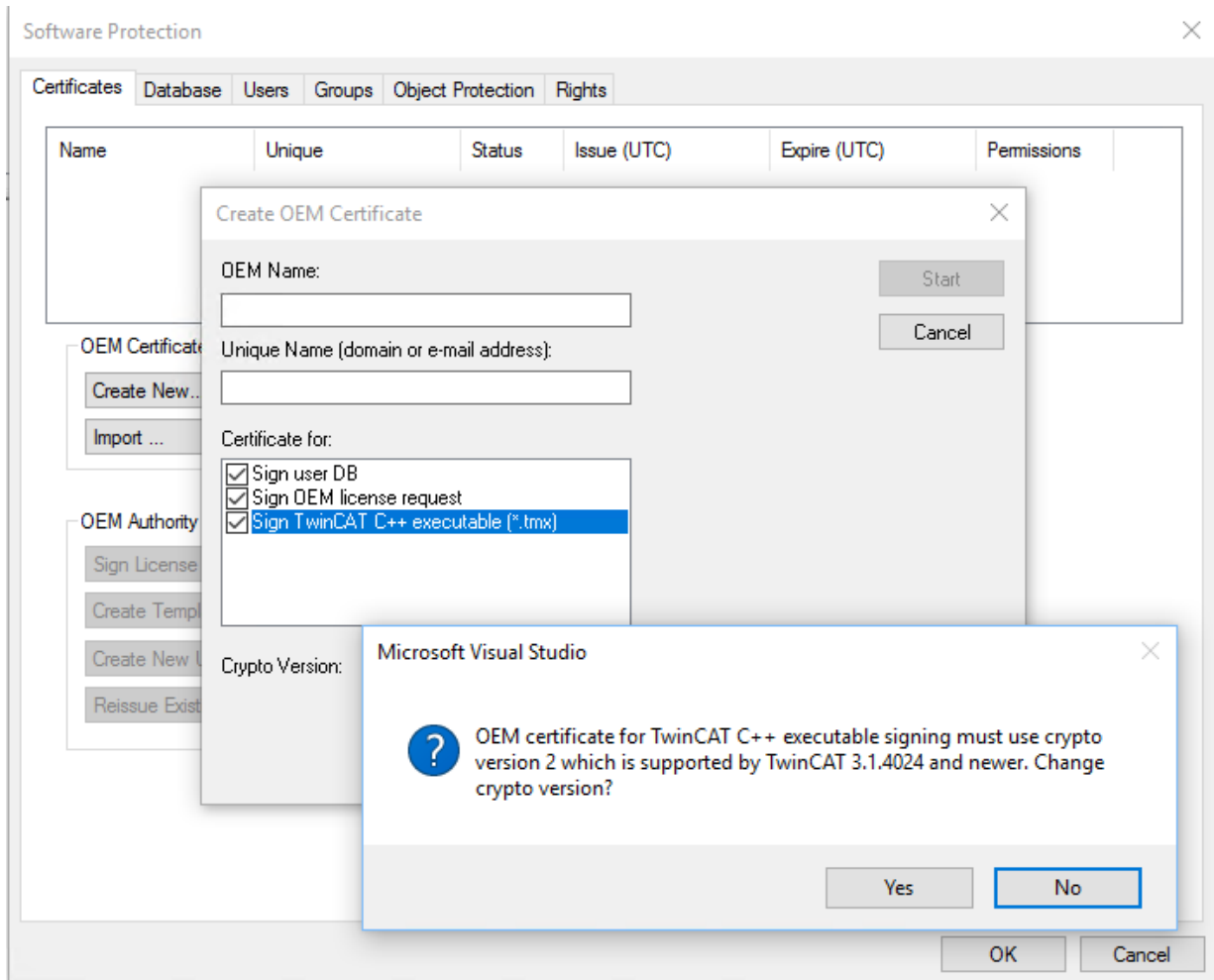
Das Nutzerzertifikat kann lokal in TwinCAT erstellt werden. Solange es nicht von Beckhoff gegengezeichnet ist, ist es nötig den Testmode zu aktivieren, wie hier [▶ 240] beschrieben.

Sobald das TwinCAT-Nutzerzertifikat von Beckhoff gegengezeichnet [▶ 242] wurde, kann auf den Testmodus entsprechend verzichtet werden. Er kann analog zum aktivieren auch wieder deaktiviert werden.

6.5.11.1.2.2.1.1 Testsignierung

Die Testsignierung kann für TwinCAT mit dem gleichen TwinCAT-Nutzerzertifikat erfolgen, wie bei der eigentlichen Auslieferung (vgl. TwinCAT 3 Nutzerzertifikat beantragen [▶ 243]).

1. Für einen Testbetrieb z. B. während der Software-Entwicklung genügt die Erstellung eines TwinCAT-Nutzerzertifikates, wie [hier \[► 244\]](#) beschrieben. Beachten Sie dabei, dass Sie den Verwendungszweck „Sign TwinCAT C++ executable (*.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

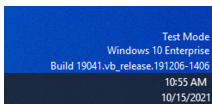
Führen Sie mittels Administrator-Eingabeaufforderung folgendes aus:

```
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.

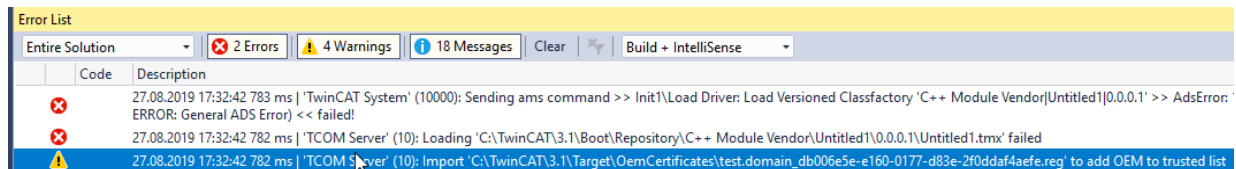
```
<Key Name="System">
  <Value Name="RunAsDevice" Type="DW">1</Value>
  <Value Name="RTimeMode" Type="DW">0</Value>
  <Value Name="AmsNetId" Type="BIN">052445B00101</Value>
  <Value Name="LockedMemSize" Type="DW">33554432</Value>
  <Value Name="EnableTestSigning" Type="DW">1</Value>
</Key>
```

Danach den TwinCAT System Service neu starten:

```
doas service TcSystemService restart
```

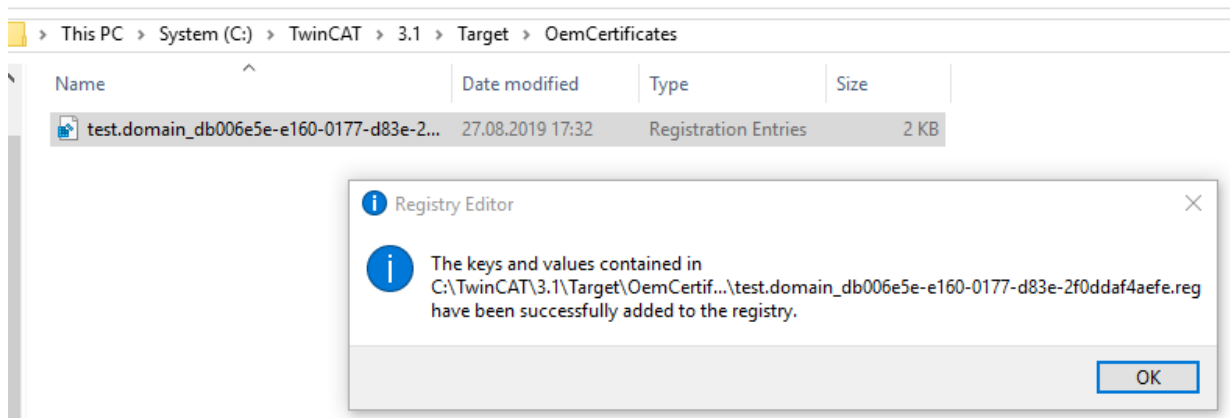
Nach der jeweiligen Vorgehensweise akzeptiert das System alle signierten Treiber zur Ausführung.

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](#) [▶ 254].

Wenn Sie das TwinCAT-Nutzerzertifikat ohne TestMode zur Auslieferung nutzen wollen, müssen Sie das Zertifikat von Beckhoff gegenzeichnen [▶ 242] lassen.

6.5.11.1.2.2.1.2 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**

i 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.

i 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.

i 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.

6.5.11.1.2.2.1.2.1 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“ [► 244].
5. Ermitteln Sie mit Hilfe des TwinCAT Engineerings den „File Fingerprint“ der OEM-Zertifikatsdatei (siehe File Fingerprint der Zertifikatsdatei ermitteln [► 247]). 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.

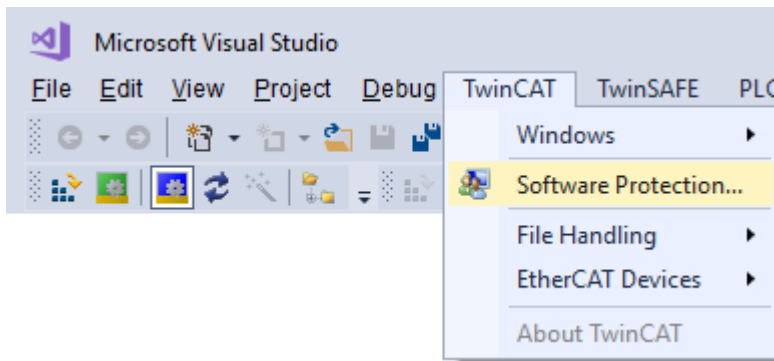
6.5.11.1.2.2.1.2.2 Erstellung der Certificate Request Datei für TC0008



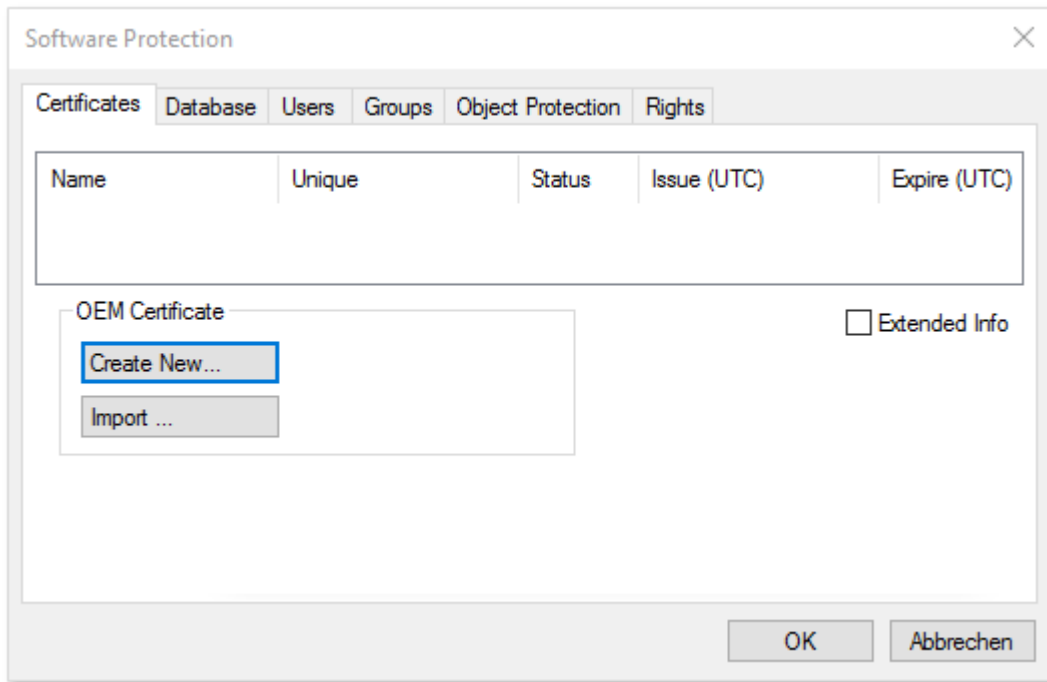
Systemvoraussetzungen

- Min. TwinCAT 3.1 Build 4024
- Min. Windows 10 oder TwinCAT/BSD (auf dem Zielsystem)

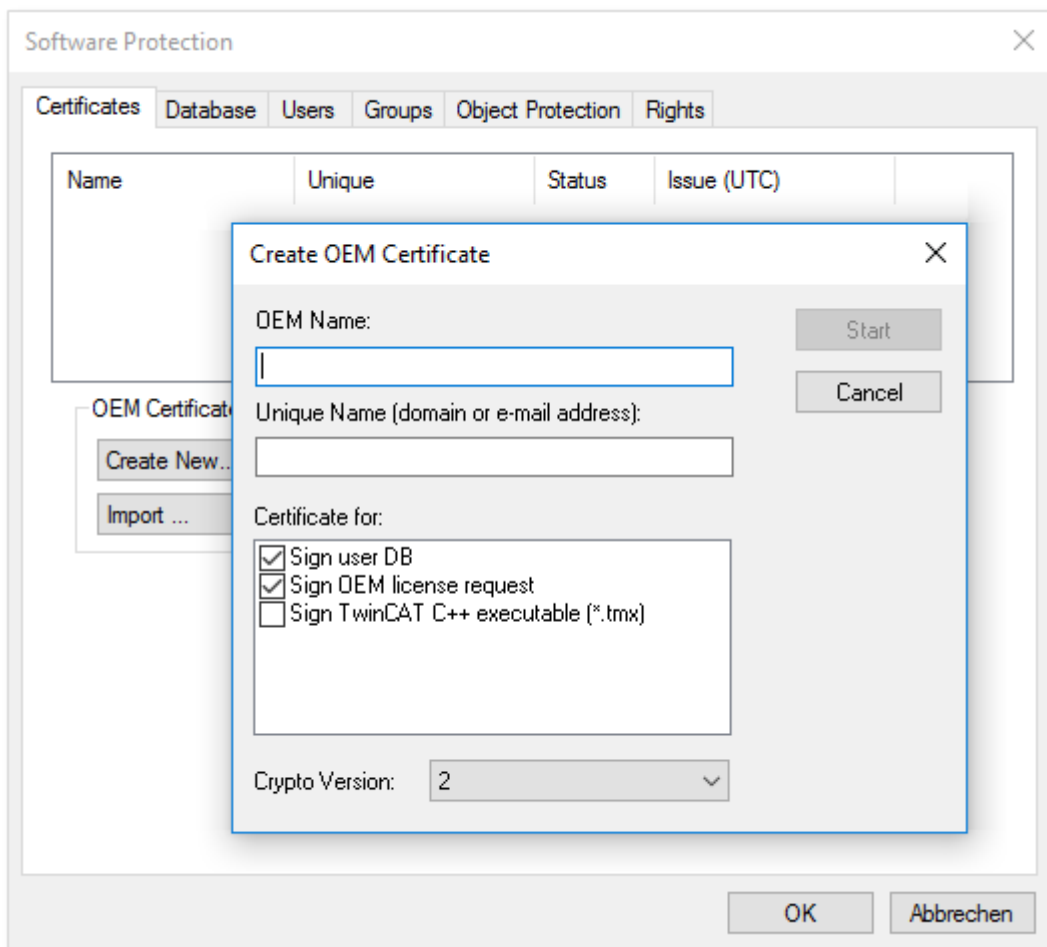
1. Rufen Sie den Konfigurator der Software Protection auf. Wählen Sie dazu im Hauptmenü unterhalb des Punktes **TwinCAT** den Menüpunkt **Software Protection** aus:



- In dem sich öffnenden Fenster wählen Sie die Registerkarte **Certificates** aus. Klicken Sie auf **Create New....**:

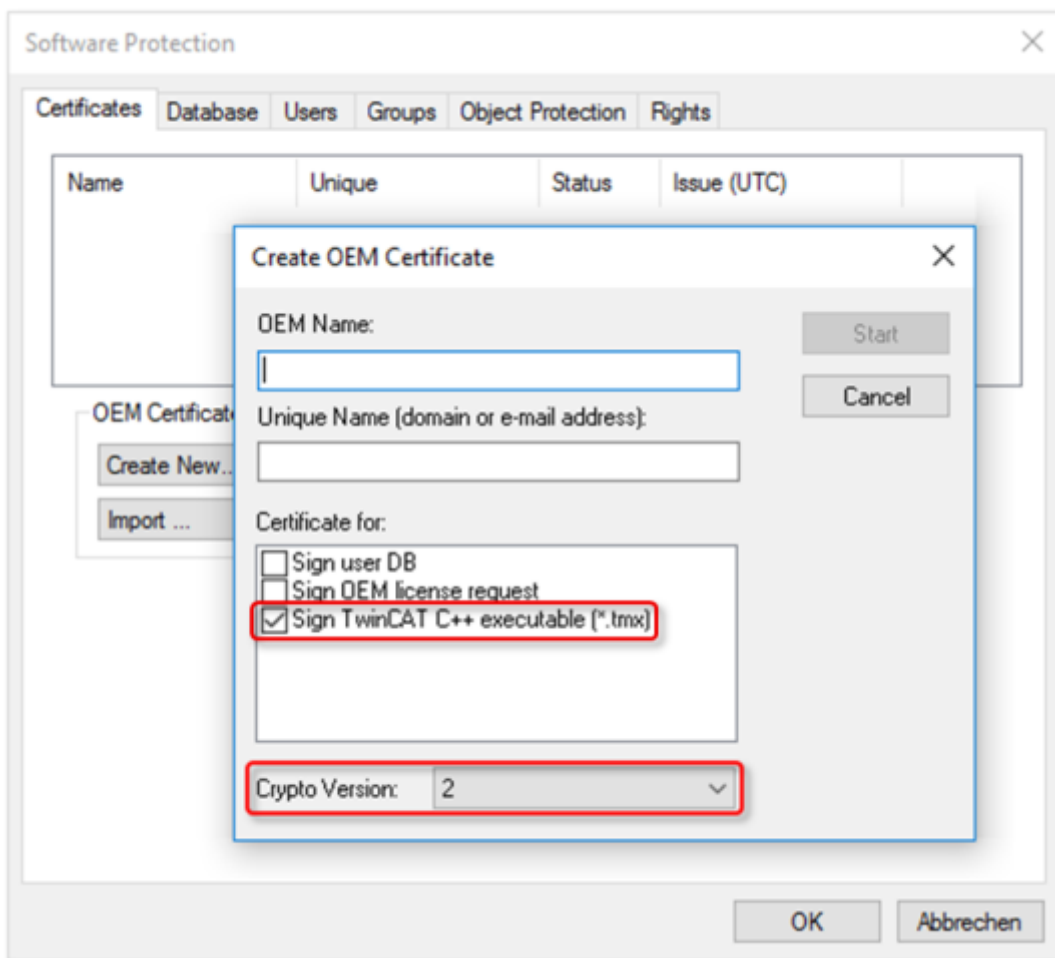


- Das Eingabefenster **Create OEM Certificate** öffnet sich:



- 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 Checkboxes 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 [► 247] 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.

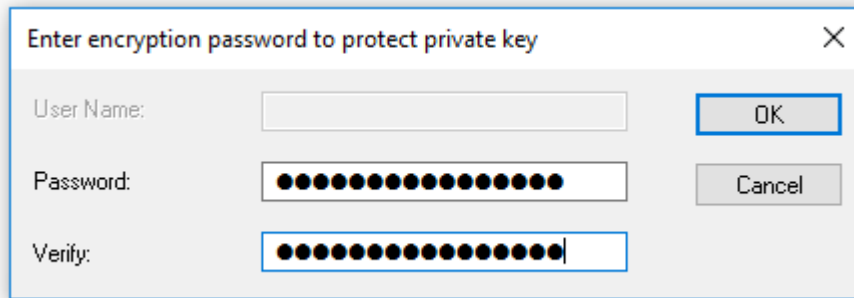
● **Wichtig: Passwort-Sicherheit!**

i 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**

i 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](#) [▶ 243] beschrieben.

6.5.11.1.2.2.1.2.3 File Fingerprint der OEM-Zertifikatsdatei ermitteln

Diese Funktionalität benötigen Sie für die Beantragung eines **TwinCAT OEM Certificate Extended Validation** (TC0008).

i Systemvoraussetzung

Diese Funktionalität erfordert mindestens die Version TwinCAT 3.1 Build 4024.

