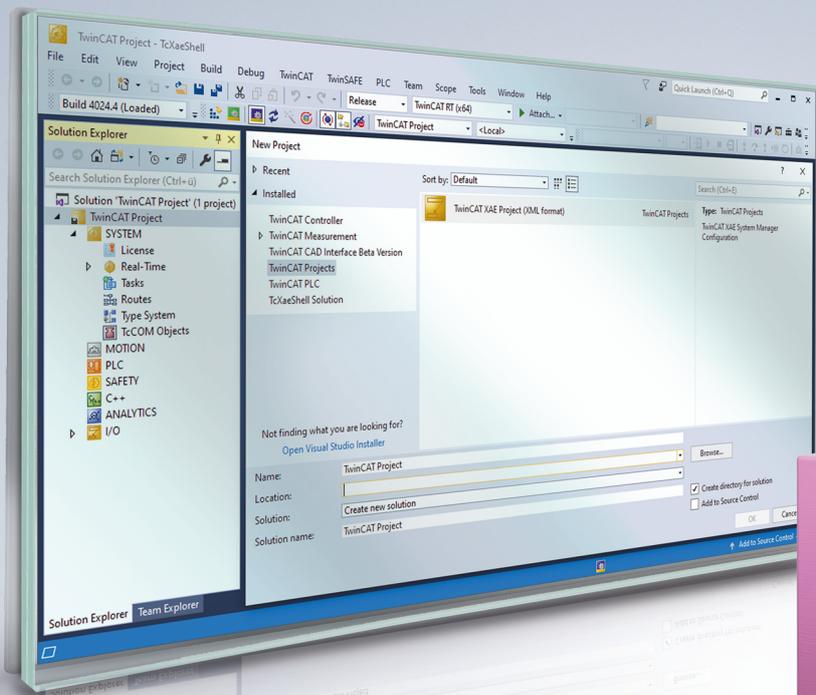


# BECKHOFF New Automation Technology

Handbuch | DE

# TF6720

TwinCAT 3 | IoT Data Agent





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>5</b>
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit	6
1.3	Hinweise zur Informationssicherheit	7
1.4	Ausgabestände der Dokumentation	7
<b>2</b>	<b>Übersicht</b>	<b>8</b>
<b>3</b>	<b>Installation</b>	<b>9</b>
3.1	Systemvoraussetzungen	9
3.2	Setup-Szenarios	9
3.3	Installation	11
3.4	Installation ab TwinCAT 4026	14
3.5	Lizenzierung	15
3.6	Lizenzierungsinformationen	17
<b>4</b>	<b>Technische Einführung</b>	<b>18</b>
4.1	Anwendungsbeispiele	18
4.1.1	AWS IoT Core	18
4.1.2	Bosch IoT Suite	20
4.1.3	Google IoT Core	20
4.1.4	IBM Watson IoT	20
4.1.5	MathWorks ThingSpeak	21
4.1.6	Microsoft Azure IoT Hub	22
4.1.7	Node-RED	22
4.2	Eigenschaften	23
4.3	Internetkonnektivität	25
4.4	MQTT Grundlagen	26
4.5	Startmodi	32
4.6	Sicherheit	33
4.7	Puffern von Daten	35
4.8	Unterstützte Protokolle und Dienste	35
4.8.1	Microsoft Azure IoT Hub	35
<b>5</b>	<b>Konfiguration</b>	<b>39</b>
5.1	Schnellstart	39
5.2	Konfigurator	41
5.2.1	Topologieansicht	43
5.2.2	Baumansicht	44
5.2.3	Zuordnungen (Mappings)	45
5.2.4	Target Browser	45
5.2.5	Cascading Editor	45
5.2.6	Parameter Editor	46
5.2.7	Settings	47
5.2.8	Fehlerprotokollierung	55
5.3	Taskleiste	56
5.4	Support Information Report	56

<b>6</b>	<b>Beispiele</b>	<b>58</b>
6.1	Schnellstart	58
6.2	Southbound	60
6.2.1	Verbindung mit Geräten von Drittanbietern über OPC UA	60
6.2.2	Verbindung mit einem BC9191 über ADS	62
6.2.3	Verbindung mit einem TwinCAT I/O Task über ADS	64
6.2.4	Verbindung mit einem TwinCAT TcCOM-Modul über ADS	65
6.3	Northbound	67
6.3.1	Veröffentlichung von Daten an AWS IoT Core	67
6.3.2	Veröffentlichung von Daten an AWS Greengrass	69
6.3.3	Veröffentlichung von Daten an IBM Watson IoT	71
6.3.4	Veröffentlichung von Daten an Microsoft Azure IoT Hub	73
6.3.5	Veröffentlichung von Daten an einen MQTT-Message-Broker	75
6.3.6	Veröffentlichung von Daten an die App TwinCAT IoT Communicator	77
6.3.7	Abonnieren eines MQTT-Message-Brokers	78
6.3.8	Verwendung eines Microsoft Azure IoT Hub Gerätezwillings	80
<b>7</b>	<b>Anhang</b>	<b>83</b>
7.1	Fehlerprotokollierung	83
7.2	Fehlerdiagnose	83
7.3	Support und Service	85

# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

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

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

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

### Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

### Marken

Beckhoff®, 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>.