i 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

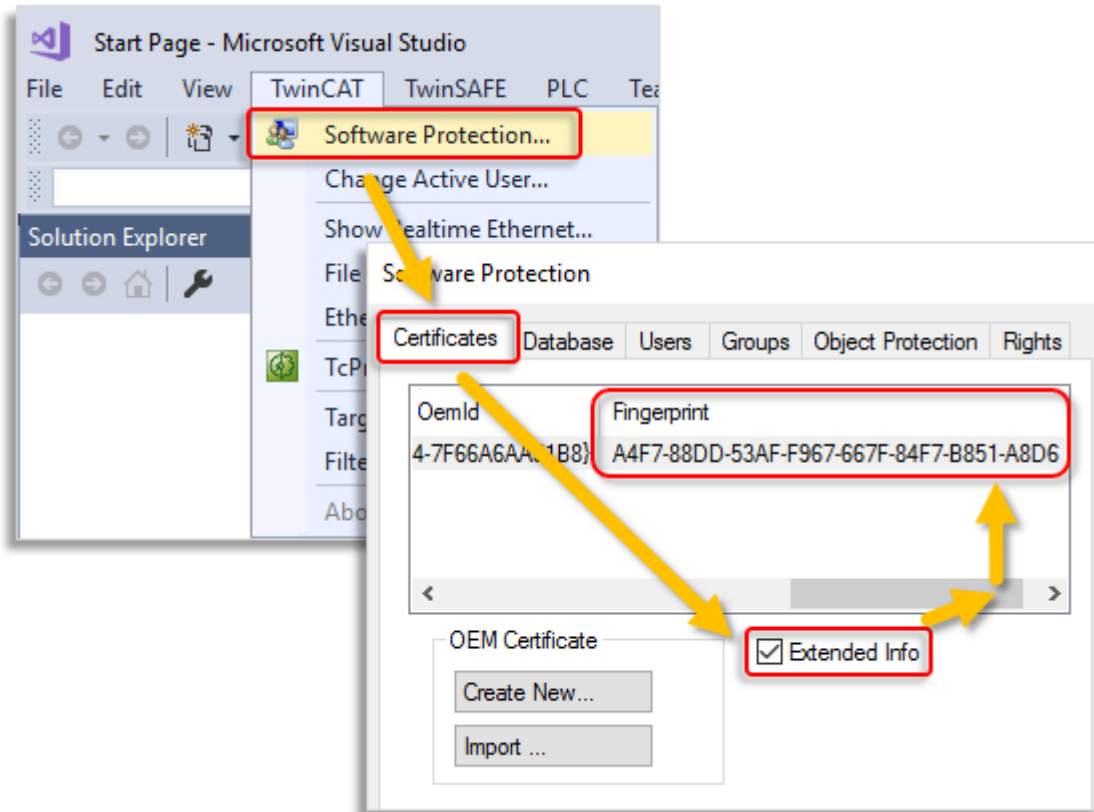
Für diese Funktion ist es erforderlich, dass die OEM-Zertifikatsdatei in diesem Verzeichnis liegt: „c:\twincat\3.1\customconfig\certificates“.

In diesem Verzeichnis liegt ihr OEM-Zertifikat, sofern Sie bereits eines haben und dieses verlängern wollen.

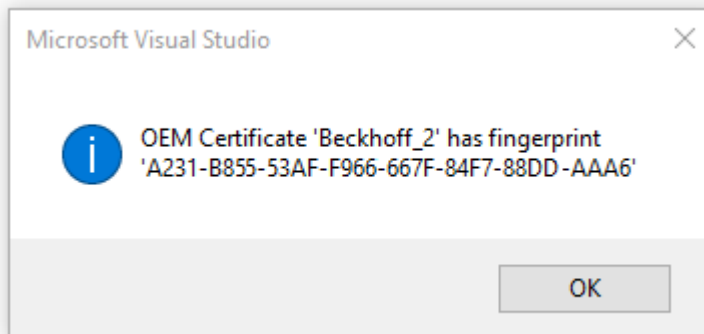
Sofern Sie beim Erstellen der „OEM Certificate Request Datei“ das vorgeschlagene Verzeichnis nicht geändert haben, liegt die Datei bereits in diesem Verzeichnis.

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.

6.5.11.1.2.2.1.2.4 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)

i 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.

i 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.

6.5.11.1.2.2.2 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 \[► 249\]](#)“ 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 \[► 249\]](#)
- [\(Test-\)Zertifikate löschen \[► 252\]](#)

Weitere Referenzen:

MSDN, MakeCert Testzertifikate (Windows Treiber),

<https://docs.microsoft.com/en-us/windows-hardware/drivers/install/makecert-test-certificate>

6.5.11.1.2.2.2.1 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.

i 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:

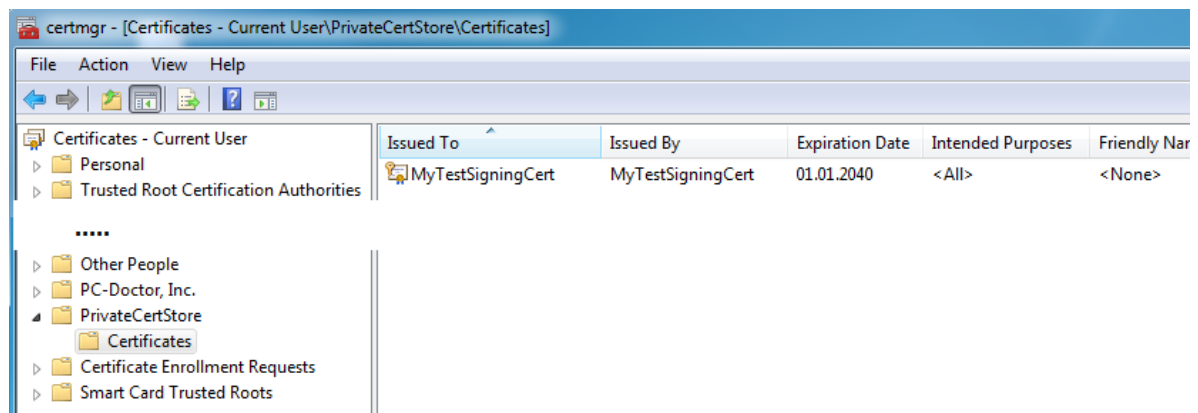
geben Sie im Engineering System folgenden Befehl in die Visual Studio 2010 / 2012 Eingabeaufforderung mit Administratorrechten ein (siehe Hinweis oben):

```
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](#) [▶ 253] 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**):



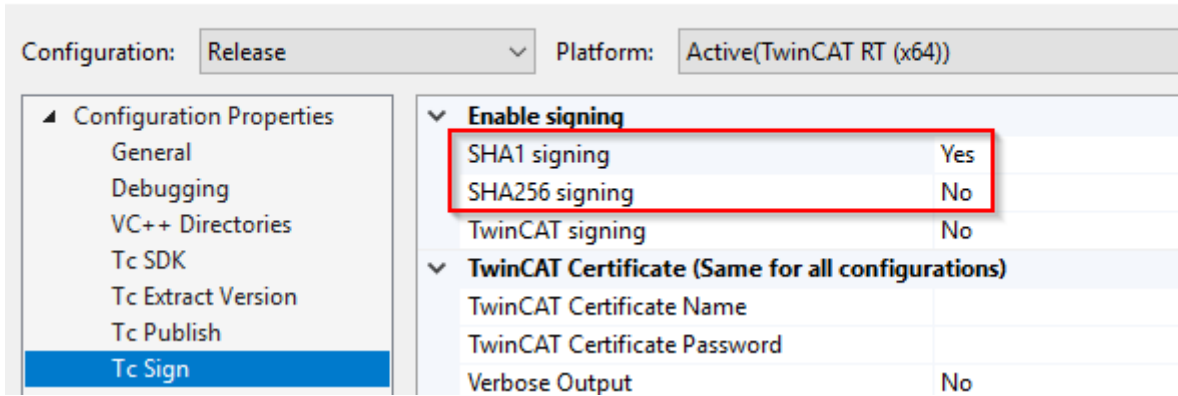
2. Auf XAE:

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:

Untitled1 Property Pages



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

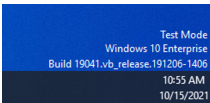
Führen Sie mittels Administrator-Eingabeaufforderung folgendes aus:

```
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> 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>
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>

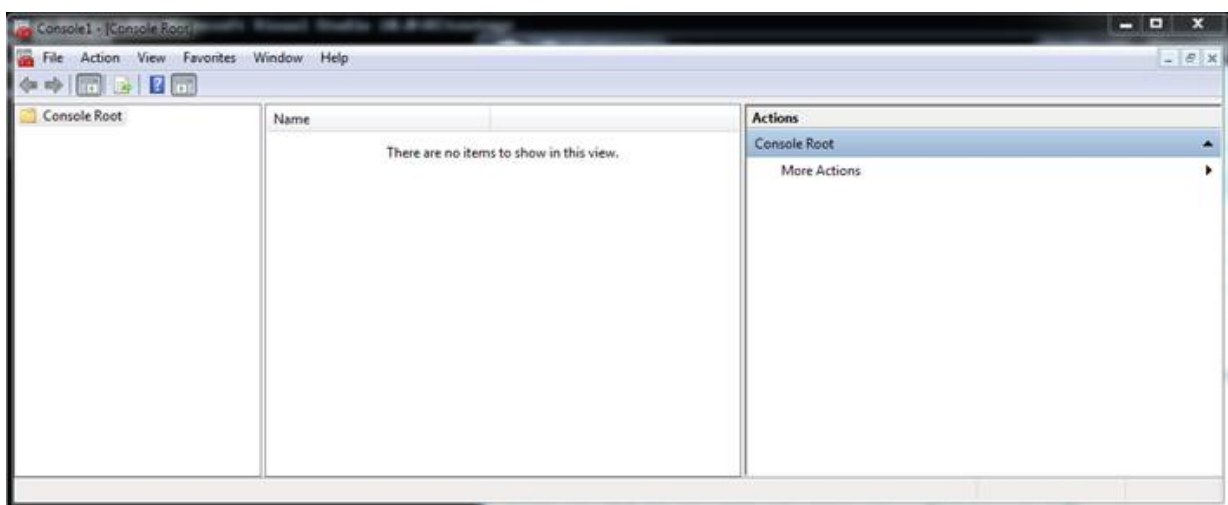
6.5.11.1.2.2.2 Testzertifikat löschen

Dieser Artikel beschreibt, wie ein Testzertifikat gelöscht wird.

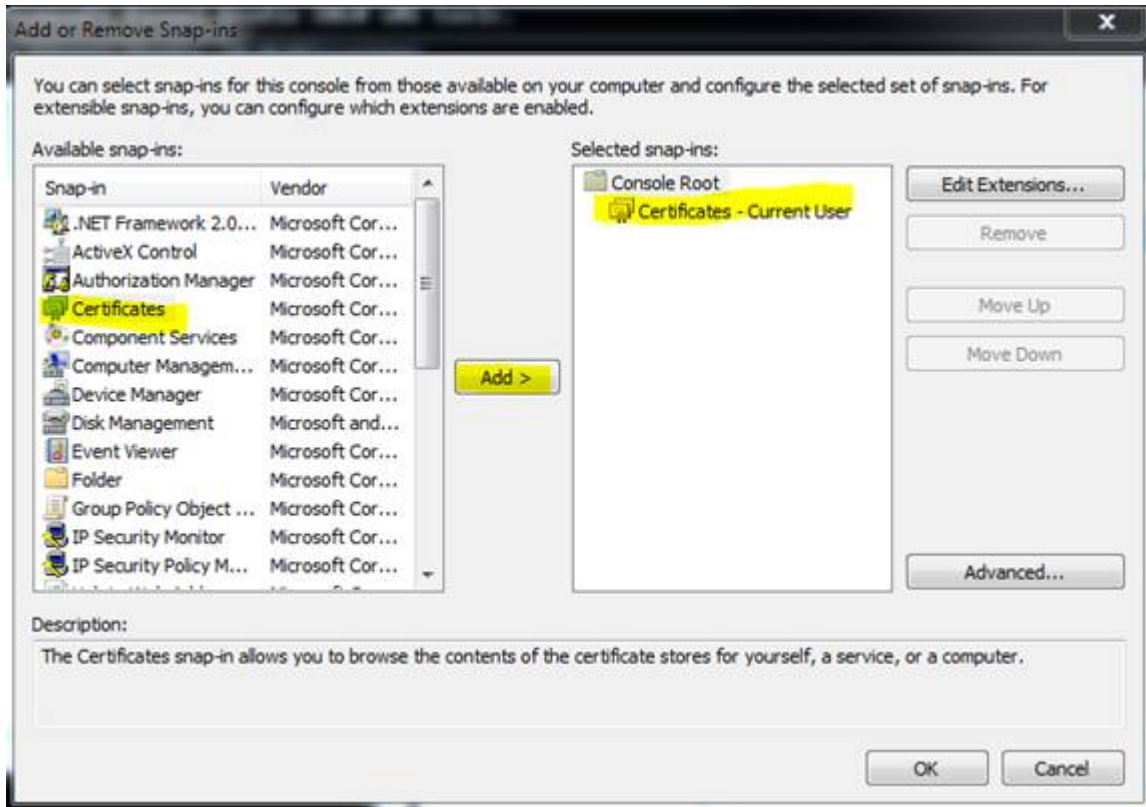
Übersicht

Ein Zertifikat kann mit der Microsoft Management Console gelöscht werden:

1. Starten Sie die Management console MMC.exe über Start-Menü oder die Oberfläche.

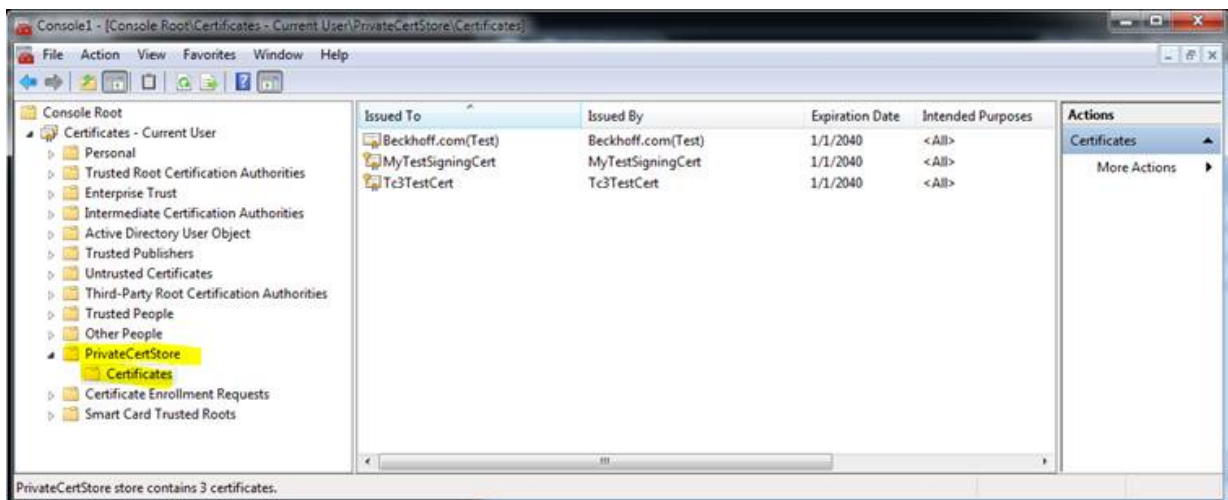


2. Klicken Sie im Menü auf **File -> Add/Remove Snap-in..** und wählen Sie Zertifikat-Snap-In für den aktuellen Nutzer, schließen Sie mit **OK** ab.



⇒ Die Zertifikate sind im Knoten unter **PrivateCertStore/Certificates** aufgelistet.

3. Wählen Sie das zu löschende Zertifikat aus.



6.5.11.1.2.2.3 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.

6.5.11.1.2.3 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\`

6.5.11.1.2.4 Umgebungsvariablen

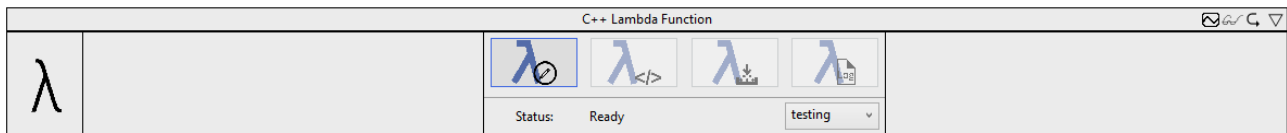
Abschließend müssen zwei Umgebungsvariablen mit dem Zertifikatsnamen erstellen werden. Diese werden während des Generierens der Lambda Funktion ausgelesen und signieren die Lambda Funktion mit den entsprechenden Zertifikaten.

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.

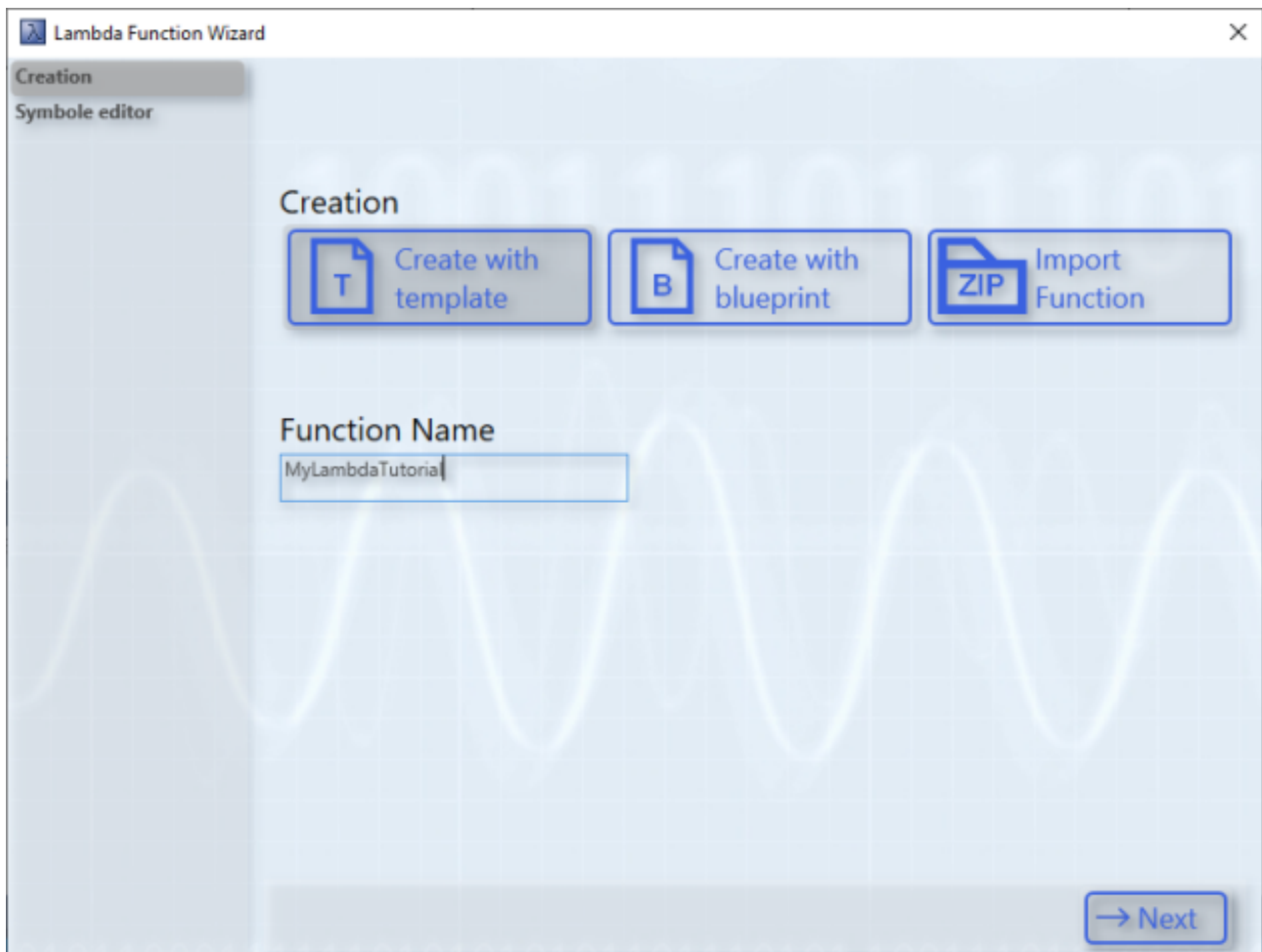
6.5.11.1.3 Lambda Template



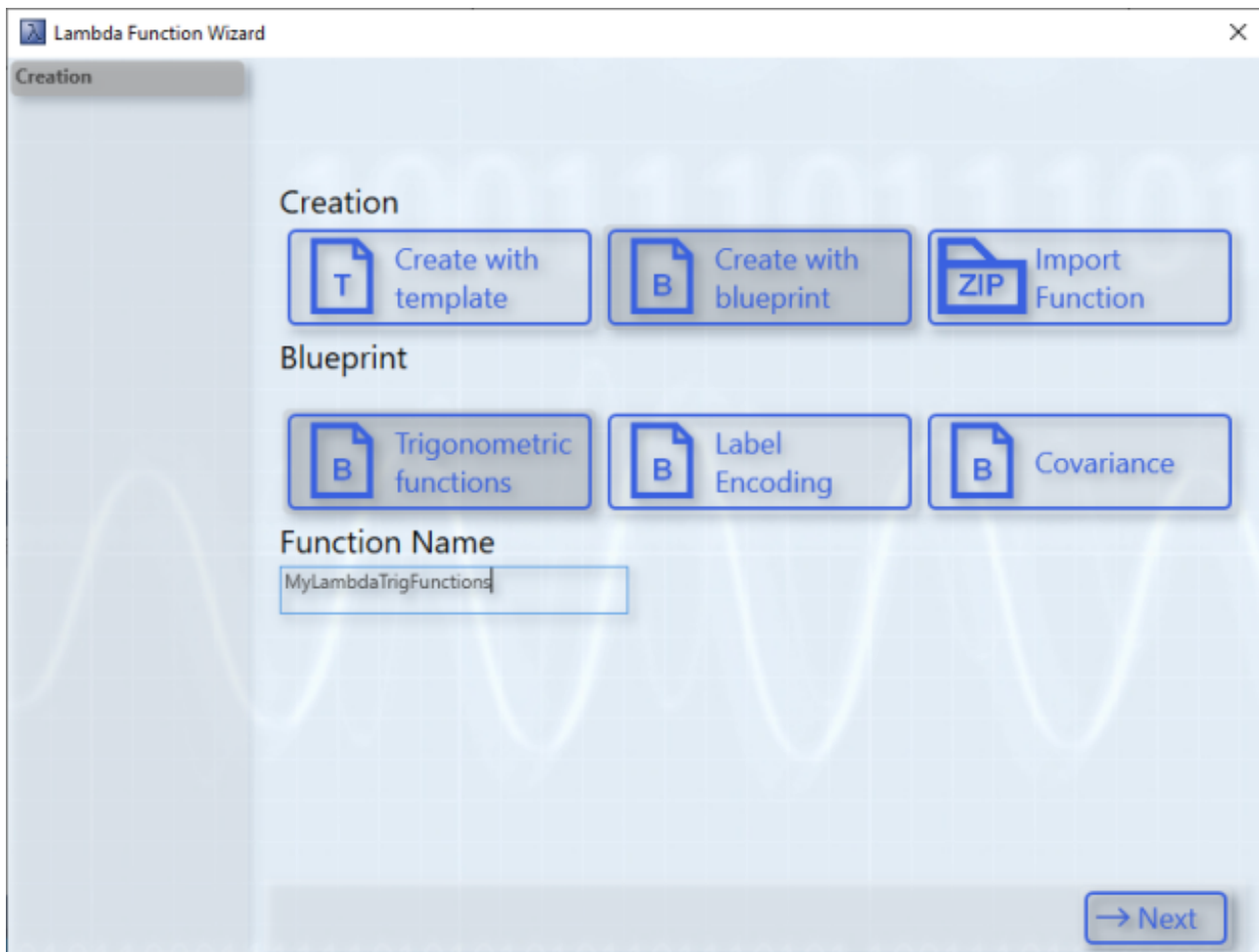
Das Lambda Template leitet durch die Entwicklung einer Lambda Funktion. Innerhalb des Parameter-Bereichs befindet sich ein Control. Dieses Control enthält vier Buttons zum Konfigurieren, zum Programmieren und zum Publishen der Lambda Funktion. Der vierte Button ist zum Einsehen der Log-Dateien des Entwicklungsprozesses. 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](#) [[▶ 263](#)]).

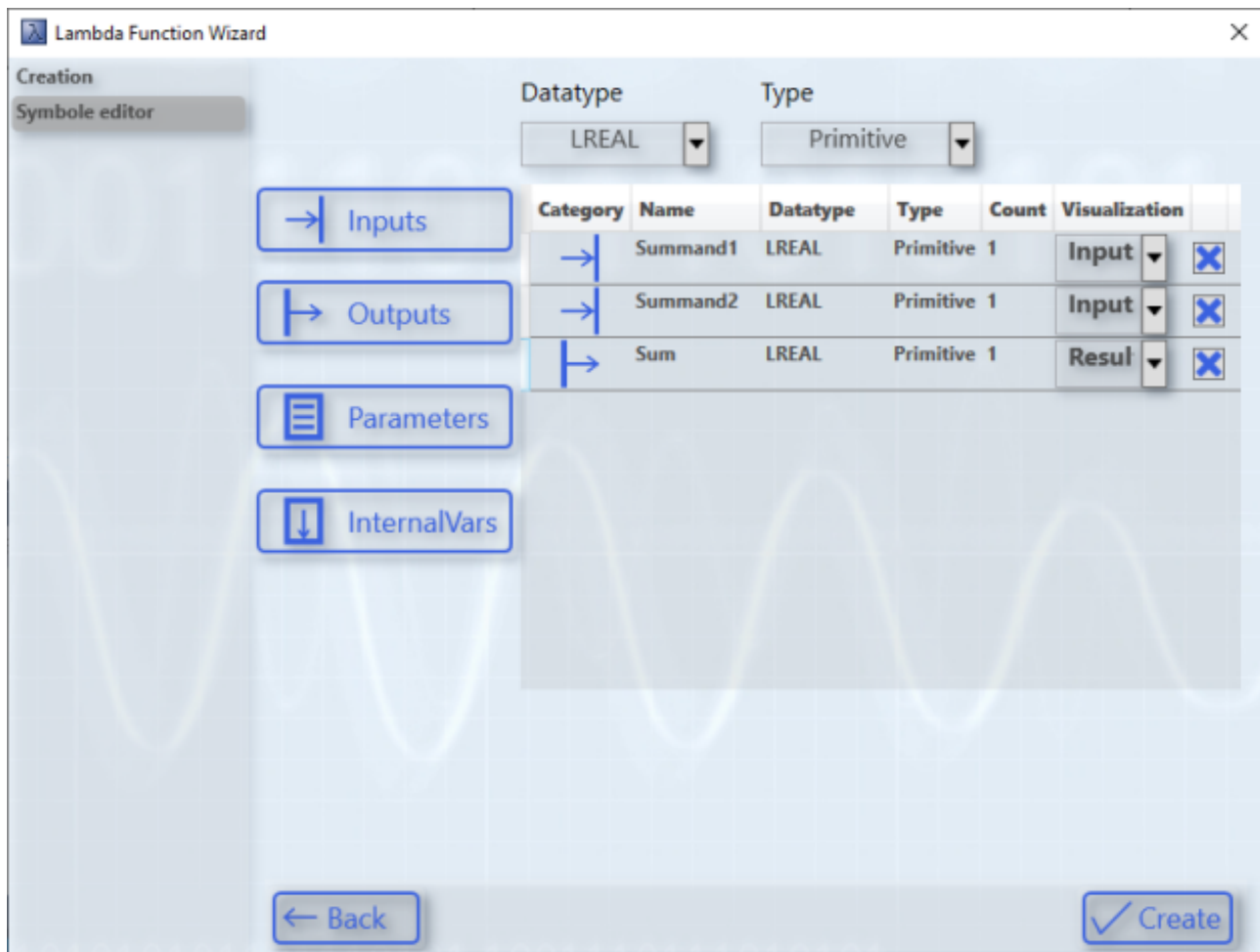
Über den dritten Modus importieren Sie exportierte Lambda Funktionen. Mit einem Klick auf den Button **Import Function**, öffnet sich ein Datei-Öffnen-Dialog. Anschließend können Sie die exportierte Lambda Funktion, die als ZIP-Archiv vorliegen sollte, auswählen. Sobald die Lambda Funktion importiert 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.

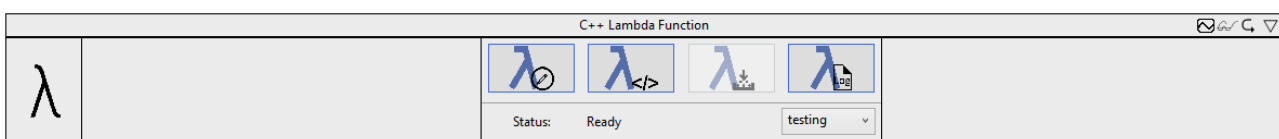
Wenn der Datenstrukturtyp Array bei einer Variablen ausgewählt ist, kann in der Spalte Count der Variable die Größe des Arrays angegeben werden. Um mehrdimensionale Arrays zu erstellen, müssen die Dimensionen mit dem Buchstaben „x“ separiert werden (Beispiel: 2x4).

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:

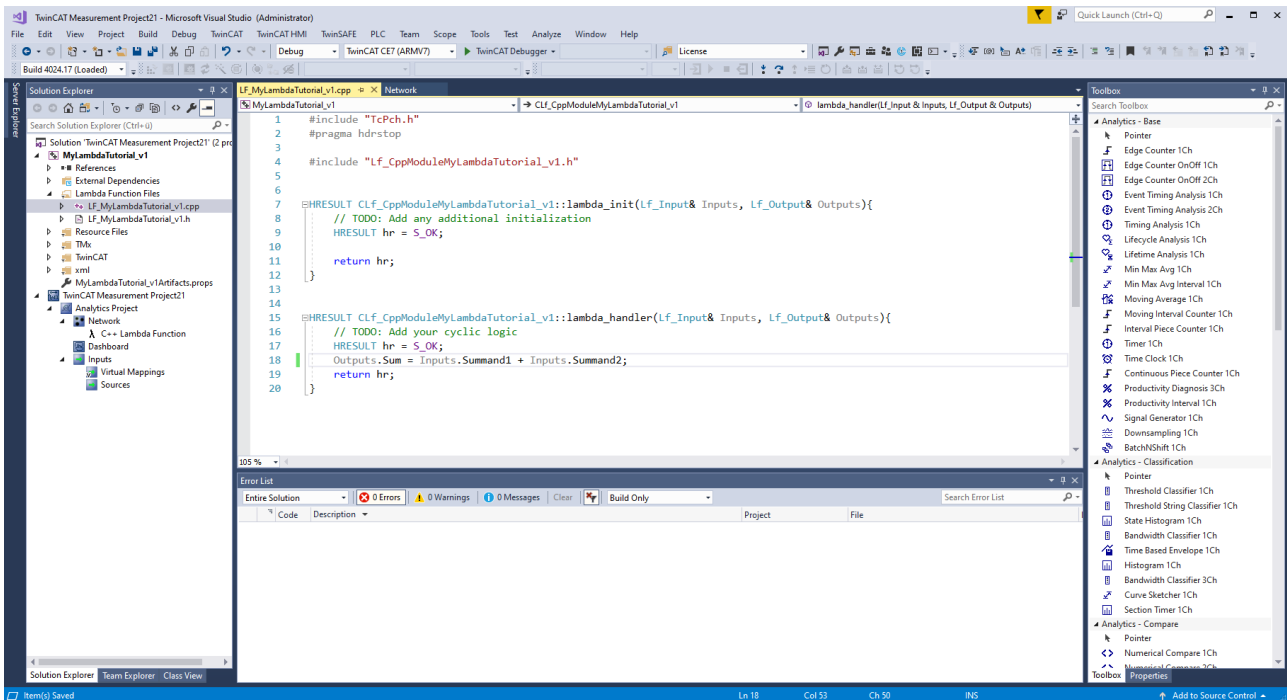
Datentyp	Visualisierung	Beispiel
STRING(255)	String (Input)	
	String (Output)	Output String String Result
	String (Parameter)	String Level 1 String Result
	FileOpen/FileSave (Parameter)	File Path twincat3.1\Boot\Teach.tas ...
ULINT	Timestamp (Output)	Last Event 15.11.2021 11:14:57.048
BOOL	Boolean (Input)	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 mehrere 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.

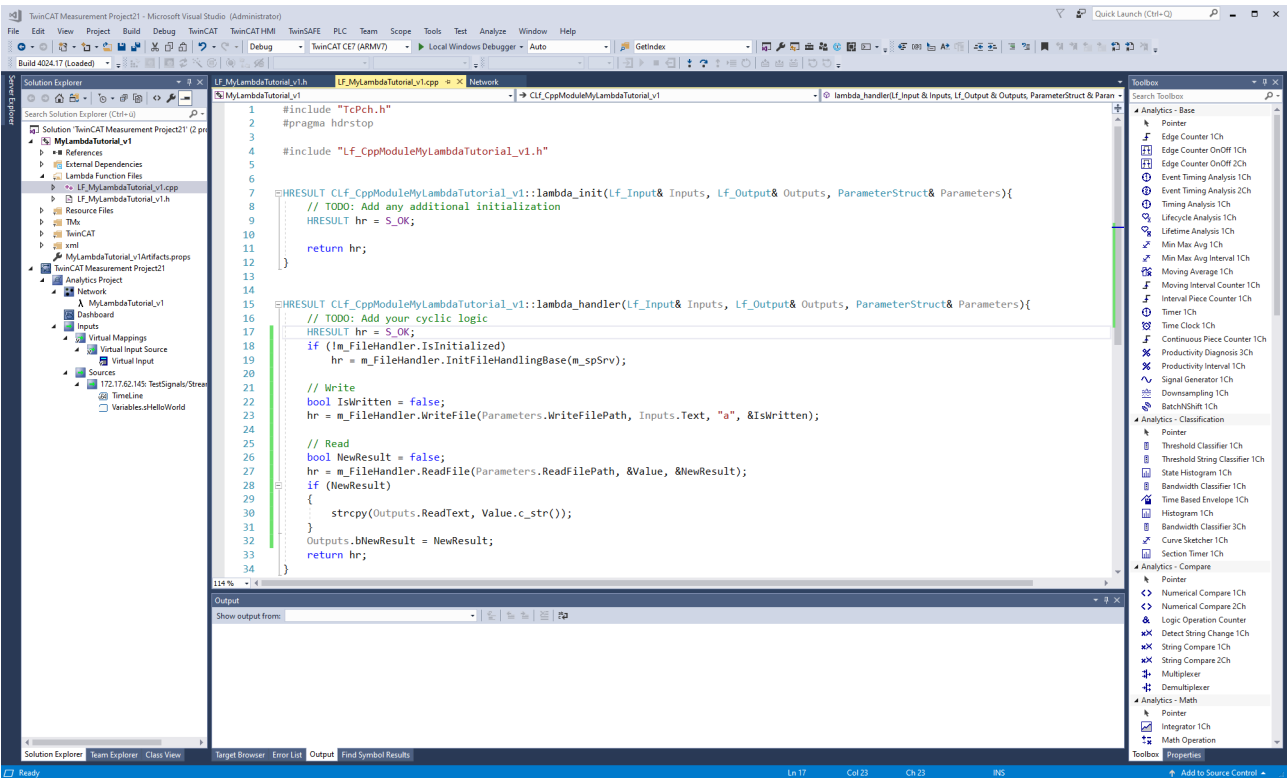
Kategorie	Symbolname	Datentyp	Beschreibung	Sichtbarkeit
Input	tTimestamp	ULINT	Die Variable enthält den TwinCAT Zeitstempel.	Versteckt
	bReset	BOOL	Die Variable setzt die konfigurierten Symbole zurück.	Optional
	bEnableExecution	BOOL	Die Variable aktiviert die lambda_handler-Methode.	Optional
Output	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* blsWritten	Hresult

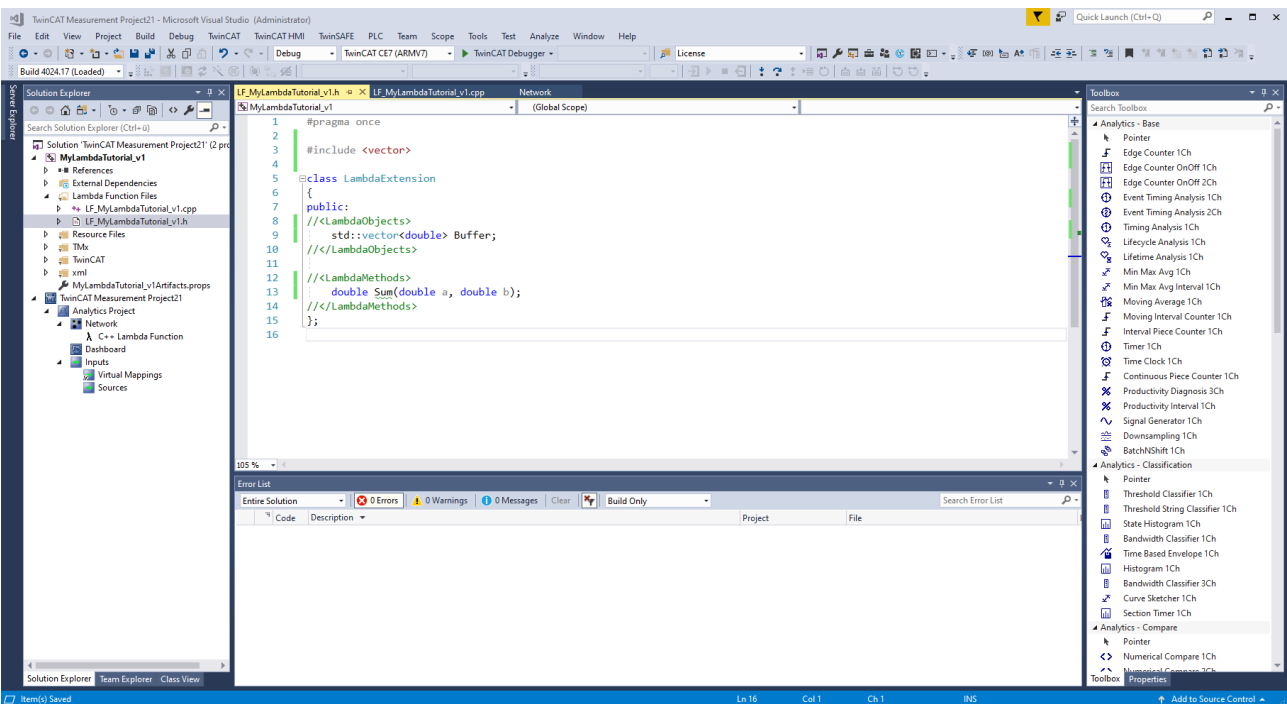
In der TwinCAT Analytics Workbench ist der Dateizugriff synchron, wohingegen er in der Analytics Runtime asynchron ist. Diese Besonderheit wird in den Methoden automatisch unterschieden. Bei einem asynchronen Lesezugriff liefert die Methode nicht im selben Zyklus den gelesenen Wert. Erst sobald der Parameter 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 blsWritten 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:

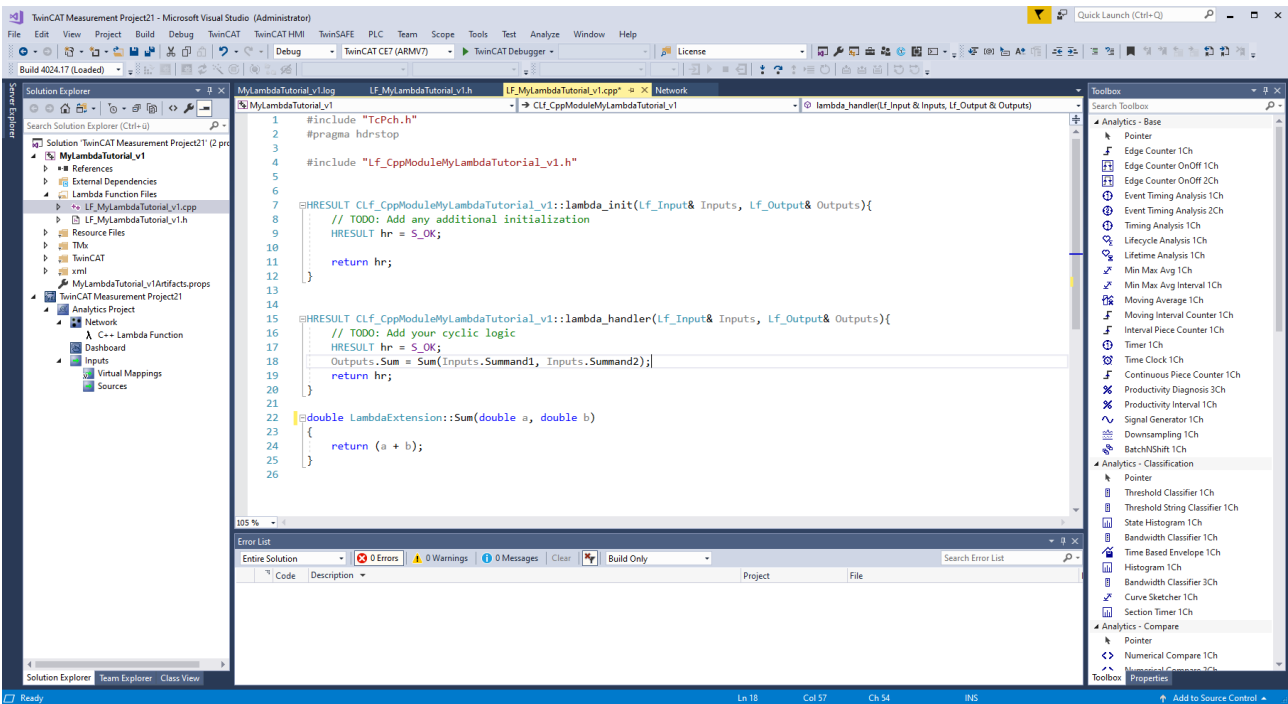


Erstellen Sie das Projekt wie gewohnt in Visual Studio®, um die Syntax zu prüfen.