## 1.4 Ausgabestände der Dokumentation

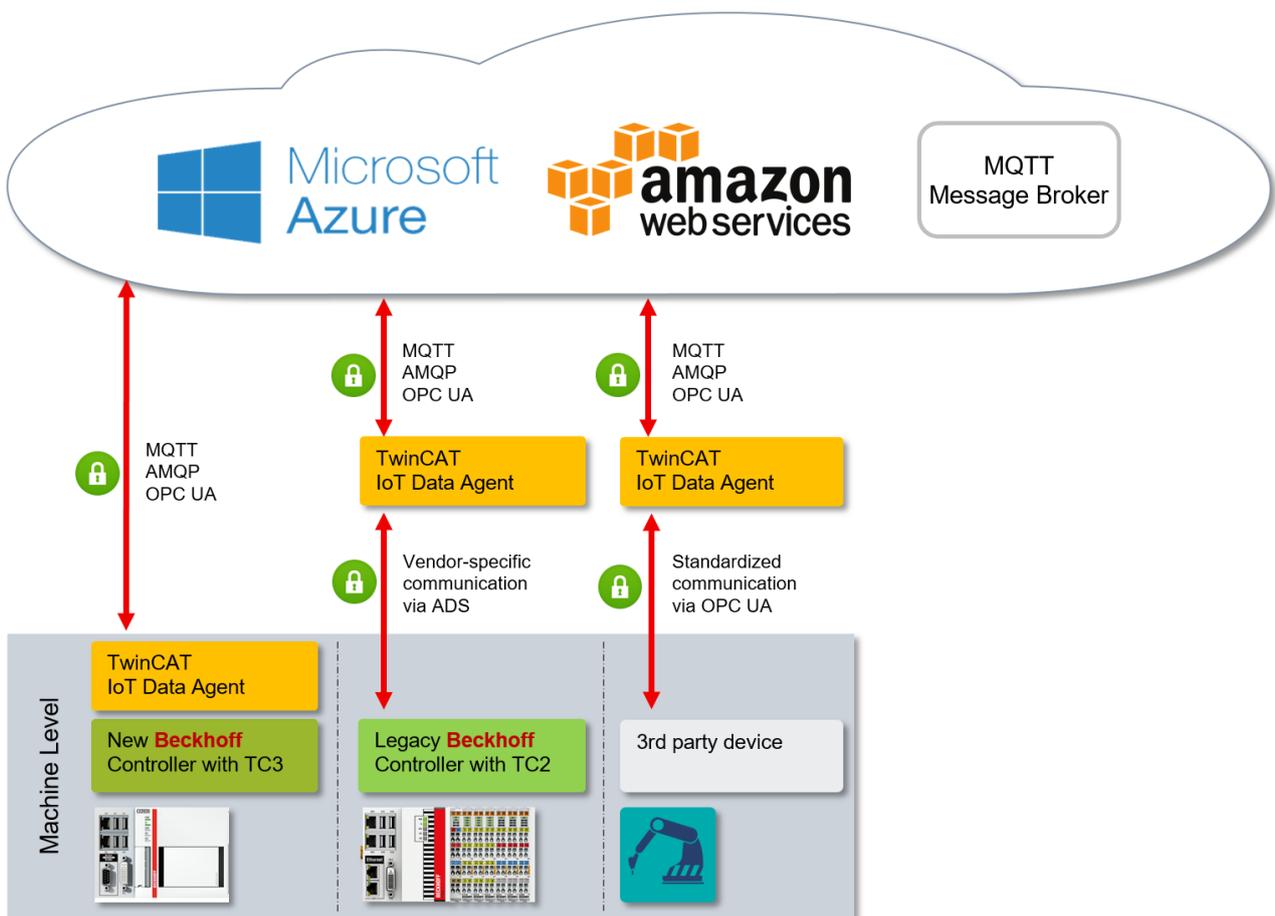
Version	Änderung
1.5.x	Neu: Installation/Installation ab TwinCAT 4026

## 2 Übersicht

Die TwinCAT-Funktion TC3 IoT Data Agent bietet bidirektionale Konnektivität mit verschiedenen Cloud-Diensten. Es handelt sich um eine Gateway-Anwendung, die entweder auf der Steuerung oder auf einem Gateway-Computer installiert werden kann. Der TC3 IoT Data Agent kann so konfiguriert werden, dass er verschiedene Datenquellen miteinander verbindet, z. B. eine TwinCAT 3 PLC (ADS) und einen MQTT-Message-Broker. Außerdem können öffentliche Cloud-Dienste genutzt werden, beispielsweise AWS IoT und Microsoft Azure IoT Hub. Der TC3 IoT Data Agent umfasst einen integrierten OPC UA-Client, der die Verbindung von Geräten von Drittanbietern mit der Cloud ermöglicht. Ältere Anwendungen mit TwinCAT 2 können durch Ausführung des TC3 IoT Data Agent auf einem Gateway-Computer und durch Nutzung der ADS- oder OPC UA-Konnektivität mit dem TwinCAT 2-System verbunden werden.

Beim Senden oder Empfangen von Daten können die Benutzer verschiedenste Datenformate auswählen, von effizienten Binärformaten bis hin zu ASCII-basierten JSON-Formaten. Dadurch kann der TC3 IoT Data Agent in unterschiedlichen Szenarios verwendet werden, z. B. um Geräte mit TwinCAT Analytics zu verbinden.

Zur Optimierung des Datenverkehrs können verschiedene Kommunikationsmuster konfiguriert werden, z. B. Polling- oder On-Change-Muster.



### Komponenten

Der TC3 IoT Data Agent besteht aus den folgenden Komponenten:

- Kernanwendung: Hintergrunddienst, der die Logik bereitstellt
- Konfigurator: grafisches Tool zur Erstellung/Bearbeitung und Bereitstellung einer Konfiguration für die Kernanwendung

Der TwinCAT 3 IoT Data Agent kann in verschiedenen Setup-Szenarios installiert und verwendet werden, z. B. direkt auf der Steuerung oder auf einem Gateway-Computer (siehe [Setup-Szenarios](#) [9]).

## 3 Installation

### 3.1 Systemvoraussetzungen

In der folgenden Tabelle sind die Systemvoraussetzungen für den TC3 IoT Data Agent aufgeführt.

Technische Daten	Beschreibung
Betriebssystem	Windows 7/10, Windows Embedded Standard 7
Zielplattform	PC-Architektur (x86, x64)
.NET Framework	4.5.2 (nur für den Konfigurator erforderlich)
TwinCAT Version <sup>1</sup>	TwinCAT 2, TwinCAT 3
TwinCAT-Installationsebene <sup>2 3</sup>	TwinCAT 3 XAE, XAR, ADS (Build 4022.22 und höher)
Erforderliche TwinCAT-Lizenz	TF6720 TC3 IoT Data Agent (enthält 4 Gates)
Zusätzliche Lizenzen <sup>4</sup>	TF6721 TC3 IoT Data Agent Gate Pack 4 TF6722 TC3 IoT Data Agent Gate Pack 16 TF6723 TC3 IoT Data Agent Gate Pack 64 TF6724 TC3 IoT Data Agent Gate Pack 256

<sup>1</sup> Version der TwinCAT-Laufzeit, mit der der TC3 IoT Data Agent verbunden werden kann (über ADS, alternativ auch über zusätzliche Funktionen, z. B. TwinCAT OPC UA)

<sup>2</sup> Für die Lizenzierung ist mindestens eine Setup-Ebene von TwinCAT 3 ADS erforderlich.

<sup>3</sup> XAE ist derzeit für den Konfigurator erforderlich.

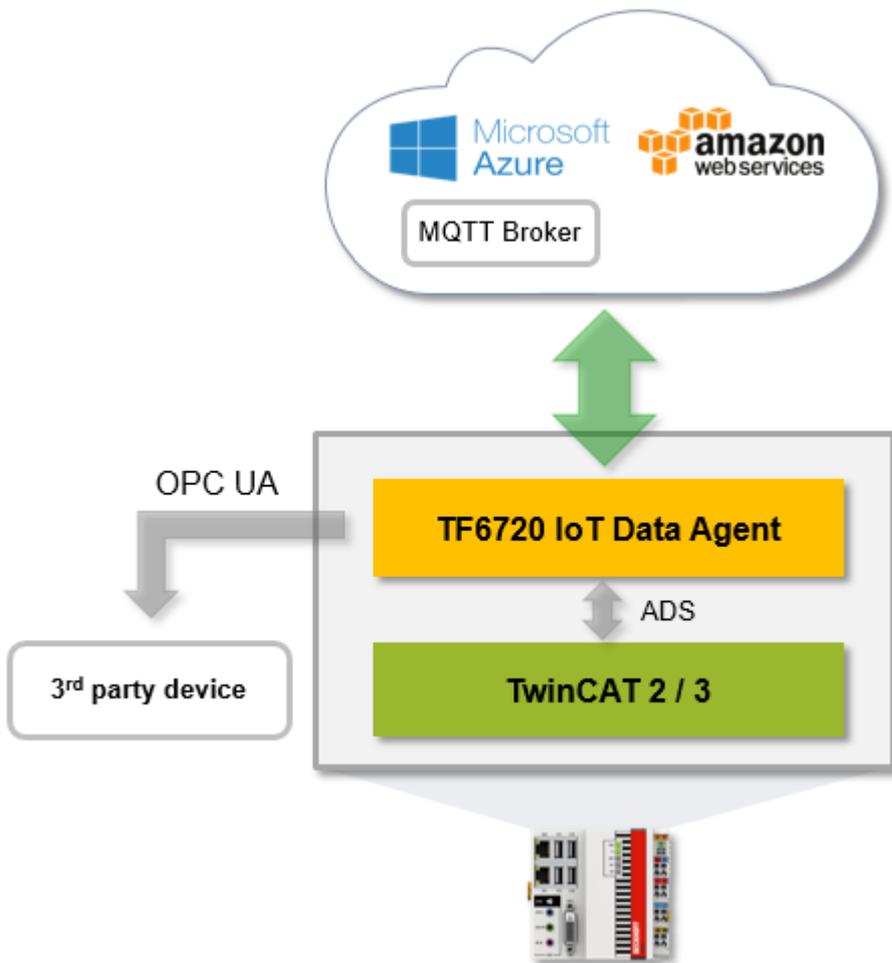
<sup>4</sup> Jede zusätzliche Lizenz kann einmal pro System erworben werden und kommt zur Gesamtanzahl der Gates hinzu.

### 3.2 Setup-Szenarios

Der TC3 IoT Data Agent kann in nahezu jedem Einsatzszenario verwendet werden. Er kann entweder direkt auf der Industriesteuerung (siehe [Systemvoraussetzungen](#) [► 9]) oder auf einem Gateway-Computer, der mehrere Geräte zusammenfasst oder zur Verbindung älterer Anwendungen wie TwinCAT 2 dient, installiert und verwendet werden. Jedes Setup-Szenario hat Vorteile und Nachteile, abhängig von der Systemumgebung und dem Projekt, für das der TC3 IoT Data Agent verwendet werden soll.

#### Ausführung des TC3 IoT Data Agent auf der Steuerung

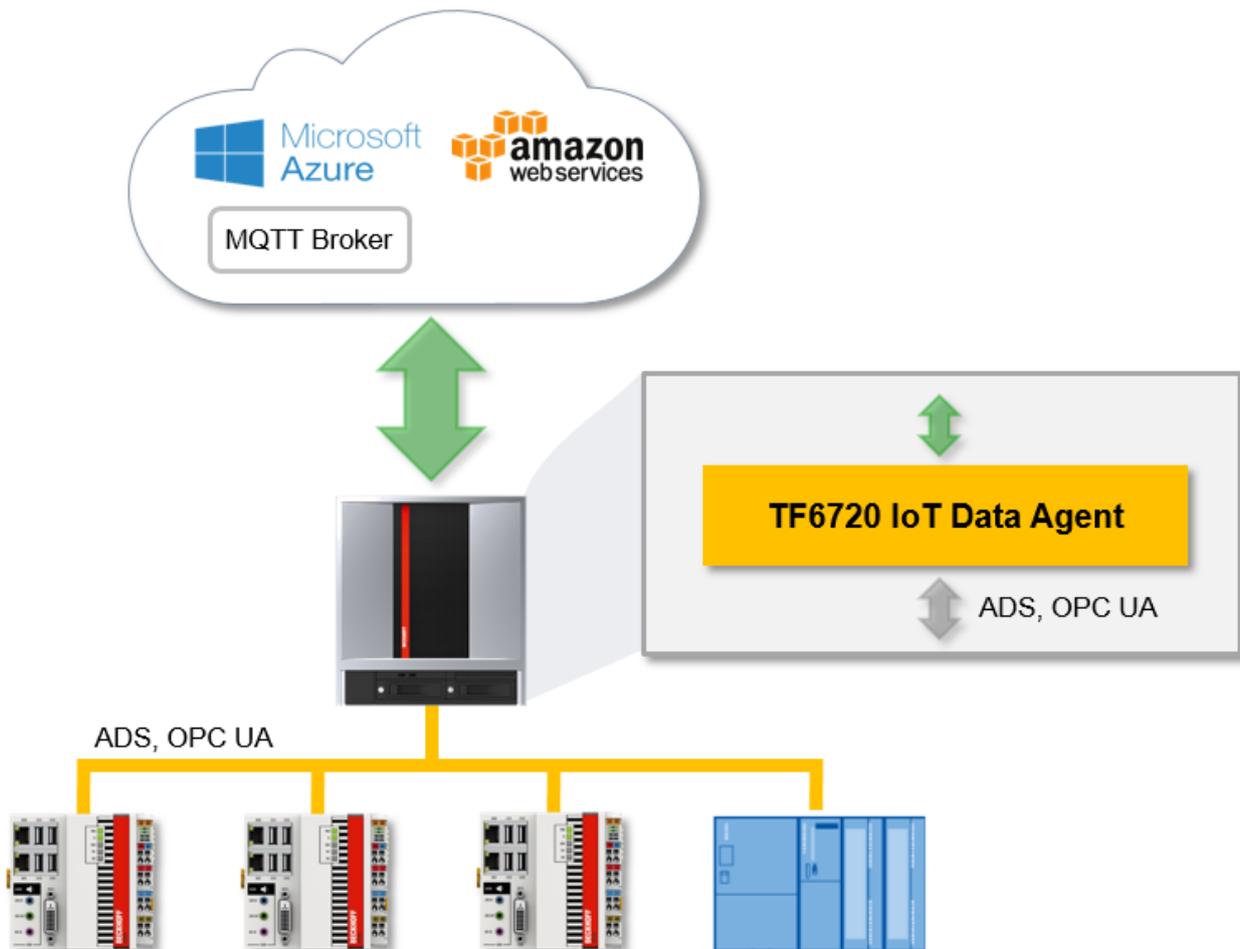
Dies ist das Standardszenario. Der TC3 IoT Data Agent wird direkt auf einer Beckhoff Industriesteuerung (IPC oder Embedded-PC) installiert und ist nur mit der TwinCAT-Laufzeit auf diesem System verbunden. Dies ist das bevorzugte Szenario, da jedes IoT-Gerät eine eigene Verbindung zum Cloud-Dienst haben sollte, obgleich Gateway-Szenarios möglich und manchmal sogar notwendig sind.