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.

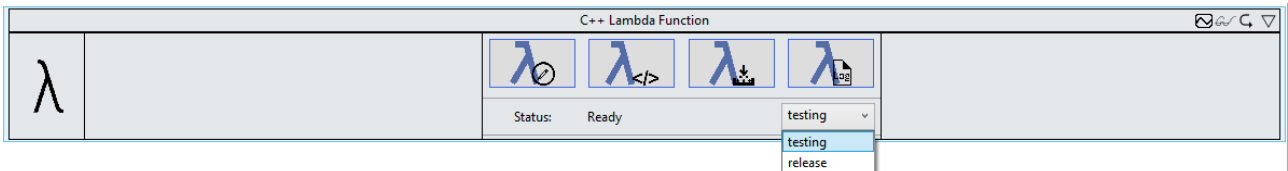


Nachdem die Logik integriert wurde, müssen Sie das Projekt speichern. Wechseln Sie danach zum Lambda Template im Analytics Projekt zurück.

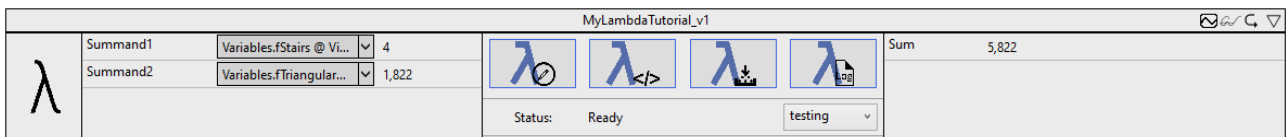
Falls ausreichend Know-how im Umgang mit TcCOM-Modulen vorhanden ist, kann das C++-Projekt individuell erweitert werden. Bei unsachgemäßer Verwendung der TcCOM-Module, können bei nachfolgenden Schritten der Lambda Funktionen Fehler auftreten. Weitere Informationen zu der Entwicklung von TcCOM-Modulen sind unter dem folgenden Link zu finden [TwinCAT C++ Entwicklung](#).

Publizieren der Lambda Funktion

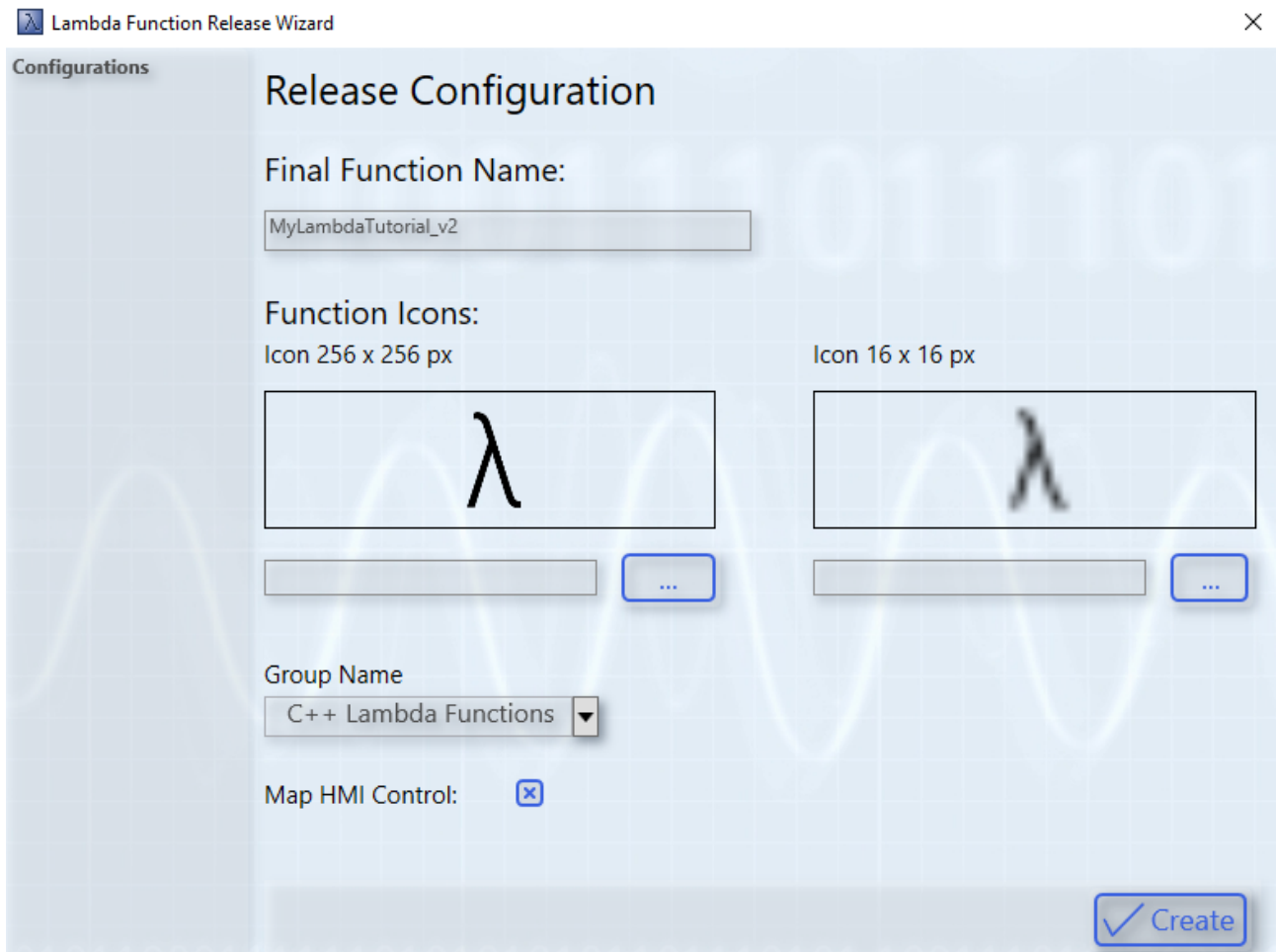
Vor dem Publizieren können Sie den Typ der Lambda Funktion über das Drop-Down-Menü unten rechts konfigurieren.



Eine Lambda-Funktion von Typ „testing“ enthält das Control mit den vier Buttons zum Entwickeln von Lambda Funktionen. Dieser Typ bietet sich während der Entwicklungs- 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 Mapping Wizard). Über den Create-Button generieren Sie die Lambda Funktion.

MyLambdaTutorial_v2			
λ	Summand1	<Empty>	-
	Summand2	<Empty>	-
			Sum Empty

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.

6.5.11.1.4 Lambda Blueprints

Trigonometric functions

TrigonometricFunctions			
λ	DataIn	<Empty>	-
	Trig_function	sin	
			DataOut Empty

Der Blueprint **Trigonometric 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 trigonometrischen 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

LabelEncoding			
λ	Message	<Empty>	-
	EncodedMessage	Empty	
	CategoryCount	Empty	

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

Covariance			
λ	xi	<Empty>	-
	yi	<Empty>	-
	WindowSize	1000	
	Cov	Empty	

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

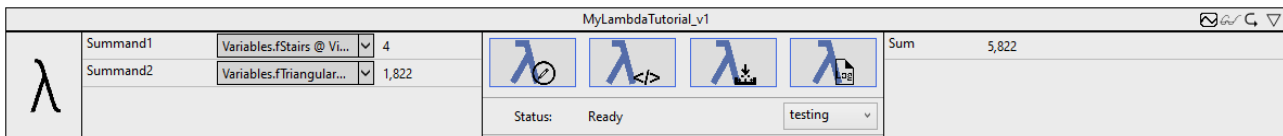
- **WindowSize:** Dieser Parameter gibt die Anzahl der Zyklen an, über die die Kovarianz berechnet wird. In dem Blueprint ist ein Sliding Window implementiert. Dabei fließen die aktuellen Werte in die Analyse ein und es werden mit jedem Zyklus die Ausgänge aktualisiert. Die Größe des Router-Speichers sollte beim Setzen dieses Parameters berücksichtigt werden.

Ausgangswerte

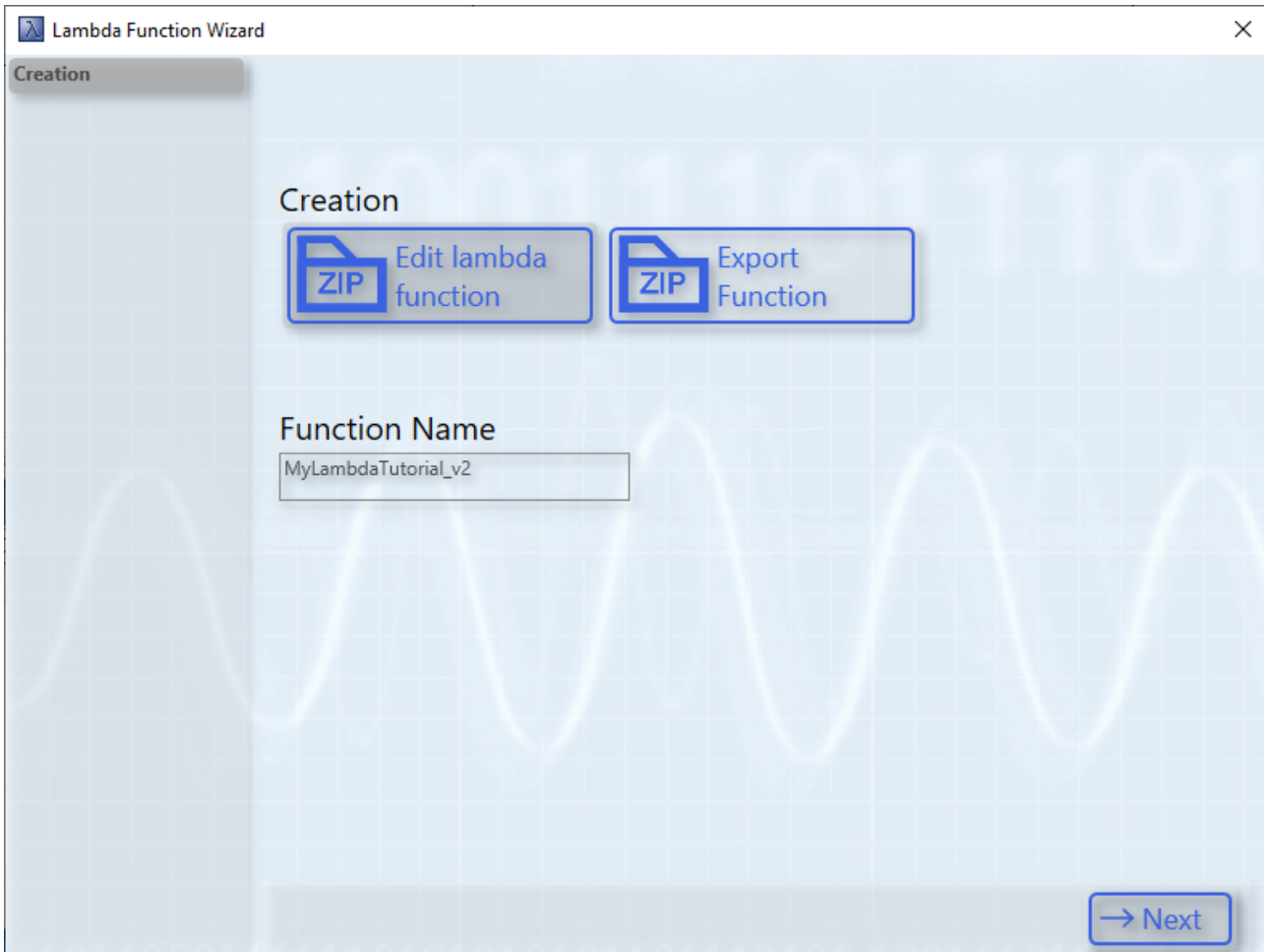
- **Cov:** Dieser Ausgangskanal gibt die Kovarianz der beiden Eingangskanäle über die konfigurierte Fenstergröße an.

6.5.11.1.5 Lambda Function

In der Entwicklungs- und Testphase sollten Lambda Funktionen als Testing-Versionen gepubliziert werden. Diese enthalten die konfigurierten Symbole, sowie das Lambda-Control mit den vier Buttons.

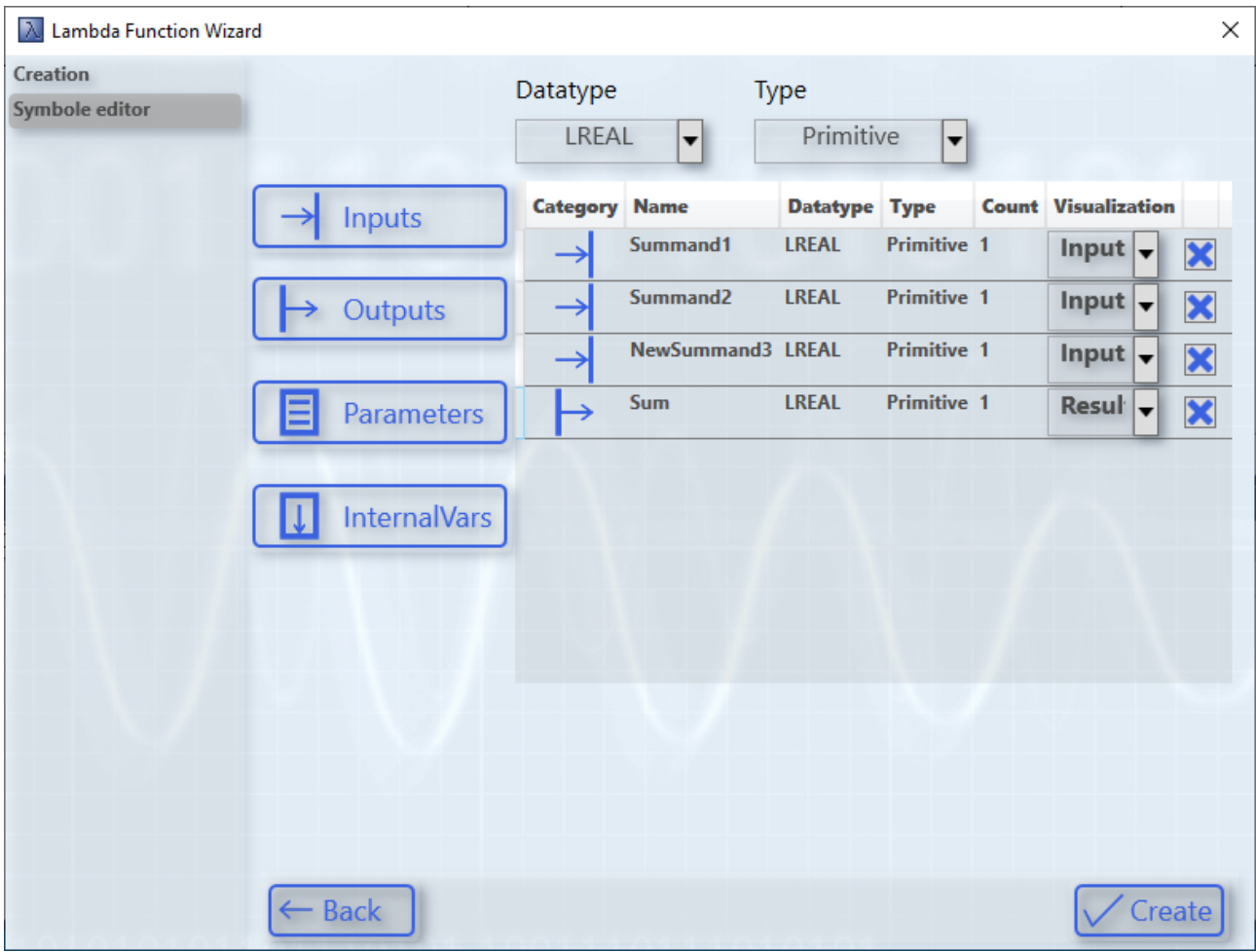


Diese Lambda Funktion kann wie jeder andere Algorithmus in der Analytics Workbench getestet werden. Sollten Änderungen von Nöten sein, leitet das Lambda-Control durch die Weiterentwicklung. Nach dem Klicken des ersten Buttons öffnet sich ein Wizard zum Konfigurieren der Lambda Funktion. Der Wizard für bestehende Lambda Funktionen bietet zwei Modi.

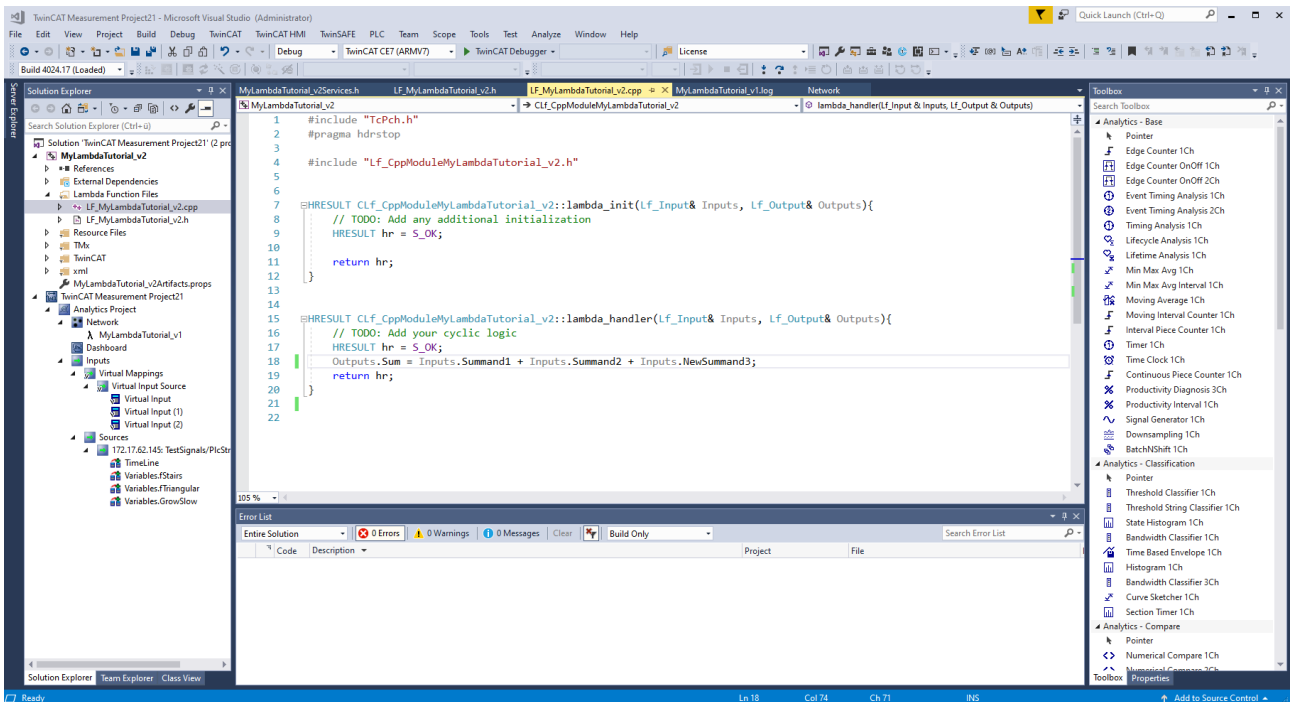


Der Button **Export Function** bietet die Möglichkeit Lambda Funktionen zu exportieren. Dazu wird das C++-Projekt und die Treiber der Lambda Funktion in einem Zip-Archiv gespeichert. Über die Import-Funktion im Lambda Template kann die Lambda Funktion auf einem anderen System komfortabel importiert und 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 angehängen 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.