### Ausführung des TC3 IoT Data Agent auf einem Gateway-Computer

In diesem Szenario wird der TC3 IoT Data Agent auf einem Gateway-Computer installiert, der mehrere Geräte zusammenfasst, z. B. zur Verbindung mit vorhandenen Maschinenanwendungen bei einer Nachrüstung. Dieses Szenario kann auch mit anderen TwinCAT-Funktionen kombiniert werden, z. B. einem TwinCAT OPC UA Server, der auch mehrere TwinCAT-Laufzeiten zusammenfassen kann. Ein typisches Hardwaregerät für ein solches Setup-Szenario ist ein Beckhoff C6015 Ultra-Kompakt-Industrie-PC.

Beachten Sie, dass der Netzwerkverkehr zwischen dem TC3 IoT Data Agent und den zugrunde liegenden Geräten in diesem Setup-Szenario sehr hoch sein kann, da der TC3 IoT Data Agent Symbole von jeder verbundenen Laufzeit abtasten muss und diese Abtastung über das Netzwerk erfolgt.



### TC3 IoT Data Agent Konfigurator

Der TC3 IoT Data Agent Konfigurator [► 41] ist ein grafisches Konfigurationstool. Er kann auf demselben Computer installiert werden, der die TC3 IoT Data Agent-Kernanwendung ausführt, oder auf einem Engineering-Computer, um die Konfiguration aus der Ferne zu erstellen/bearbeiten und im Netzwerk bereitzustellen.

## 3.3 Installation

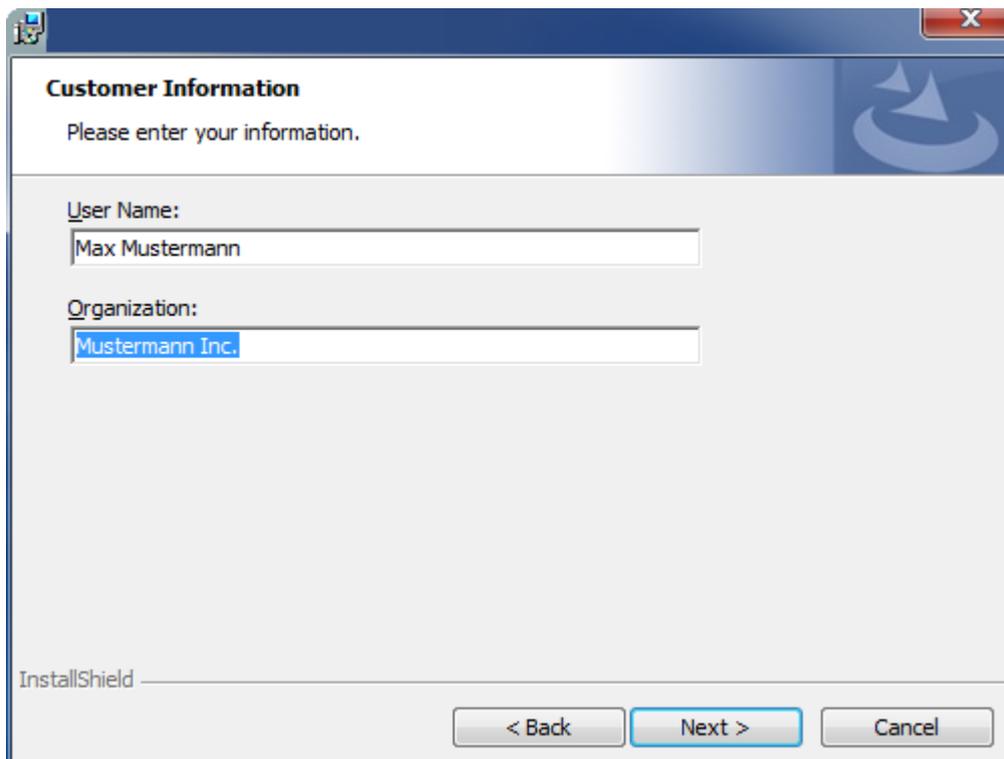
Nachfolgend wird beschrieben, wie die TwinCAT 3 Function für Windows-basierte Betriebssysteme installiert wird.

- ✓ Die Setup-Datei der TwinCAT 3 Function wurde von der Beckhoff-Homepage heruntergeladen.
- 1. Führen Sie die Setup-Datei als Administrator aus. Wählen Sie dazu im Kontextmenü der Datei den Befehl **Als Administrator ausführen**.
  - ⇒ Der Installationsdialog öffnet sich.

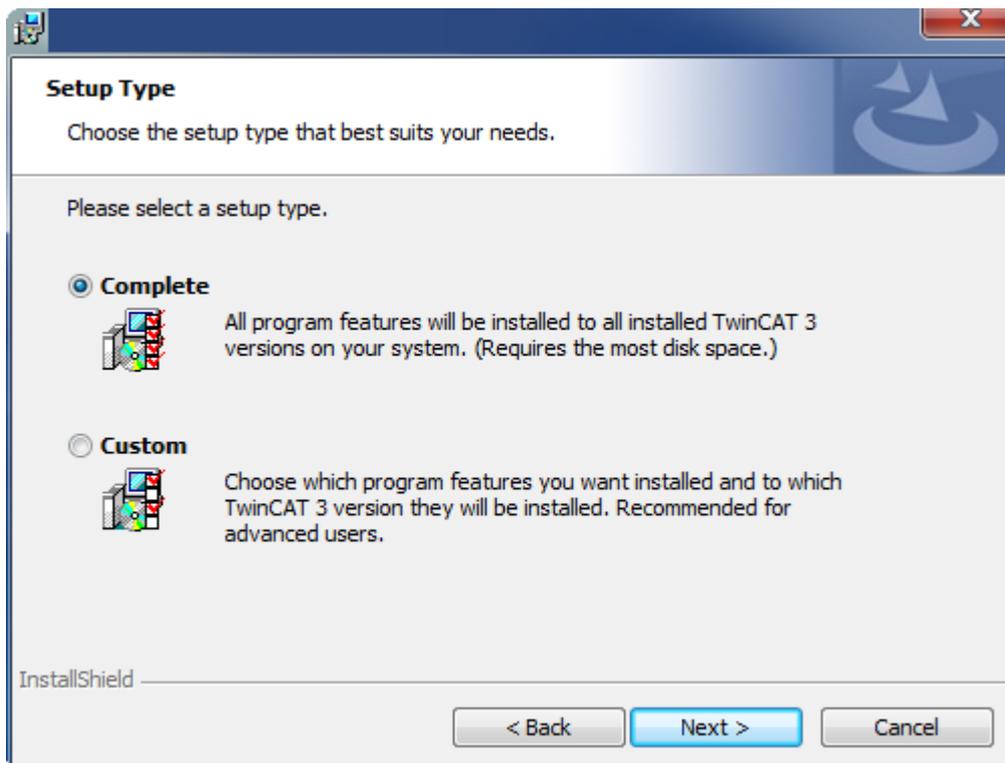
2. Akzeptieren Sie die Endbenutzerbedingungen und klicken Sie auf **Next**.



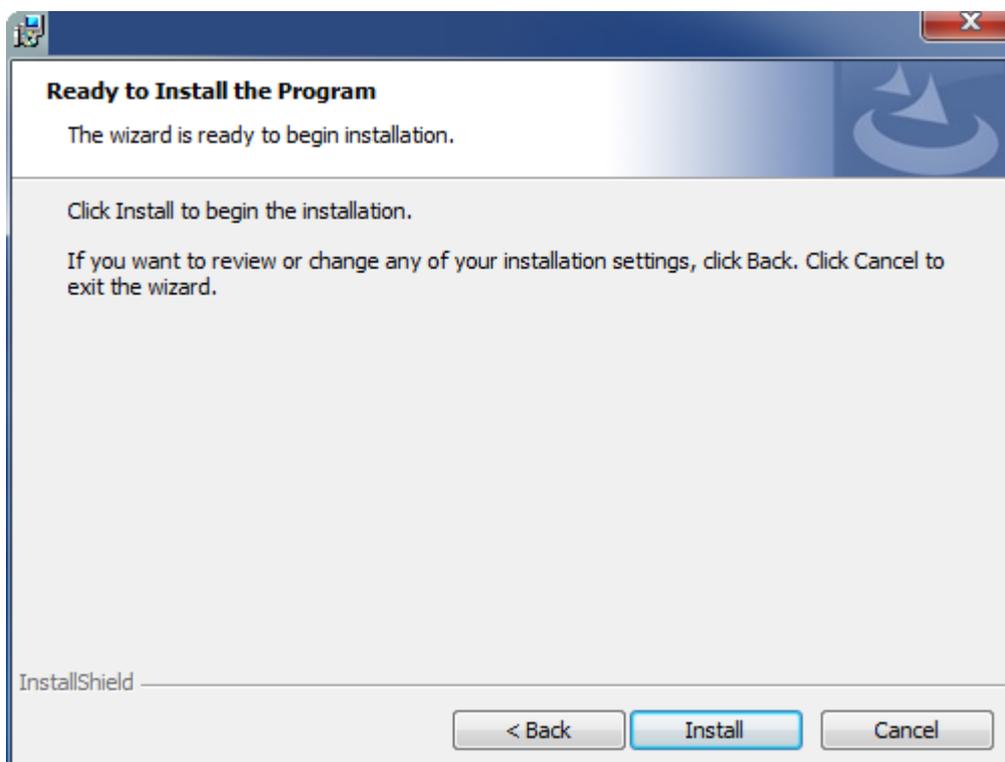
3. Geben Sie Ihre Benutzerdaten ein.



4. Wenn Sie die TwinCAT 3 Function vollständig installieren möchten, wählen Sie **Complete** als Installationstyp. Wenn Sie die Komponenten der TwinCAT 3 Function separat installieren möchten, wählen Sie **Custom**.