6.5.11.1.6 Weitere Schritte für die Lambda Funktionen

Runtime-Deployment

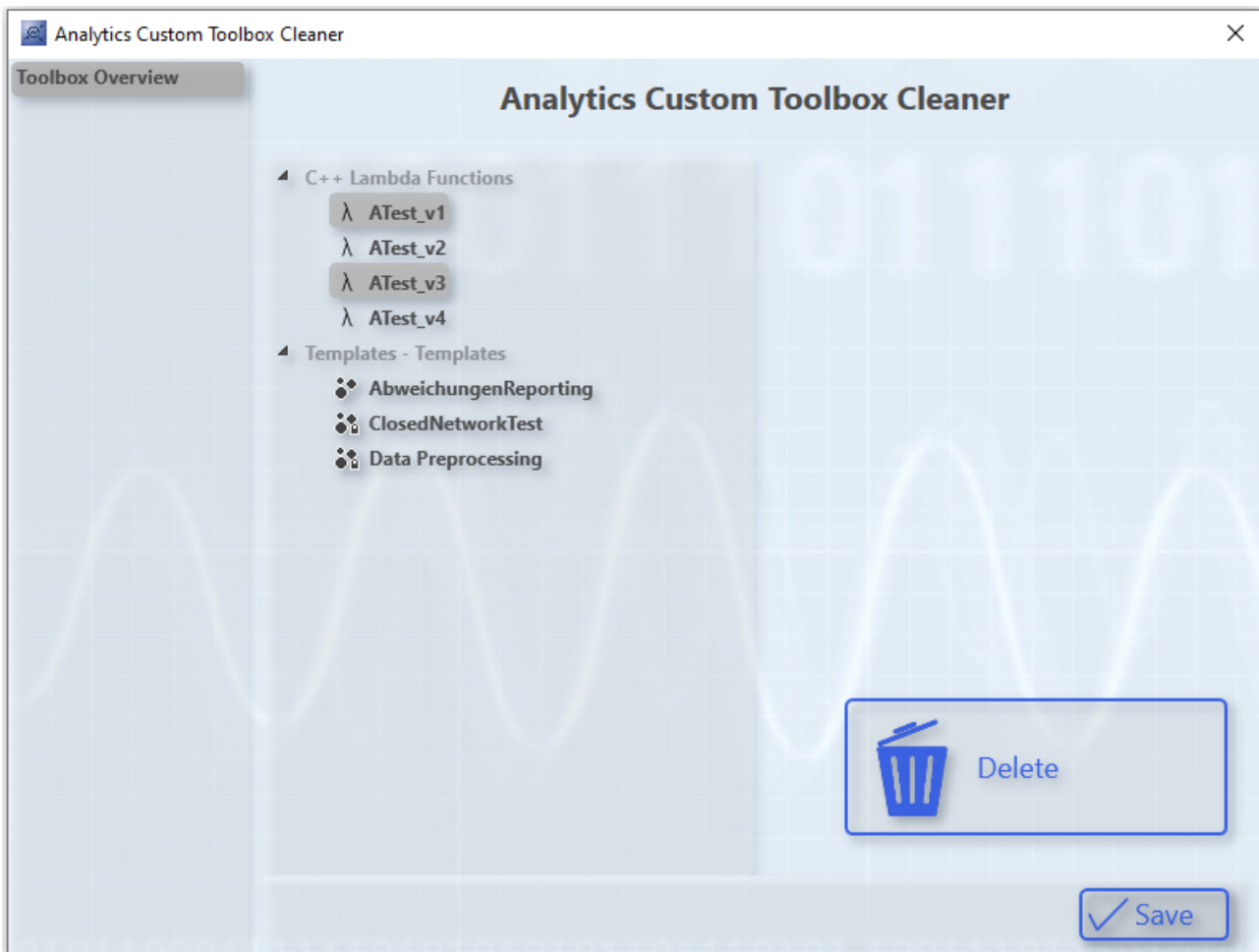
Auch die Lambda Funktionen sind für das Runtime-Deployment geeignet. Die entwickelte Lambda Funktion wird dafür als TcCOM-Modul in dem TwinCAT-Projekt angelegt. Ein SPS-Baustein übernimmt die Handhabung des TcCOM-Moduls. Der SPS-Baustein ruft dafür die entsprechenden Schnittstellen-Methoden der Lambda 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).

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-Knoten).

6.5.11.1.7 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.

6.5.11.2 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	<i>Filter</i>	TF3680 TC3 Filter / TF3600 TC3 Condition Monitoring	
	PT1		PT1 [► 286]
	IIRCoeff		IIRCoeff [► 287]
	IIRSpec		IIRSpec [► 287]
	PT2		PT2 [► 288]
	PT3		PT3 [► 288]
	PTn		PTn [► 289]
	IIRSos		IIRSos [► 289]
	Notch		Notch [► 289]
	LeadLag		LeadLag [► 290]
	PT2oscillation		PT2oscillation [► 290]
	PTt		PTt [► 291]
	Median		Median [► 291]
	ActualValue		ActualValue [► 291]
	Gaussian		Gaussian [► 292]
TwinCAT Condition Monitoring	<i>Condition Monitoring</i>	TF3600 TC3 Condition Monitoring	
	ArgSort		ArgSort [► 271]
	Crest Factor		Crest Factor [► 272]
	Crest Factor Plus		Crest Factor Plus [► 272]
	DiscreteClassification		Discrete Classification [► 273]
	Envelope		Envelope [► 273]
	Envelope Spectrum		Envelope Spectrum [► 274]
	Fatigue Analysis		Fatigue Analysis [► 275]
	Instantaneous Frequency		Instantaneous Frequency [► 277]
	Instantaneous Phase		Instantaneous Phase [► 271]
	Integrated RMS		Integrated RMS [► 277]
	Magnitude Spectrum		Magnitude Spectrum [► 279]
	Multi-Band RMS		Multi Band RMS [► 280]
	Power Spectrum		Power Spectrum [► 281]
	RMS		RMS [► 282]
	Spike Energy Spectrum		Spike Energy Spectrum [► 282]
	Vibration Assessment		Vibration Assessment [► 284]
	Watch Upper Thresholds		Watch Upper Thresholds [► 285]

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.


6.5.11.2.1 Condition Monitoring

Die Bibliothek **TwinCAT Condition Monitoring** bietet mathematische Algorithmen zur Zustandsüberwachung von Maschinen und Anlagen.

Übersicht der Algorithmen

Algorithmus	Beschreibung	SPS Funktionsbaustein
ArgSort [▶ 271]	Sortiert die eingehenden Argumente.	FB_CMA_ArgSort
Crest Factor [▶ 272]	Berechnet für ein- und mehrkanalige Zeitreihen den Crest-Faktor für jeden Kanal.	FB_CMA_CrestFactor
Crest Factor Plus [▶ 272]	Berechnet für ein- und mehrkanalige Zeitreihen den Crest-Faktor Plus für jeden Kanal.	FB_CMA_CrestFactorPlus
Discrete Classification [▶ 273]	Klassifikation von mehrkanaligen Daten anhand konfigurierbarer Schwellwerte.	FB_CMA_DiscreteClassification
Envelope [▶ 273]	Berechnet die Einhüllende eines Zeitsignals.	FB_CMA_Envelope
Envelope Spectrum [▶ 274]	Berechnet das Einhüllendenspektrum eines Zeitsignals.	FB_CMA_EnvelopeSpectrum
Fatigue Analysis [▶ 275]	Lebensdaueranalyse und Schädigungsrechnung.	FB_CMA_RainflowCounting FB_CMA_MeanStressCorrection FB_CMA_MinersRule
Instantaneous Frequency [▶ 277]	Berechnung der Momentanfrequenz eines Zeitsignals.	FB_CMA_InstantaneousFrequency
Instantaneous Phase [▶ 271]	Berechnung der Momentanphase eines Zeitsignals.	FB_CMA_InstantaneousPhase
Integrated RMS [▶ 277]	Berechnet (optional integrierte) RMS Werte für ein- und mehrkanalige reell-wertige Zeitreihen.	FB_CMA_IntegratedRMS
Magnitude Spectrum [▶ 279]	Berechnet das Magnitudenspektrum (auch Betragsspektrum oder Amplitudenspektrum genannt) eines reell-wertigen Eingangssignals.	FB_CMA_MagnitudeSpectrum
Multi Band RMS [▶ 280]	Berechnet RMS Werte für ein- und mehrkanalige reell-wertige Zeitreihen für konfigurierbarer Frequenzbänder.	FB_CMA_MultiBandRMS
Power Spectrum [▶ 281]	Berechnung des Leistungsspektrums eines reell-wertigen Eingangssignals sowie optionale Skalierung zu Dezibel.	FB_CMA_PowerSpectrum
RMS [▶ 282]	Berechnet für ein- und mehrkanalige reell-wertige Signale den zeitlichen RMS Wert.	FB_CMA_RMS
Spike Energy Spectrum [▶ 282]	Analyse von Spitzenenergie hochfrequenter Signalanteile.	FB_CMA_SpikeEnergySpectrum
Vibration Assessment [▶ 284]	Schwingungsbeurteilung reell-wertiger Eingangssignale.	FB_CMA_VibrationAssessment
Watch Upper Thresholds [▶ 285]	Konfigurierbare Schwellwertüberwachung von mehrkanaligen Daten.	FB_CMA_WatchUpperThresholds

6.5.11.2.1 Instantaneous Phase

Instantaneous Phase			
	Input	FFT Length	512
	Input 00	Window Length	400
	Reset	Number of Channels	1
	Enable Execution	Phase Unwrap Method	eCM_ThresholdUnwrapping
	Phase Threshold	2,3E-308	
Output		Cnt Results	Cnt Results @ tccm_algorithm
		New Result	New Result @ tccm_algorithm

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.

6.5.11.2.2 ArgSort

ArgSort			
	Input	Input Length	200
	Input 00	Shift NaNs to End	<input checked="" type="checkbox"/>
	Reset	Sort Downward	<input type="checkbox"/>
	Enable Execution	Scale Factor	1
Output Data		Cnt Results	Cnt Results @ tccm_algorithm
Output Index		New Result	New Result @ tccm_algorithm

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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.3 Crest Factor

Crest Factor			
	Input		Number of Channels
	Input 00	<Empty>	1
	Reset	<Empty>	Buffer Length
	Enable Execution	<Empty>	250
			Decibel Threshold
		2,3E-308	Output
			Output 00
			Cnt Results
			<i>Cnt Results @ tccm_algorithm</i>
			New Result
			<i>New Result @ tccm_algorithm</i>

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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.4 Crest Factor Plus

Crest Factor Plus			
	Input		Number of Channels
	Input 00	<Empty>	1
	Reset	<Empty>	Buffer Length
	Enable Execution	<Empty>	250
			Transform to Decibel
		<input checked="" type="checkbox"/>	Output
		Config: [c1, c2, c3]	Output 00
		0;0;1	Cnt Results
		Decibel Threshold	<i>Cnt Results @ tccm_algorithm</i>
		2,3E-308	New Result
			<i>New Result @ tccm_algorithm</i>

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.5 Discrete Classification

Discrete Classification			
	Input		Number of Channels
	Input 00	<Empty>	1
	Reset	<Empty>	Number of SubChannels
	Enable Execution	<Empty>	257
			Max Number of Classes
			3
			Config: [classes]
			-1;0;1
			Cnt Results
			Cnt Results @ tccm_algorithm
			New Result
			New Result @ tccm_algorithm

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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.6 Envelope

Envelope			
	Input		FFT Length
	Input 00	<Empty>	512
	Reset	<Empty>	Window Length
	Enable Execution	<Empty>	400
			Number of Channels
			1
			Cnt Results
			Cnt Results @ tccm_algorithm
			New Result
			New Result @ tccm_algorithm

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 \[► 274\]](#).

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.

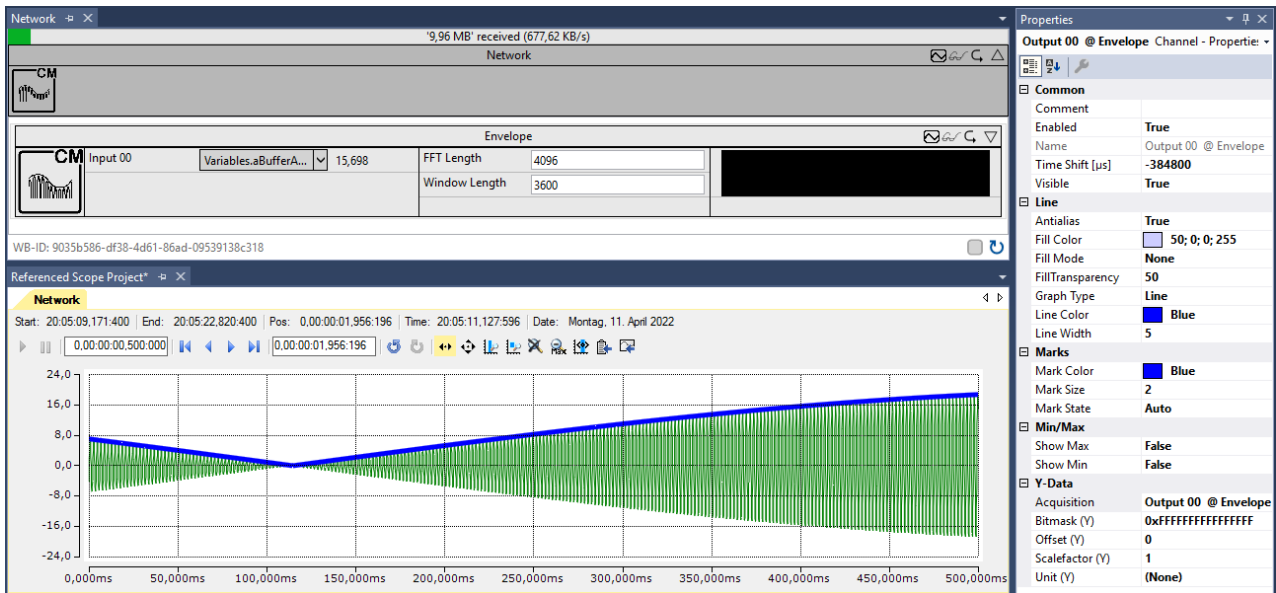
- **Window Length:** Ist die Länge des Analysefensters in Samples. Die Länge muss größer als eins und eine gerade Zahl sein.
- **Number of Channels:** Definiert die Anzahl von unabhängigen Kanälen. Diese muss größer als null sein.

Ausgangswerte

- **Output:** Ausgangsarray mit halber Fensterlänge, welches die berechnete Einhüllende enthält.
- **Cnt Results:** Gibt die Anzahl der berechneten Ausgangsarrays an.
- **New Result:** Ist ein boolescher 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 Ergebnis um $(nFFTLentgth + nWindowLength)/2$ Samples verzögert. Dies lässt sich mittels geeigneter Wahl des Time Shift am Kanal des Outputs korrigieren.



6.5.11.2.1.7 Envelope Spectrum

Envelope Spectrum		
Input	FFT Length	512
Input 00	Window Length	400
Reset	Window Type	eCM_HannWindow
Enable Execution	Scaling Type	eCM_DiracScaling
	Transform to Decibel	<input type="checkbox"/>
	Number of Channels	1
	Window Overlap	200
	Decibel Threshold	2,3E-308
	Use Recommended Overlap	<input checked="" type="checkbox"/>
Window Parameters		
Window Parameters 00		2,5
Window Parameters 01		1
Window Parameters 02		1
Window Parameters 03		1
Window Parameters 04		1
Output	Output 00	
	Cnt Results	Cnt Results @ tccm_algorithm
	New Result	New Result @ tccm_algorithm

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.8 Fatigue Analysis

Fatigue Analysis		
Input	Number of Stress Range Bins	100
Input 00	Minimum Stress Range	0
Enable Execution	Maximum Stress Range	100
Reset	Number of Mean Stress Bins	100
	Minimum negative Mean Stress	-50
	Maximum positive Mean Stress	50
	Mean Stress Correction	eCM_Goodman
	Use UTS Correction	<input checked="" type="checkbox"/>
	UTS	700
	K1	3
	K2	5
	SRI	350
	NC1	100
	NC2	100000
	Woehler Curve	
	Additional Damage	0
	Additional Halfcycles	0
	Input Length	200
	Damage	Damage @ tccm_algorithm
	HalfCycles	HalfCycles @ tccm_algorithm
	Range Bins	
	Mean Bins	
	Cnt Results	Cnt Results @ tccm_algorithm
	New Result	New Result @ tccm_algorithm

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 $N = 1 \dots NC1$.
- **K2:** Definiert den Gradienten der Wöhlerkurve startend bei $N \geq NC2$.
- **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 Anzahl 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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.9 Instantaneous Frequency

Instantaneous Frequency			
	Input	FFT length	512
	Input 00	Window Length	400
	Reset	Sample Rate	10000
	Enable Execution	Number of Channels	1
	Magnitude Threshold	2,3E-308	
Output		Cnt Results	Cnt Results @ tccm_algorithm
		New Result	New Result @ tccm_algorithm

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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.10 Integrated RMS

Integrated RMS			
	Input	FFT Length	512
	Input 00	Window Length	400
	Reset	Sample Rate	20000
	Enable Execution	Integration Order	2
		Window Type	eCM_HannWindow
		Lower Frequency Limit	20
		Upper Frequency Limit	1000
		Max Frequency Bands	1
		Number of Channels	1
		Window Overlap	200
		Use recommended Overlap	<input checked="" type="checkbox"/>
		Transform to Decibel	<input checked="" type="checkbox"/>
		Decibel Threshold	2,3E-308
		Config: Bands x [lower, upper]	20;1000
		aWindowParameters	
	aWindowParameters 00	2,5	
	aWindowParameters 01	1	
	aWindowParameters 02	1	
	aWindowParameters 03	1	
	aWindowParameters 04	1	
Ch ^Order		Cnt Results	Cnt Results @ tccm_algorithm
Ch 00 Order 00		New Result	New Result @ tccm_algorithm
Ch 00 Order 01			
Ch 00 Order 02			

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.11 Magnitude Spectrum

Magnitude Spectrum			
Input	FFT Length	512	Output
Input 00 <Empty>	Window Length	400	Output 00
Reset <Empty>	Window Type	eCM_HannWindow	Cnt Results Cnt Results @ tccm_algorithm
Enable Execution <Empty>	Scaling Type	eCM_PeakAmplitude	New Result New Result @ tccm_algorithm
	Transform to Decibel	<input type="checkbox"/>	
	Number of Channels	1	
	Window Overlap	200	
	Decibel Threshold	2,3E-308	
	Use Recommended Overlap	<input checked="" type="checkbox"/>	
	Window Parameters		
	Window Parameters 00	2,5	
	Window Parameters 01	1	
	Window Parameters 02	1	
	Window Parameters 03	1	
Window Parameters 04	1		

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.12 Multi Band RMS

Multi-Band RMS		
Input	FFT Length	512
Input 00 <Empty>	Window Length	400
Reset <Empty>	Max Number of Bands	2
Enable Execution <Empty>	Window Type	eCM_HannWindow
	Config: MaxBands x [lower,upper]	10;1000;2;2000
	Sample Rate	20000
	Number of Channels	1
	Use recommended Overlap	<input checked="" type="checkbox"/>
	Window Overlap	200
	Transform To Decibel	<input checked="" type="checkbox"/>
	Decibel Threshold	2,3E-308
	Window Parameters	
	Window Parameters 00	2,5
	Window Parameters 01	1
	Window Parameters 02	1
	Window Parameters 03	1
	Window Parameters 04	1
	Channel^Band	
	Channel 00 Band 00	
	Channel 00 Band 01	
	Cnt Results	Cnt Results @ tccm_algorithm
	New Result	New Result @ tccm_algorithm

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.13 Power Spectrum

Power Spectrum			
Input	FFT Length	512	Output
Input 00 <Empty>	Window Length	400	Output 00
Reset <Empty>	Window Type	eCM_HannWindow	
Enable Execution <Empty>	Scaling Type	eCM_PeakAmplitude	Cnt Results Cnt Results @ tccm_algorithm
	Transform to Decibel	<input type="checkbox"/>	New Result New Result @ tccm_algorithm
	Number of Channels	1	
	Window Overlap	200	
	Decibel Threshold	2,3E-308	
	Use Recommended Overlap	<input checked="" type="checkbox"/>	
	Window Parameters		
	Window Parameters 00	2,5	
	Window Parameters 01	1	
	Window Parameters 02	1	
	Window Parameters 03	1	
	Window Parameters 04	1	

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.14 RMS

		RMS				
<div style="border: 1px solid black; padding: 2px; text-align: center;"> CM RMS </div>	Input		Number of Channels	1	Output	
	Input 00	<Empty>	Buffer Length	2000	Output 00	
	Reset	<Empty>	Transform to Decibel	<input checked="" type="checkbox"/>	Cnt Results	<i>Cnt Results @ tccm_algorithm</i>
	Enable Execution	<Empty>	Decibel Threshold	2,3E-308	New Result	<i>New Result @ tccm_algorithm</i>

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.15 Spike Energy Spectrum

		Spike Energy Spectrum				
<div style="border: 1px solid black; padding: 2px; text-align: center;"> CM </div>	Input	FFT Length	1024	Output		
	Input 00	<Empty>	Window Length	800	Output 00	
	Reset	<Empty>	Sample Rate	10000		
	Enable Execution	<Empty>	IIR Filter Order	4	Cnt Results	<i>Cnt Results @ tccm_algorithm</i>
			Window Type	eCM_HannWindow	New Result	<i>New Result @ tccm_algorithm</i>
			Scaling Type	eCM_RootPowerSum		
			Number of Channels	1		
			Transform to Decibel	<input type="checkbox"/>		
			Window Overlap	400		
			Decibel Threshold	2,3E-308		
			Use Recommended Overlap	<input checked="" type="checkbox"/>		
			Config: Ch x [decay, f min, f max]	0;0;0		
			Window Parameters			
			Window Parameters 00	2,5		
		Window Parameters 01	1			
		Window Parameters 02	1			
		Window Parameters 03	1			
		Window Parameters 04	1			

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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' $> 1 / \text{'Fehlerfrequenz'}$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.16 Vibration Assessment

Vibration Assessment		
Input	Number of Channels	2
Input 01	Max Integration Order	2
Input 02	Max Number of Bands	1
Reset	Max Number of Classes	3
Enable Execution	FFT Length	2048
	Window Length	1600
	Sample Rate	10000
	Memorize Classification	<input type="checkbox"/>
	Window Type	eCM_HannWindow
	Window Overlap	800
	Transform to Decibel	<input type="checkbox"/>
	Decibel Threshold	2,3E-308
	Config: (Order+ 1) x MaxClasses	1000000;1000000;1000000;0,0023;0,0045;0,0071;2,9E-05;5,7E-05;9E-05
	Config: MaxBands x [f min, f max]	10;1000
	Use Recommended Overlap	<input checked="" type="checkbox"/>
	Window Parameters	
	Window Parameters 00	2,5
	Window Parameters 01	1
	Window Parameters 02	1
	Window Parameters 03	1
	Window Parameters 04	1
Output	Classification	
	Integration Order	
	Channel	
	Cnt Results	Cnt Results @ tccm_algorithm
	New Result	New Result @ tccm_algorithm

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 $x \rightarrow 20 * \log_{10}(x)$.