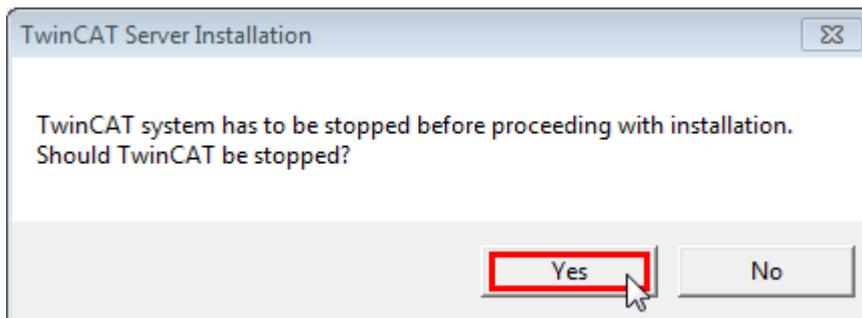


5. Wählen Sie **Next** und anschließend **Install**, um die Installation zu beginnen.

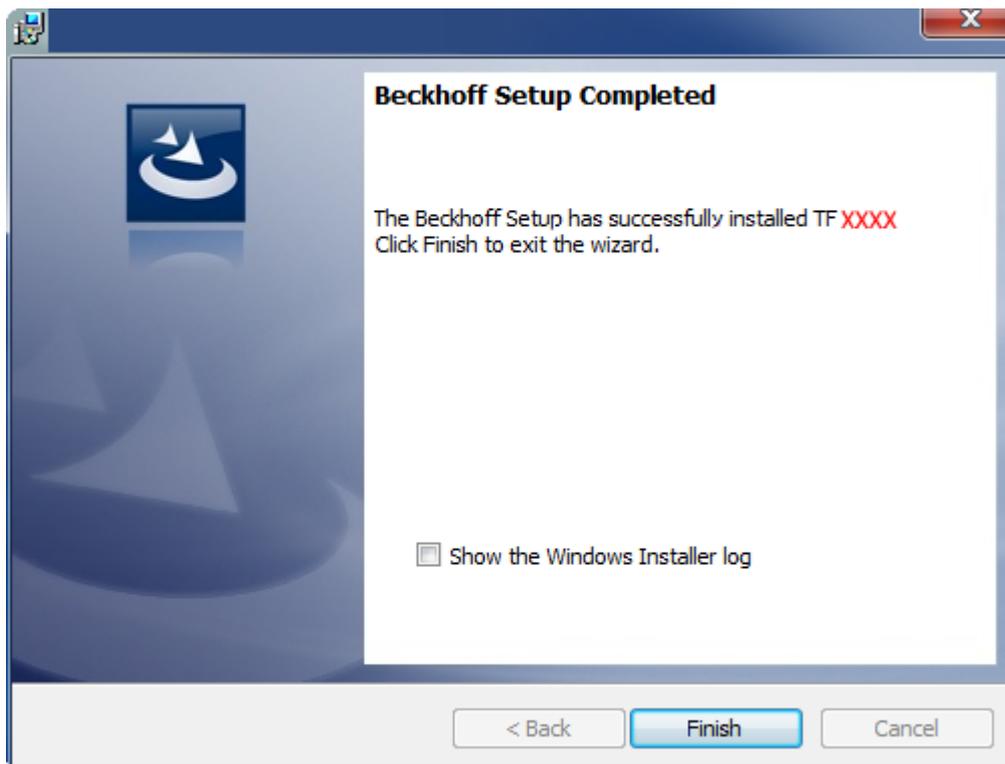


- ⇒ Ein Dialog weist Sie darauf hin, dass das TwinCAT-System für die weitere Installation gestoppt werden muss.

6. Bestätigen Sie den Dialog mit **Yes**.



7. Wählen Sie **Finish**, um das Setup zu beenden.



⇒ Die TwinCAT 3 Function wurde erfolgreich installiert.

## 3.4 Installation ab TwinCAT 4026

### TwinCAT Package Manager

Wenn Sie TwinCAT 3.1 Build 4026 (und höher) auf dem Betriebssystem Microsoft Windows verwenden, können Sie diese Function über den TwinCAT Package Manager installieren, siehe [Dokumentation zur Installation](#).

Normalerweise installieren Sie die Function über den entsprechenden Workload; dennoch können Sie die im Workload enthaltenen Pakete auch einzeln installieren. Diese Dokumentation beschreibt im Folgenden kurz den Installationsvorgang über den Workload.

### Kommandozeilenprogramm TcPkg

Über das TcPkg **Command Line Interface (CLI)** können Sie sich die verfügbaren Workloads auf dem System anzeigen lassen:

```
tcpkg list TF6720
```

Über das folgende Kommando können Sie den Workload der TF6720 TC3 IoT Data Agent-Function installieren.

```
tcpkg install TF6720.IoTDataAgent.XAE
tcpkg install TF6720.IoTDataAgent.XAR
```

## TwinCAT Package Manager UI

Über das **User Interface (UI)** können Sie sich alle verfügbaren Workloads anzeigen lassen und diese bei Bedarf installieren.

Folgen Sie hierzu den entsprechenden Anweisungen in der Oberfläche.

## 3.5 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

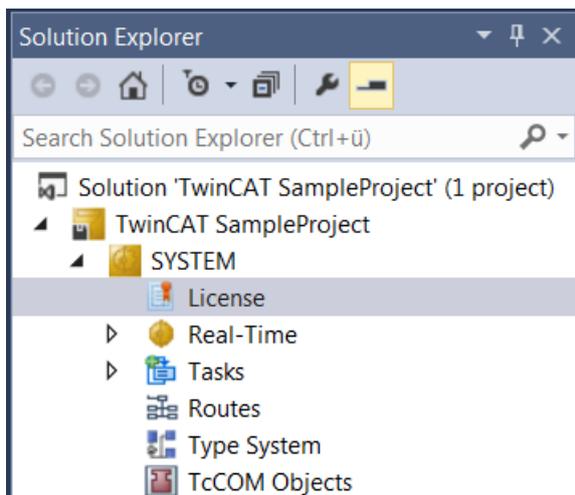
Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT-3-Lizenzierung](#)“.

### 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“).

Order No	License	Add License
TF3601	TC3 Condition Monitoring Level 2	<input type="checkbox"/> cpu license
TF3650	TC3 Power Monitoring	<input type="checkbox"/> cpu license
TF3680	TC3 Filter	<input type="checkbox"/> cpu license
TF3800	TC3 Machine Learning Inference Engine	<input type="checkbox"/> cpu license
TF3810	TC3 Neural Network Inference Engine	<input type="checkbox"/> cpu license
TF3900	TC3 Solar-Position-Algorithm	<input type="checkbox"/> cpu license
TF4100	TC3 Controller Toolbox	<input checked="" type="checkbox"/> cpu license
TF4110	TC3 Temperature-Controller	<input type="checkbox"/> cpu license
TF4500	TC3 Speech	<input type="checkbox"/> cpu license

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									
<input type="checkbox"/> Disable automatic detection of required licenses for project												
<table border="1"> <thead> <tr> <th>License Device</th> <th>Target (Hardware Id)</th> <th>Add...</th> </tr> </thead> <tbody> <tr> <td>System Id:</td> <td>Platform:</td> <td></td> </tr> <tr> <td>2DB25408-B4CD-81DF-5488-6A3D9B49EF19</td> <td>other (91)</td> <td></td> </tr> </tbody> </table>				License Device	Target (Hardware Id)	Add...	System Id:	Platform:		2DB25408-B4CD-81DF-5488-6A3D9B49EF19	other (91)	
License Device	Target (Hardware Id)	Add...										
System Id:	Platform:											
2DB25408-B4CD-81DF-5488-6A3D9B49EF19	other (91)											
<table border="1"> <thead> <tr> <th colspan="2">License Request</th> </tr> </thead> <tbody> <tr> <td>Provider:</td> <td>Beckhoff Automation</td> </tr> <tr> <td>License Id:</td> <td>Customer Id:</td> </tr> <tr> <td>Comment:</td> <td></td> </tr> </tbody> </table>				License Request		Provider:	Beckhoff Automation	License Id:	Customer Id:	Comment:		
License Request												
Provider:	Beckhoff Automation											
License Id:	Customer Id:											
Comment:												
<table border="1"> <thead> <tr> <th colspan="2">License Activation</th> </tr> </thead> <tbody> <tr> <td><b>7 Days Trial License...</b></td> <td>License Response File...</td> </tr> </tbody> </table>				License Activation		<b>7 Days Trial License...</b>	License Response File...					
License Activation												
<b>7 Days Trial License...</b>	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:	OK
Kg8T4	
<input type="text"/>	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.

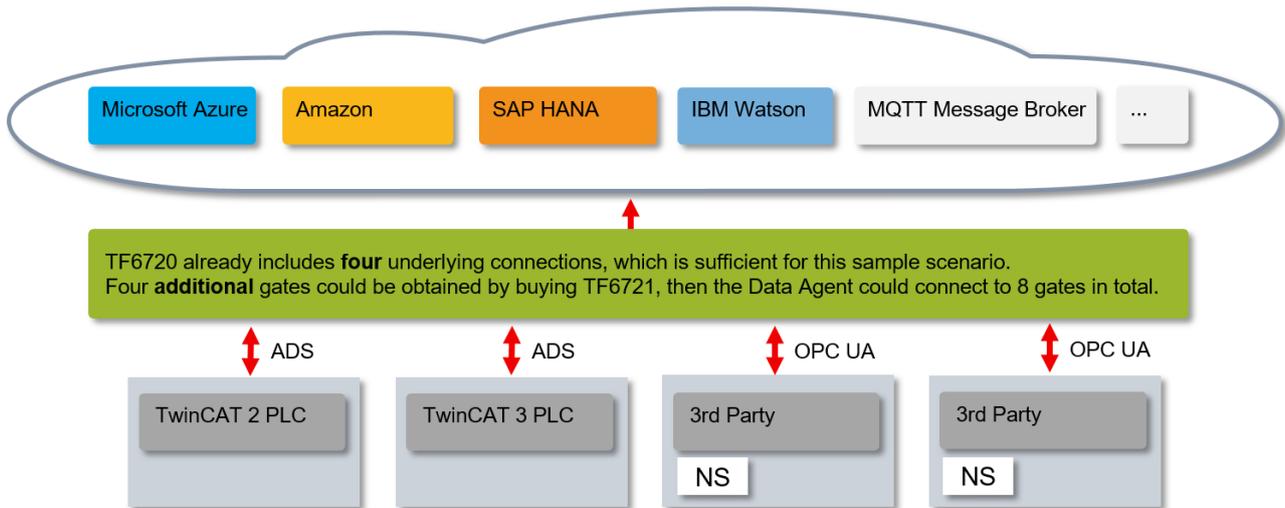
⇒ In der tabellarischen Übersicht der Lizenzen gibt der Lizenzstatus nun das Ablaufdatum der Lizenz an.

10. Starten Sie das TwinCAT-System neu.

⇒ Die 7-Tage-Testversion ist freigeschaltet.

### 3.6 Lizenzierungsinformationen

Die Lizenzierung des TC3 IoT Data Agent erfolgt auf der Grundlage sogenannter „Gate Packs“. Jedes ADS-Gerät oder jeder OPC UA Server-Namensraum, mit dem sich der TC3 IoT Data Agent verbindet, wird als „Gate“ betrachtet. Die TC3 IoT Data Agent-Basislizenz „TF6720“ enthält bereits vier Gates. Zusätzliche Gates können durch den Kauf einer der „Gate Pack“-Lizenzen (TF6721-TF6724) erworben werden. Diese Lizenzen kommen dann zu der Basislizenz hinzu, z. B. TF6720 + TF6721 = 8 Gates.



## 4 Technische Einführung

Der TC3 IoT Data Agent ist eine Gateway-Anwendung, die mehrere Datenquellen miteinander verbindet. Eine Datenquelle ist in der Regel ein Kommunikationsprotokoll, z. B. ADS, OPC UA oder MQTT. Der TC3 IoT Data Agent wird über einen grafischen Konfigurator konfiguriert, der eine XML-basierte Konfigurationsdatei erstellt oder bearbeitet. Beim Start wird diese Datei von der TC3 IoT Data Agent-Kernanwendung gelesen, die als Dienst installiert ist und daher im Hintergrund des Betriebssystems läuft.

Einfach ausgedrückt: Die XML-Datei gibt dem TC3 IoT Data Agent vor, was zu tun ist.

- Welche Datenquellen verbunden werden sollen
- Welche Cloud-Dienste genutzt werden sollen
- Welche Symbole (Variablen) in welcher Richtung ausgetauscht werden sollen
- Welche Datenraten und Abtastmodi verwendet werden sollen