- **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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.

6.5.11.2.1.17 Watch Upper Thresholds

Watch Upper Thresholds					
	Input		Number of Channels	1	Output
	Input 00	<Empty>	Number of SubChannels	257	Output [Class;SubCh] 00
	Reset	<Empty>	Max Number of Classes	3	Cnt Results Cnt Results @ tccm_algorithm
	Enable Execution	<Empty>	Config: [Classes]	-1;0;1	New Result New Result @ tccm_algorithm
			Memorize Classification	<input checked="" type="checkbox"/>	

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 boolescher Wert, der angibt, ob im aktuellen Zyklus ein neues Ergebnis erhalten wurde.


6.5.11.2.2 Filter

Die Bibliothek **TwinCAT Filter** bietet Algorithmen zur Realisierung von Digitalfiltern.

Übersicht der Algorithmen

Algorithmus	Beschreibung	SPS Funktionsbaustein
PT1 [▶ 286]	Realisierung eines Verzögerungsglieds 1. Ordnung (PT1-Glied).	FB_FTR_PT1
IIRCoeff [▶ 287]	Realisierung eines IIR-Filters (Infinite-Impulse-Response) über Filterkoeffizienten.	FB_FTR_IIRCoeff
IIRSpec [▶ 287]	Realisierung eines IIR-Filters (Infinite-Impulse-Response) über Filterspezifikationen.	FB_FTR_IIRSpec
PT2 [▶ 288]	Realisierung eines Verzögerungsglieds 2. Ordnung (PT2-Glied).	FB_FTR_PT2
PT3 [▶ 288]	Realisierung eines Verzögerungsglieds 3. Ordnung (PT3-Glied).	FB_FTR_PT3
PTn [▶ 289]	Realisierung eines Verzögerungsglieds n-ter Ordnung mit gleichen Zeitkonstanten.	FB_FTR_PTn
IIRSos [▶ 289]	Realisierung eines IIR-Filters (Infinite-Impulse-Response) über Filterkoeffizienten.	FB_FTR_IIRSos
Notch [▶ 289]	Realisierung eines Bandstopp-Filters mit schmaler Bandbreite.	FB_FTR_Notch
LeadLag [▶ 290]	Realisierung eines Phasenkorrekturglieds 1. Ordnung.	FB_FTR_LeadLag
PT2oscillation [▶ 290]	Realisierung eines schwingungsfähigen Verzögerungsglieds 2. Ordnung.	FB_FTR_PT2oscillation
PTt [▶ 291]	Realisierung eines Totzeitglieds (PTt-Glied).	FB_FTR_PTt
Median [▶ 291]	Realisierung eines Median Filters.	FB_FTR_Median
ActualValue [▶ 291]	Realisierung einer Plausibilitätskontrolle und Filterung einer gemessenen Eingangsgröße.	FB_FTR_ActualValue
Gaussian [▶ 292]	Realisierung eines Gauß-Filters.	FB_FTR_Gaussian

6.5.11.2.2.1 PT1

PT1			
	Input	SamplingRate	1000
	Input	Kp	1
	Enable Execution	T1	0.006366
	Reset	Initial Values	[0]
		Output	Empty

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:** Abtastezeit 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.

6.5.11.2.2 IIRCoeff

IIRCoeff			
	Input		Coefficient A [1, -0.993736471541]
	Input	<Empty>	Coefficient B [0.003131764, 0.003131764]
	Enable Execution	<Empty>	Initial Values [0]
	Reset	<Empty>	
		Output Empty	

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) $[a_0, a_1, a_2, \dots, a_N]$ sind frei wählbar.
- **Coefficient B:** Die Koeffizienten b_k (Zähler) $[b_0, b_1, b_2, \dots, b_M]$ sind frei wählbar.
- **Initial Values (optional):** Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output:** Manipuliertes Ausgangssignal.

6.5.11.2.3 IIRSpec

IIRSpec			
	Input		FilterName eButterworth
	Input	<Empty>	FilterType eLowPass
	Enable Execution	<Empty>	FilterOrder 1
	Reset	<Empty>	SamplingRate 1000
			Cutoff 100
			Bandwidth 100
			PassBandRipple 0.1
			Initial Values [0]
			Output Empty

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.

6.5.11.2.2.4 PT2

PT2					
	Input		SamplingRate	1000	Output <i>Empty</i>
	Input	<Empty>	Kp	1	
	Enable Execution	<Empty>	T1	0.006366	
	Reset	<Empty>	T2	0.006366	
			Initial Values	[0]	

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.

6.5.11.2.2.5 PT3

PT3					
	Input		SamplingRate	1000	Output <i>Empty</i>
	Input	<Empty>	Kp	1	
	Enable Execution	<Empty>	T1	0.006366	
	Reset	<Empty>	T2	0.006366	
			T3	0.006366	
		Initial Values	[0]		

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.

6.5.11.2.2.6 PTn

PTn						
	Input		SamplingRate	1000	Output	Empty
	Input	<Empty>	Kp	1		
	Enable Execution	<Empty>	T1	0.006366		
	Reset	<Empty>	Order	1		
			Initial Values	[0]		

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.

6.5.11.2.2.7 IIRSos

IIRSos						
	Input		Coefficient Sos	[0.001568, 0.001568, 0, 1, -0.99686, 0]	Output	Empty
	Input	<Empty>	Initial Values	[0]		
	Enable Execution	<Empty>				
	Reset	<Empty>				

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 $[b_{01}, b_{11}, b_{21}, a_{01}, a_{11}, a_{21}, b_{02}, b_{12}, b_{22}, a_{02}, a_{12}, a_{22}, \dots, b_{0M}, b_{1M}, b_{2M}, a_{0M}, a_{1M}, a_{2M}]$ sind frei wählbar.
- **Initial Values (optional):** Initialwerte definieren den internen Zustand des Filters.

Ausgangswerte

- **Output:** Manipuliertes Ausgangssignal.

6.5.11.2.2.8 Notch

Notch						
	Input		SamplingRate	1000	Output	Empty
	Input	<Empty>	NotchFrequency	100		
	Enable Execution	<Empty>	Q	30		
	Reset	<Empty>	Initial Values	[0]		

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.

6.5.11.2.2.9 LeadLag

		LeadLag			
	Input		SamplingRate	1000	Output
	Input	<Empty>	T1	0.0015915	Empty
	Enable Execution	<Empty>	T2	0.0031831	
	Reset	<Empty>	Low Pass Frequency	50	
			High Pass Frequency	100	
		Initial Values	[0]		

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.

6.5.11.2.2.10 PT2oscillation

		PT2oscillation			
	Input		SamplingRate	1000	Output
	Input	<Empty>	Kp	1	Empty
	Enable Execution	<Empty>	T1	0.0015915	
	Reset	<Empty>	Theta	0.5	
			Cutoff Frequency	100	
		Initial Values	[0]		

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.

6.5.11.2.2.11 PTt

PTt				
	Input		SamplingRate 1000	Output Empty
	Input	<Empty>	Kp 1	
	Enable Execution	<Empty>	Tt 0.1	
	Reset	<Empty>		

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.

6.5.11.2.2.12 Median

Median				
	Input		SamplesToFilter 2	Output Empty
	Input	<Empty>	Initial Values [0]	
	Enable Execution	<Empty>		
	Reset	<Empty>		

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.

6.5.11.2.2.13 ActualValue

ActualValue				
	Input		DeltaMax 1	Output Empty
	Input	<Empty>	Initial Values [0]	FilterActive Empty
	Enable Execution	<Empty>		
	Reset	<Empty>		

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.

6.5.11.2.2.14 Gaussian

Gaussian				
	Input	<Empty>	-	
	Enable Execution	<Empty>	-	
	Reset	<Empty>	-	
	Calculate SamplesToFilter automaticly			<input checked="" type="checkbox"/>
	Output			Empty
	SamplingRate	1000		
	Cutoff	100		
	SamplesToFilter	11		
	Initial Values	[0]		

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 $\text{SamplingRate}/2$).
- **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.

6.5.12 Analytics - Visualization Only

Die Algorithmen der Kategorie *Visualization Only* stellen Hilfsbausteine dar, die Funktionen für die spätere Visualisierung im Kontext des HMI-Dashboards bieten.

6.5.12.1 Trend Line

Trend Line			
	Channel 00	<Empty>	-
	Num Channels	+ -	
	Channel Name 00	Channel 1	

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.

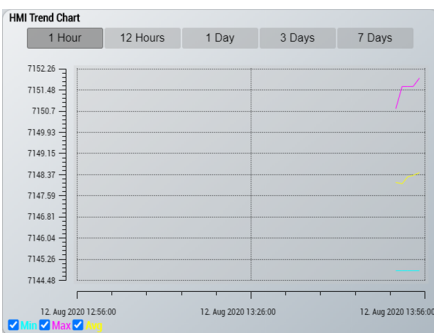
Konfigurationsoptionen

- **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.

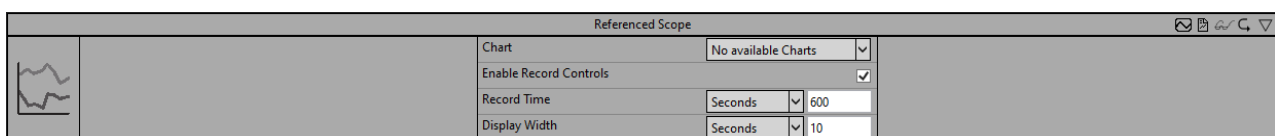


Dem Baustein können keine weiteren HMI Controls hinzugefügt werden, da dieser ausschließlich zur Darstellung des Trend Line Controls gedacht ist.

6.5.12.2 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

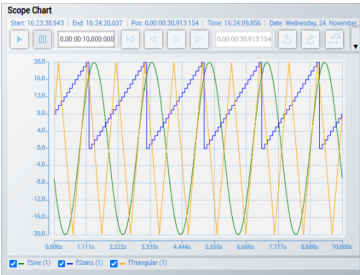
- **Chart:** Das Chart aus dem Referenced Scope, welches in der HMI angezeigt werden soll. Bisher werden folgende Charts unterstützt: YT Chart, XY Chart, Array Bar Chart und Single Bar Chart.

- **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.

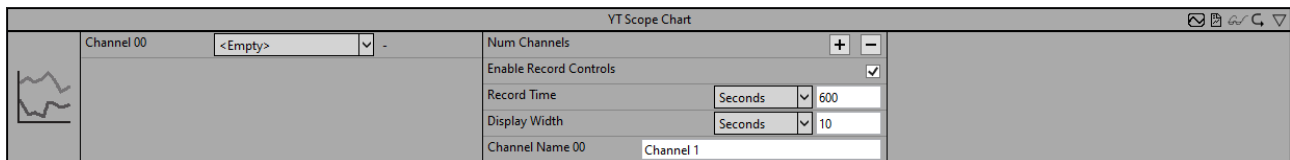


Dem Baustein können keine weiteren HMI Controls hinzugefügt werden, da dieser ausschließlich zur Darstellung des HMI Scope Controls gedacht ist.

6.5.12.3 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.

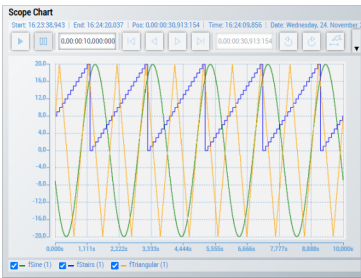
Konfigurationsoptionen

- **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.

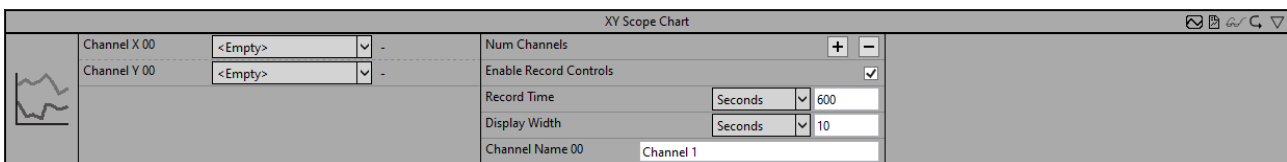


Dem Baustein können keine weiteren HMI Controls hinzugefügt werden, da dieser ausschließlich zur Darstellung des HMI Scope Controls gedacht ist.

6.5.12.4 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.

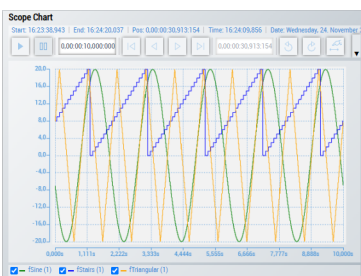
Konfigurationsoptionen

- **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

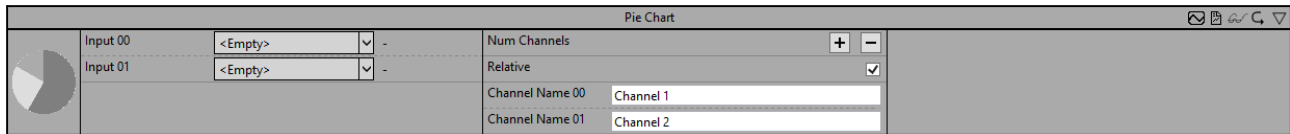
Für den Algorithmus *XY Scope Chart* ist das folgende HMI Control für die Generierung eines Analytics Dashboards verfügbar.

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.

6.5.12.5 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* parametrisiert 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.

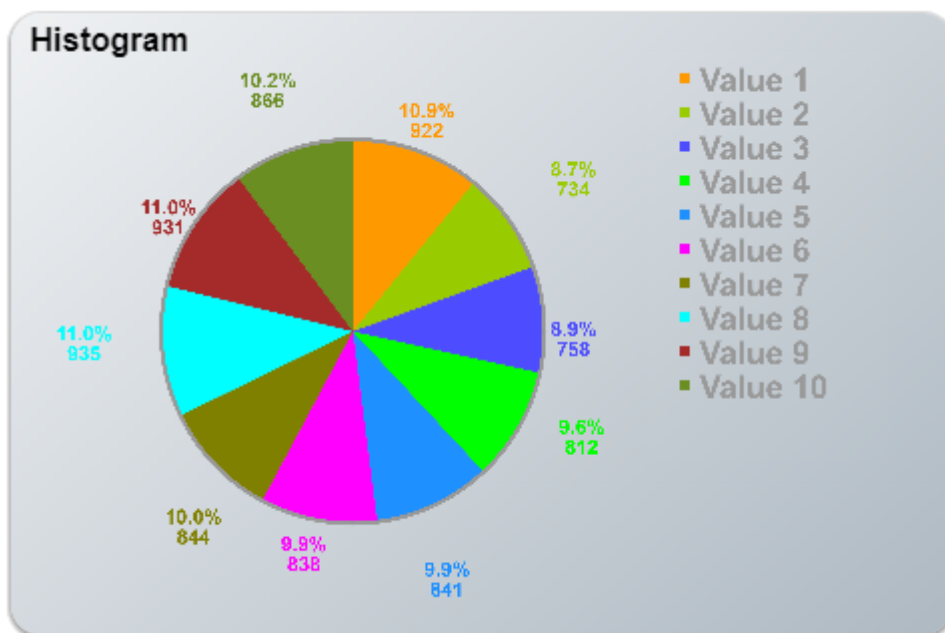
Konfigurationsoptionen

- **Num Channels:** 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.

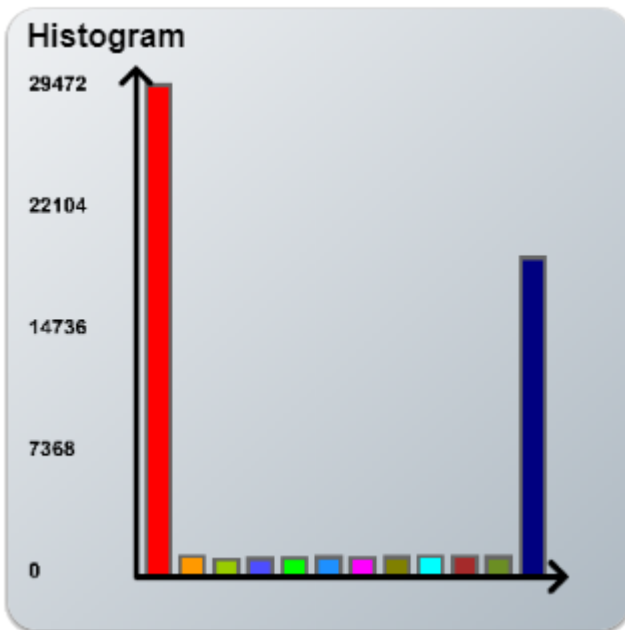
Standard HMI Controls

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 kundenspezifische HMI Controls gemappt werden.

6.5.12.6 Sankey Diagram

Sankey Diagram			
Input Inflow 00	<Empty>	-	Num Channels Inflow
Input Outflow 00	<Empty>	-	Num Channels Outflow
Input Outflow 01	<Empty>	-	Channel Name Rest
			Channel Name Inflow 00
			Channel Name Outflow 00
			Channel Name Outflow 01

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* parametrisiert 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.

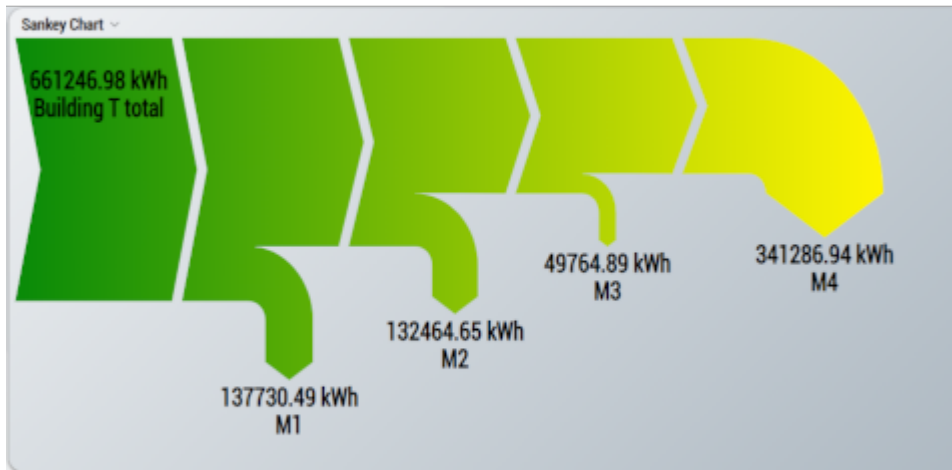
Konfigurationsoptionen

- **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.

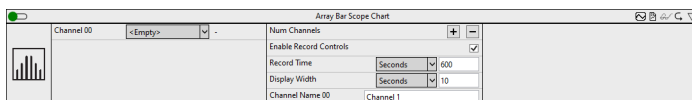


Alternativ können im Algorithmus *Sankey Diagram* mit dem Mapping Wizard kundenspezifische HMI Controls gemappt werden.

6.5.12.7 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

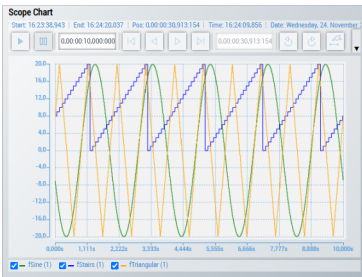
- **Num Channels:** Die Anzahl der Inputs, die im HMI Scope Control angezeigt werden sollen.
- **Channel Name 00/XX:** Die Anzeigenamen der Inputs. Diese tauchen als Legende im HMI Scope Control auf.

- **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.

6.5.13 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.

6.5.13.1 Reporting Collector

Die Reporting Collectoren sammeln Daten und versenden diese nach einem Event in einer Daten-Nachricht an den Reporting Server.

6.5.13.1.1 Reporting Collector Edge

Reporting Collector - Edge			
Edge	<Empty>	Threshold Edge	1
Channel 00	<Empty>	Report Name	Report Name
		Data Key	DataKey101
		Buffer Size	1
		Buffer on event	<input checked="" type="checkbox"/>
		Num Channels	+ -
		Alias Name 00	Channel 1
		New Result	Empty
		Last Message	Empty
		Buffer Count	Empty

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.

6.5.13.1.2 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.

6.5.13.1.3 Reporting Collector Time

Reporting Collector - Time		New Result	
Timestamp	-	Time On	hh 2 mm 0 ss 0
Channel 00	<Empty>	Day Of Week Mask	<input checked="" type="checkbox"/> Mon <input checked="" type="checkbox"/> Tue <input checked="" type="checkbox"/> Wed <input checked="" type="checkbox"/> Thu <input checked="" type="checkbox"/> Fri <input checked="" type="checkbox"/> Sat <input checked="" type="checkbox"/> Sun
		Report Name	Report Name
		Data Key	DataKey101
		Buffer Size	1
		Buffer on event	<input checked="" type="checkbox"/>
		Num Channels	+ -
		Alias Name 00	Channel 1
		Last Message	Empty
		Next Trigger	Empty
		Buffer Count	Empty

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.

6.5.13.2 Reporting Trigger

Die Reporting Trigger versenden nach einem Event eine Trigger-Nachricht an den Reporting Server und lösen damit die Erstellung eines Reports aus.

6.5.13.2.1 Reporting Trigger Edge

Reporting Trigger - Edge			
Timestamp	-	Num Channels	+ -
Channel 00	<Empty>	Threshold Edge	1
		Report Name	Report Name
		Last Trigger	Empty
		Edge Overview	Empty

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.

6.5.13.2.2 Reporting Trigger Interval

Reporting Trigger - Interval			
Timestamp	-	Interval	Seconds 1
		Report Name	Report Name
		Last Trigger	Empty
		Current Interval T...	Empty

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.

6.5.13.2.3 Reporting Trigger Time

Reporting Trigger - Time			
Timestamp	Time On	hh 2 mm 0 ss 0	Last Trigger Empty
	Day Of Week Mask	<input checked="" type="checkbox"/> Mon <input checked="" type="checkbox"/> Tue <input checked="" type="checkbox"/> Wed <input checked="" type="checkbox"/> Thu <input checked="" type="checkbox"/> Fri <input checked="" type="checkbox"/> Sat <input checked="" type="checkbox"/> Sun	Next Trigger Empty
	Report Name	Report Name	

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

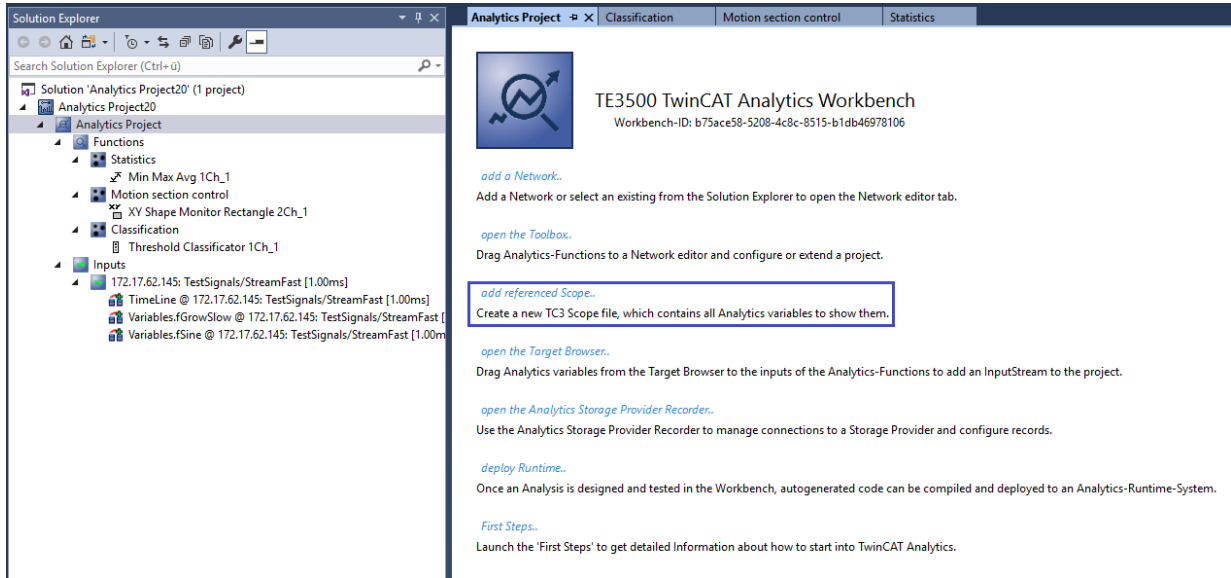
- **Last Message:** Zeigt an, wann die letzte Nachricht an den Reporting Server gesendet wurde.
- **Next Trigger:** Zeigt die verbleibende Zeit bis zur nächsten Nachricht an.

6.6 Interaktion mit Scope

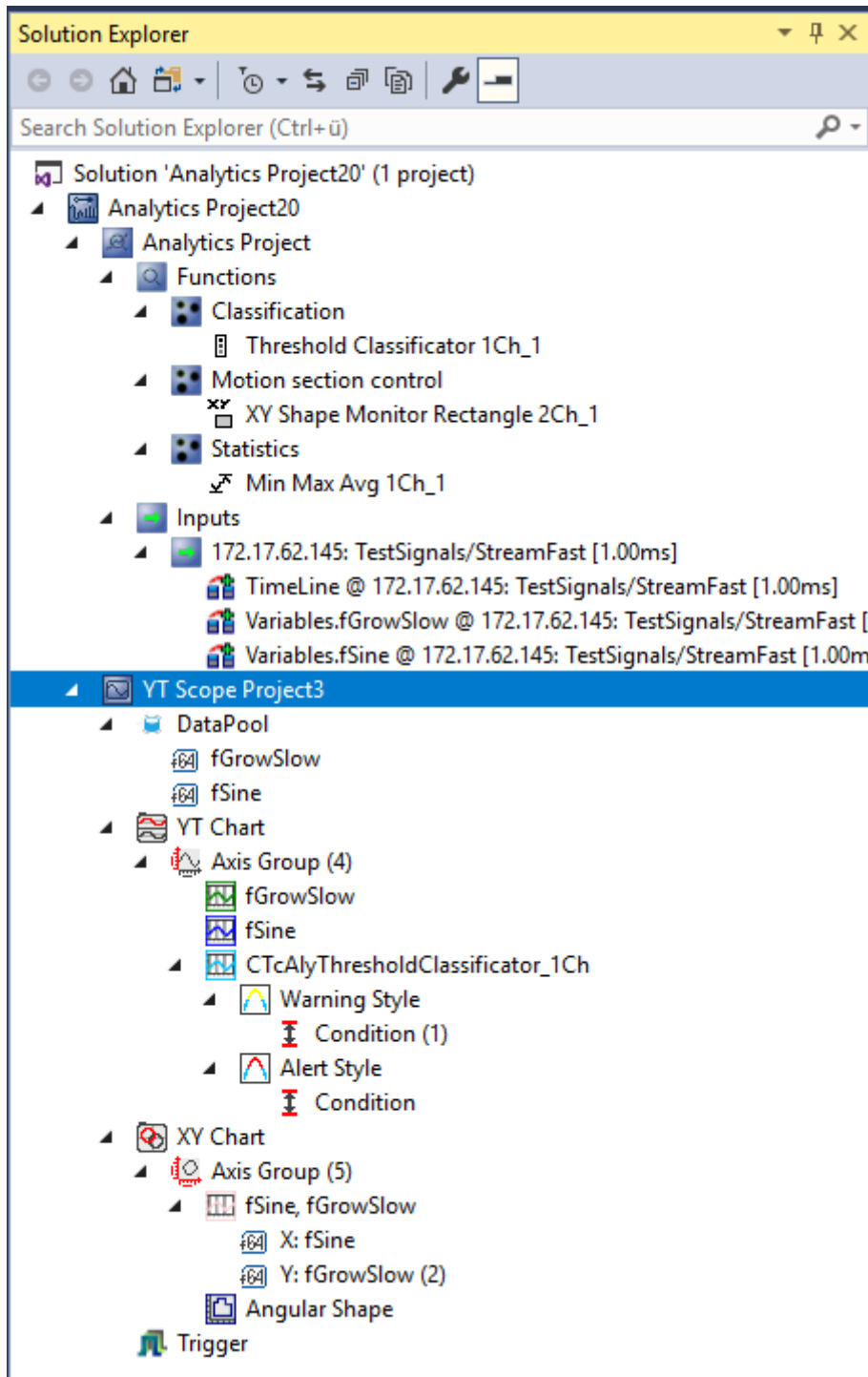
Die TwinCAT Analytics Engineering-Tools bieten eine einfache Interaktion zwischen Analytics-Konfigurator und Scope View. Sie ermöglichen es Ihnen, signifikante Werte im Datenstrom zu markieren und diese mit anderen Prozessdaten innerhalb des genauen Zyklus zu untersuchen. Zudem ist es möglich, die Ergebnisdaten des Algorithmus, wie einen durchschnittlichen oder maximalen Wert, im Scope View anzuzeigen.

- ✓ Wechseln Sie nach der Konfiguration der Analyse zur Startseite **Analytics Project**.

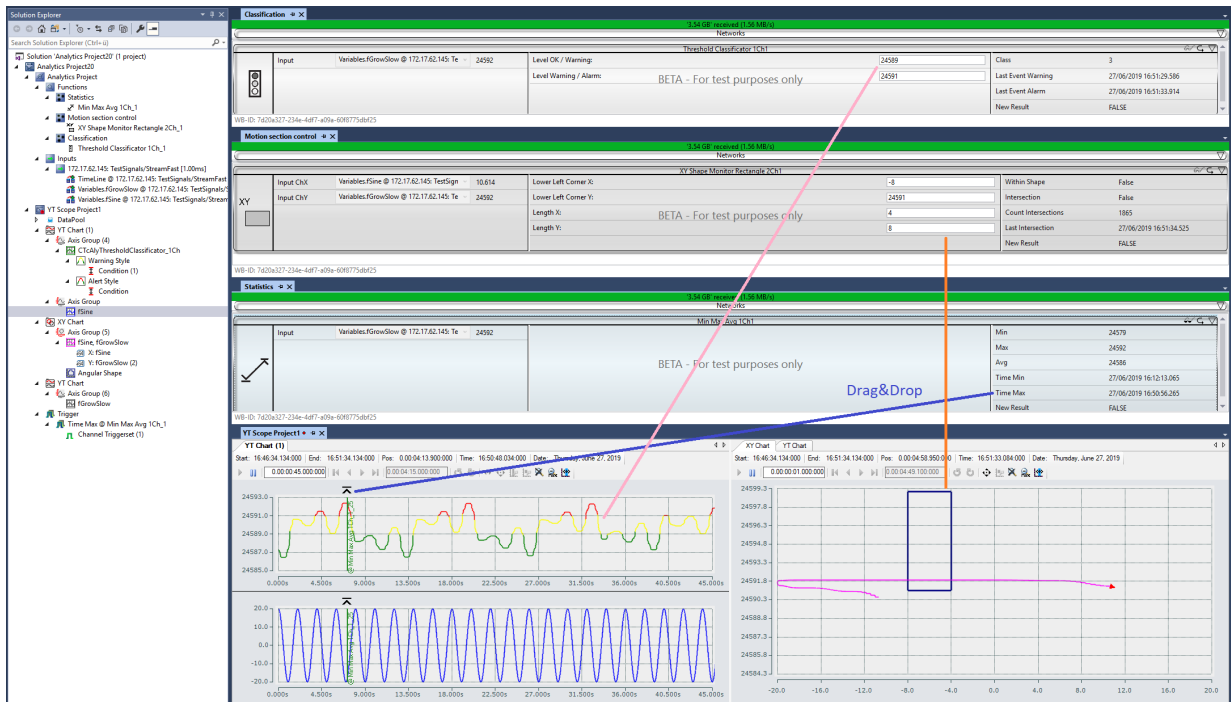
1. Klicken Sie auf **add referenced Scope...**



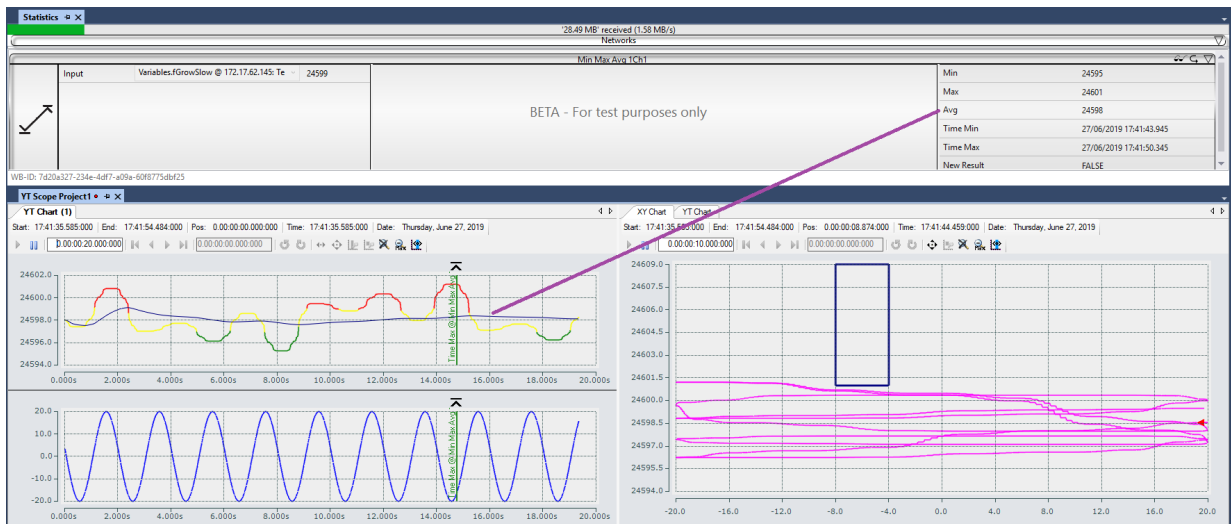
⇒ Dem Projekt wird automatisch eine geeignete Scope-Konfiguration hinzugefügt.



- 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).



- 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).
- 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.



- 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 festlegen.

The screenshot shows the TwinCAT configuration environment. The Solution Explorer on the left displays a project structure:

- YT Scope Project1
 - DataPool
 - YT Chart (1)
 - Axis Group (4)
 - CTcAlyThresholdClassifier_1Ch
 - Warning Style
 - Condition (1)
 - Alert Style
 - Condition
 - Avg @ Min Max Avg 1Ch_1 (1)
 - Axis Group
 - fSine
 - XY Chart
 - Axis Group (5)
 - fSine, fGrowSlow
 - X: fSine
 - Y: fGrowSlow (2)
 - Max @ Min Max Avg 1Ch_1, Avg @ Min Max Avg 1Ch_1
 - Y: Avg @ Min Max Avg 1Ch_1 (2)
 - Y: Max @ Min Max Avg 1Ch_1 (1)
 - Angular Shape
 - YT Chart
 - Axis Group (6)
 - fGrowSlow
 - Trigger
 - Last Intersection @ XY Shape Monitor Rectangle 2Ch_1
 - Channel Triggerset (2)
 - Time Max @ Min Max Avg 1Ch_1
 - Channel Triggerset (1)

Solution Explorer | Team Explorer

Properties ⌵ ⌵ ✕

Last Intersection @ XY Shape Monitor Rectangle 2Ch_1 TwinCAT.Measurement.B ⌵

Common
 Name: Last Intersection @ XY Shape Monitor Rec

Trigger Action
 Trigger Action: Set Mark

Trigger Group
 Color: ■ 27, 183, 0
 Enabled: True
 Trigger Category: CustomPicture
 Trigger Category Image: .\Images\5efb9f84-42f0-4d33-9f37-690b:
 Trigger Image Size: 16
 Trigger Release Capacity: 200
 Visible: True
Visible Trigger Release Capacity: Show Last 5 ⌵

Show All
 Hide all
 Show Last
 Show Last 2
Show Last 5
 Show Last 10
 Show Last 20
 Show Last 50

⇒ 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 [► 308], um bei einer erneuten Verwendung eines Netzwerks automatisch die gleiche Scope-Konfiguration zu erhalten.

i 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!

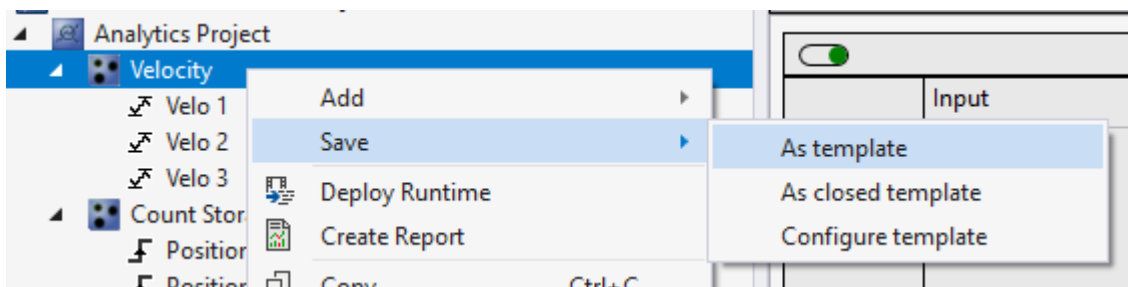
6.6.1 Scope Konfiguration im Netzwerk Template

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

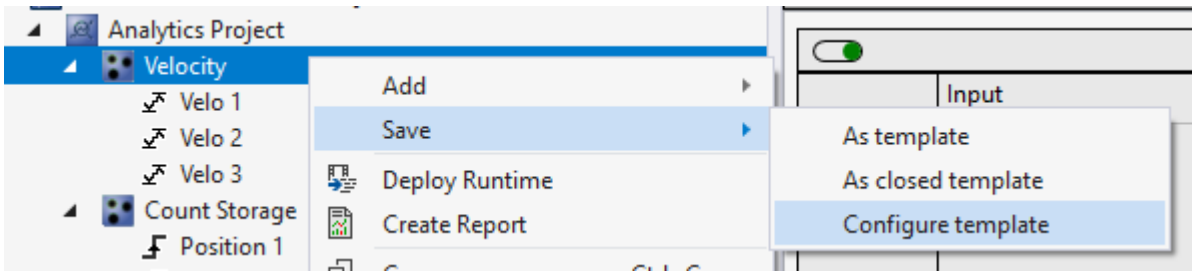
Wenn die Scope-Konfiguration ohne eine manuelle Anpassung im Template gespeichert werden soll, wählen Sie die Kontextmenü-Einträge **Save-As template** und **Save-As closed template**.



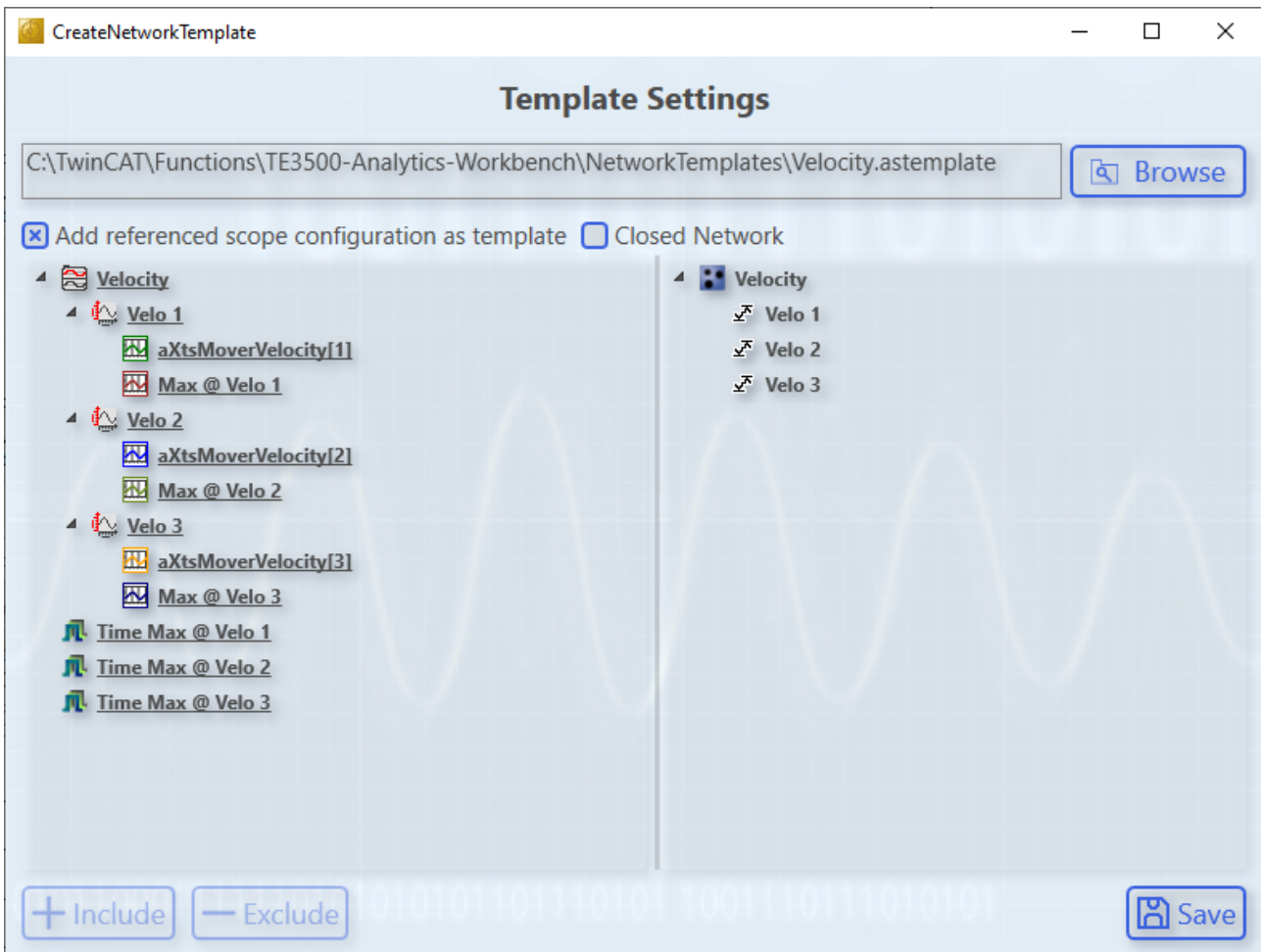
Legen Sie im nachfolgenden Dialog den Speicherort und den Namen des Templates fest. An der Stelle können Sie über dem Dateityp **Save as type** einstellen, ob die Scope-Konfiguration mit abgespeichert werden soll (**Scope Network Template**) oder ob nur das Netzwerk (**Network Template**) abgespeichert 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:



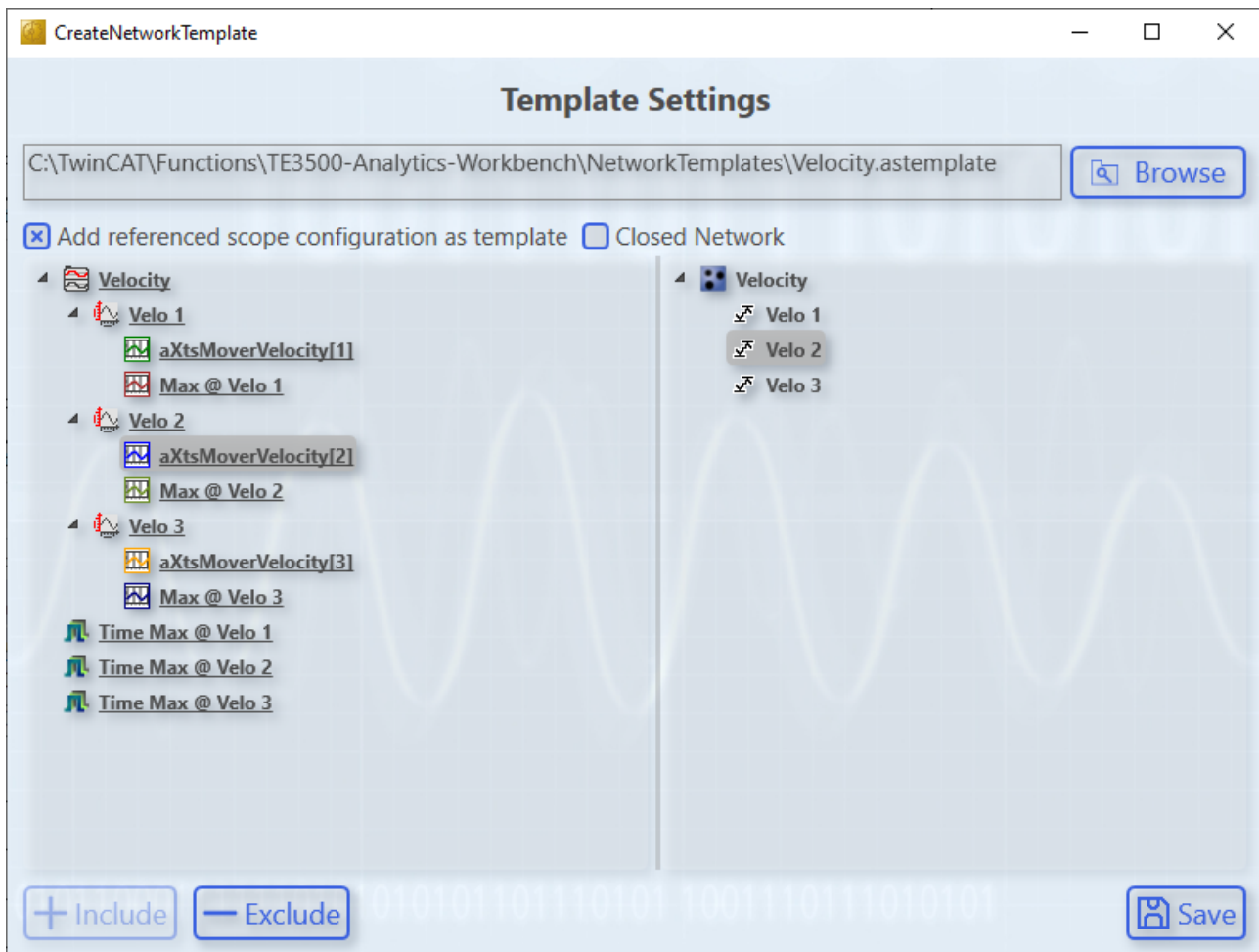
In der oberen Zeile können Sie den Speicherort und den Namen des Templates einstellen. Direkt darunter können Sie generelle Einstellungen vornehmen, welche das Template betreffen.

- **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 [▶ 67] abgespeichert werden soll. Diese Einstellung ist nur sichtbar, wenn in dem Netzwerk mindestens ein Netzwerk-Input und ein Netzwerk-Output konfiguriert sind.

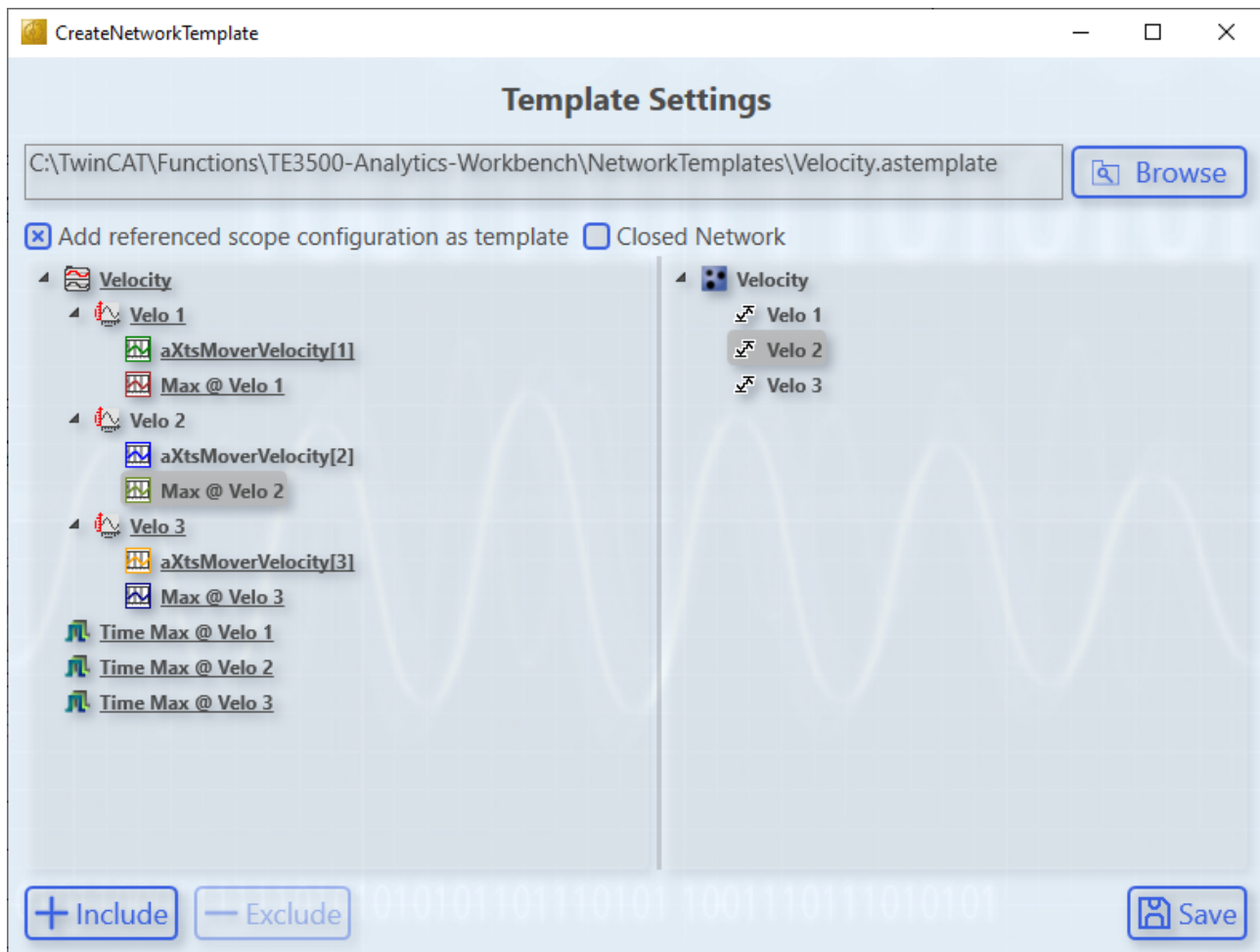
In den beiden Baumansichten darunter werden die Konfigurationen angezeigt, um die Scope-Konfiguration gegebenenfalls noch zu bearbeiten. Hier können Sie Scope-Komponenten aus dem Template ausschließen aber auch wieder hinzufügen.

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 hinzufügen. 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.

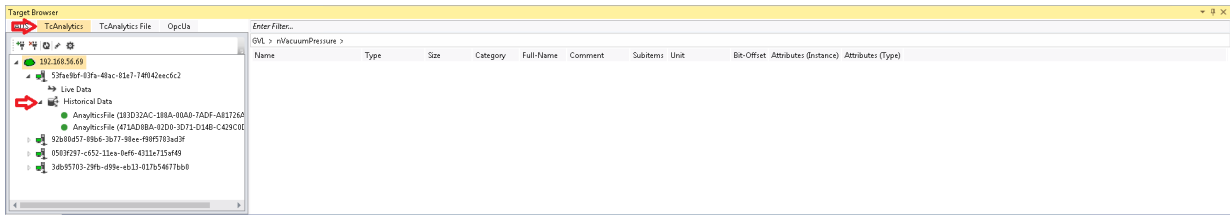
6.7 Arbeit mit historischen Daten

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.

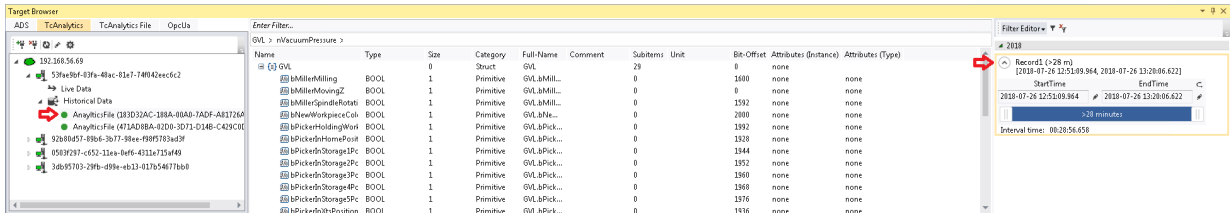
Auswahl der Daten aus dem TwinCAT Target Browser

Die historischen Daten können direkt aus dem Target Browser zu einem Eingang eines Analysealgorithmus gezogen werden.

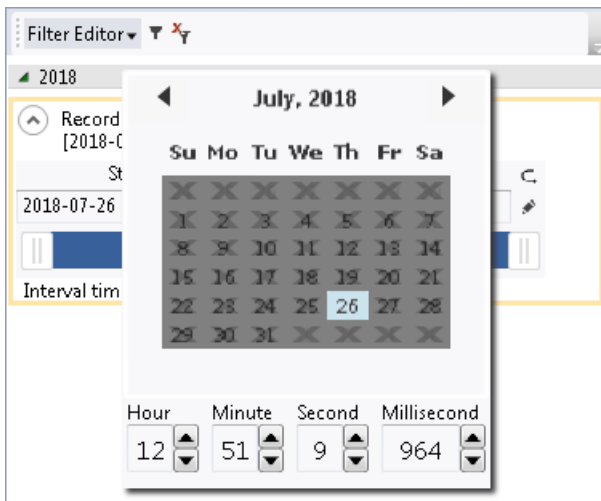
1. Zunächst müssen Sie in der linken Ecke des Target Browsers auf **TcAnalytics** klicken. Dort können Sie Ihren konfigurierten Broker sehen, der die Live- und historischen Daten Ihrer verschiedenen Geräte auflistet. Dies sollte wie in der folgenden Abbildung aussehen.



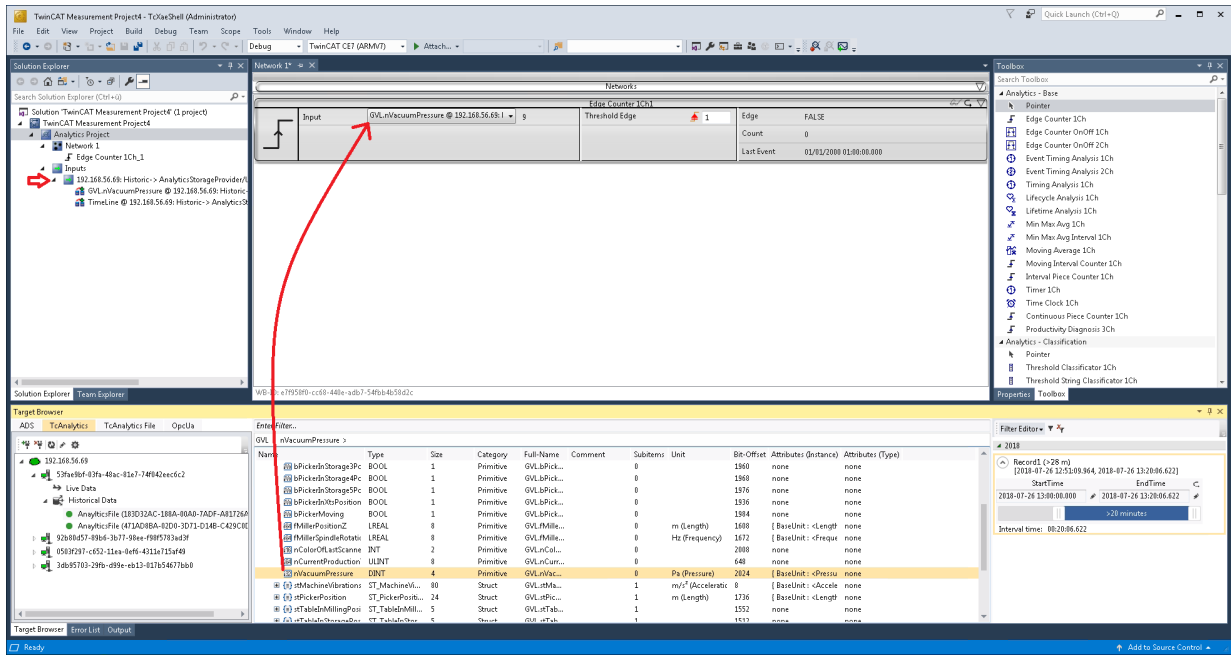
2. Gehen Sie zu dem historischen Stream, den Sie erstellt haben, und wählen Sie die Aufzeichnung aus, die analysiert werden soll. Alle Ihre Aufzeichnungen sind rechts im Fenster **Record** aufgelistet. Standardmäßig wird die letzte Aufzeichnung ausgewählt.



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.



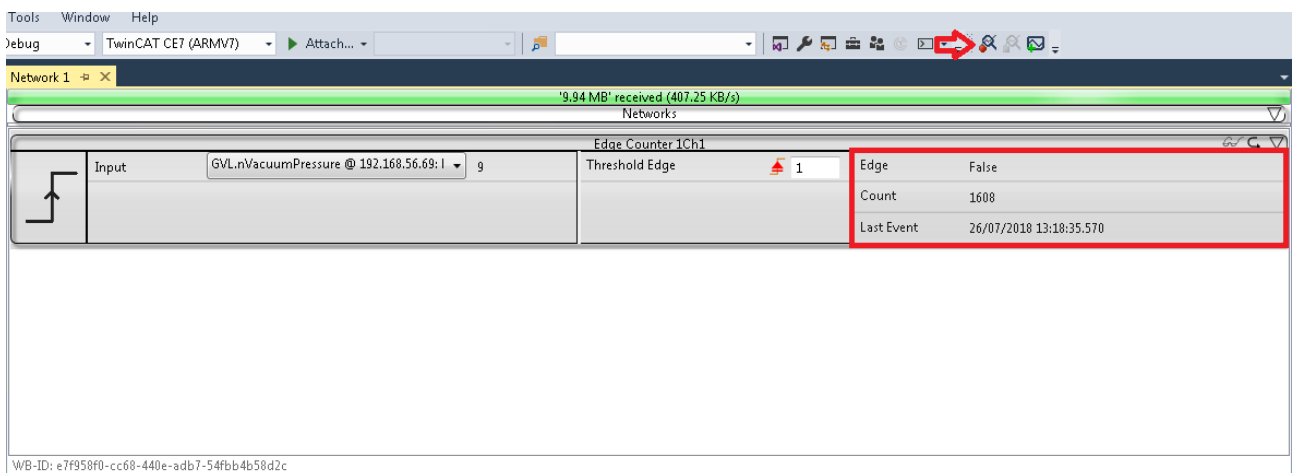
- Nach diesen Schritten können Sie ein Symbol ebenso wie bei den Symbolen der Live-Daten per Drag-and-Drop zu einem Eingang eines Algorithmus ziehen.



- ⇒ 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.



7 Anhang

7.1 FAQ – Häufig gestellte Fragen und Antworten

In diesem Abschnitt werden häufig gestellte Fragen beantwortet, um Ihnen die Arbeit mit dem TwinCAT Analytics Service Tool zu erleichtern. Falls Sie weitere Fragen haben, wenden Sie sich bitte an unser Support-Team support@beckhoff.com.

1. Ist es möglich, neben der Analytics Workbench mit dem Analytics Service Tool historische Streams zu definieren und historische Daten zu analysieren? [► 314]
2. Werden in TwinCAT Measurement Produkten Open Source Softwarekomponenten verwendet? [► 314]

Ist es möglich, neben der Analytics Workbench mit dem Analytics Service Tool historische Streams zu definieren und historische Daten zu analysieren?

Ja, zusammen mit dem Analytics Service Tool erhalten Sie den Analytics Storage Provider Recorder. Mit dem Recorder können Sie neue historische Streams definieren. Außerdem können Sie die vom TwinCAT Target Browser bereitgestellten historischen Streams für Ihre Analyse nutzen.

Werden in TwinCAT Measurement Produkten Open Source Softwarekomponenten verwendet?

Ja, es werden verschiedene Open Source Komponenten verwendet. Eine Auflistung finden Sie samt Lizenzbedingung im Verzeichnis `...\TwinCAT\Functions\TwinCAT Measurement\Legal`.

Mehr Informationen:
www.beckhoff.de/te3520

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