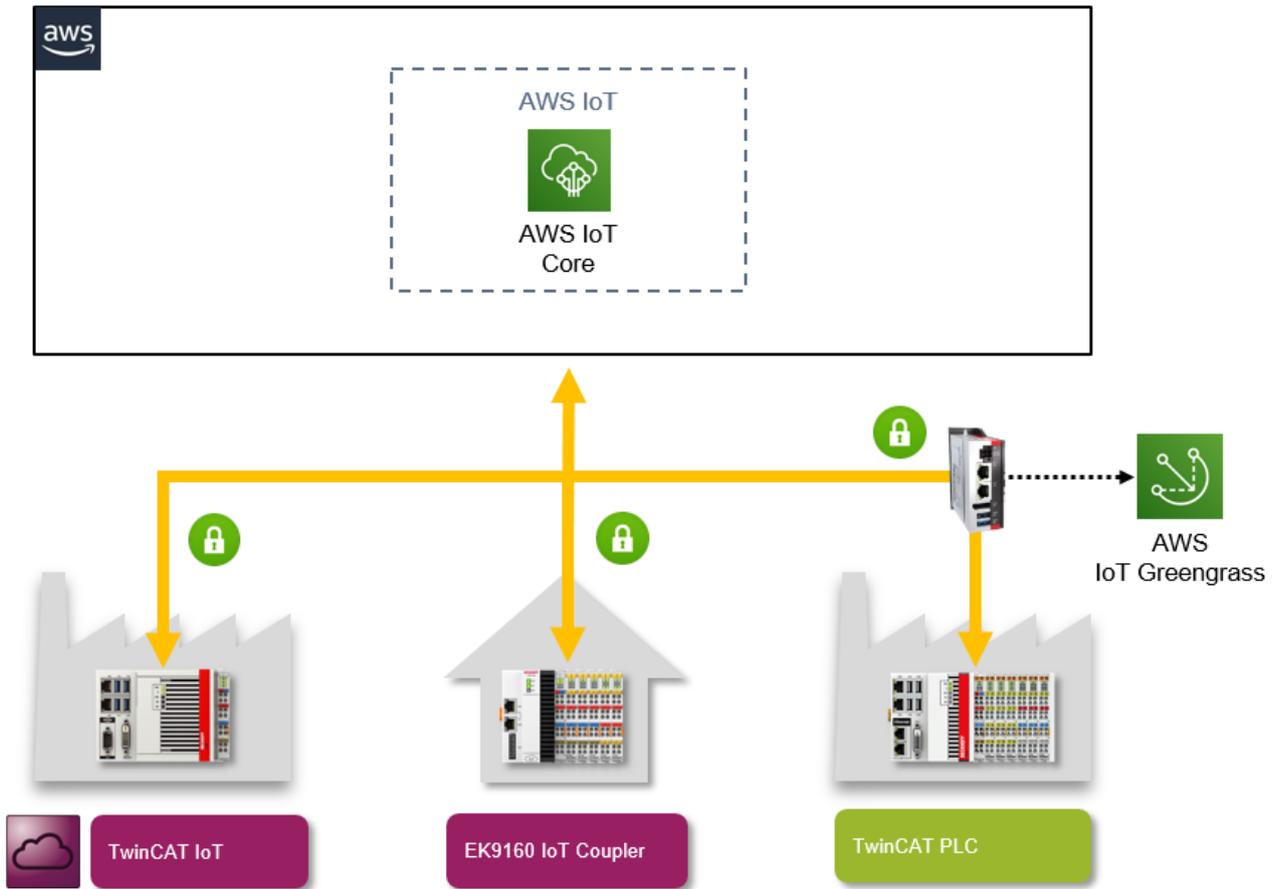
Eine typische Installation des TC3 IoT Data Agent auf einer Steuerung oder einem Gateway-Computer umfasst nur die TC3 IoT Data Agent-Kernanwendung. Der grafische Konfigurator wird üblicherweise auf einem Engineering-Computer installiert und verwendet, um die Kernanwendung über das Netzwerk zu verbinden, er kann jedoch auch auf der Steuerung oder dem Gateway-Computer installiert und verwendet werden.

### 4.1 Anwendungsbeispiele

Die folgenden Abschnitte enthalten einige Anwendungsbeispiele, in denen der TC3 IoT Data Agent verwendet werden kann.

#### 4.1.1 AWS IoT Core

AWS IoT Core ist ein verwalteter Cloud-Service, über den verbundene Geräte einfach und sicher mit Cloud-Anwendungen und anderen Geräten bi-direktional kommunizieren können. Spezielle Funktionalitäten, wie z. B. der AWS IoT Device Shadow Service, ermöglichen hierbei auch die Kommunikation mit Geräten die aktuell noch nicht verbunden sind.



**AWS IoT Core und TwinCAT IoT**

AWS IoT Core basiert auf dem Transportprotokoll MQTT. Daher ist es möglich TwinCAT IoT zu verwenden, um Nachrichten an AWS IoT Core oder AWS IoT Greengrass zu senden bzw. von dort zu empfangen.

Diverse Beispiele zeigen wie Sie eine Verbindung zum AWS IoT Core Dienst herstellen können.

Beispiel	Produkt	Beschreibung
lotMqttSampleAwsIoT	TF6701	Dieses Beispiel demonstriert, wie Sie MQTT Funktionsbausteine innerhalb der SPS Logik verwenden können, um eine Verbindung zu AWS IoT Core herzustellen und Daten auszutauschen.
Data Agent AWS IoT [▶ 67]	TF6720	Dieses Beispiel demonstriert, wie Sie den TwinCAT IoT Data Agent konfigurieren können, um eine Verbindung zu AWS IoT Core herzustellen und Daten auszutauschen.

Die Beispiele lassen sich äquivalent auch auf AWS IoT Greengrass abbilden. Der zugehörige AWS IoT Greengrass Core lässt sich hierbei zum Beispiel auf einem C6015 Industrie-PC installieren und betreiben. Typischerweise initiiert der Greengrass Core (bzw. eine dort bereitgestellte Lambda-Funktion) die Datenkommunikation mit der Steuerung, z. B. über OPC UA. Da der Greengrass Core jedoch einen integrierten Message Broker besitzt, kann diese Kommunikationsrichtung auch umgedreht werden. In diesem Fall würde zum Beispiel TwinCAT IoT über einen MQTT Kanal eine Verbindung zum Greengrass Core herstellen und entsprechend Daten austauschen.

**• Weitere Informationen**

**i** Weitere Informationen zu AWS IoT Core entnehmen Sie bitte der [AWS IoT Core Dokumentation](#).

## 4.1.2 Bosch IoT Suite

Die Bosch IoT Suite ist eine auf offenen Standards und Open Source basierende IoT-Plattform und unterstützt die nahtlose Integration in das Bosch IoT Eco-System. Der Bosch IoT Hub stellt hierbei Geräten diverse Kommunikationsschnittstellen zur Verfügung um Daten mit der Bosch IoT Cloud auszutauschen.

### Bosch IoT Suite und TwinCAT IoT

Die Bosch IoT Suite beinhaltet einen MQTT Message Broker - den sogenannten Bosch IoT Hub. Daher ist es möglich TwinCAT IoT zu verwenden, um Nachrichten an die Bosch IoT Suite zu versenden oder von dort zu empfangen.

Folgendes Beispiel zeigt wie Sie eine Verbindung zum Bosch IoT Hub aufbauen können.

Beispiel	Produkt	Beschreibung
lotMqttSampleBoschIoT	TF6701	Dieses Beispiel demonstriert, wie Sie MQTT Funktionsbausteine innerhalb der SPS Logik verwenden können, um eine Verbindung zum Bosch IoT Hub herzustellen und Daten auszutauschen.

### Weitere Informationen



Weitere Informationen zur Bosch IoT Suite entnehmen Sie bitte der [offiziellen Bosch IoT Suite Webseite](#).

## 4.1.3 Google IoT Core

Google IoT Core ist ein verwalteter Clouddienst, welcher es verteilten Geräten über einen sicheren Transportkanal ermöglicht, Daten mit der Google Cloud auszutauschen.

### Google IoT Core und TwinCAT IoT

Google IoT Core beinhaltet einen MQTT Message Broker. Daher ist es möglich, TwinCAT IoT zu verwenden, um Nachrichten an Google IoT Core zu versenden oder von dort zu empfangen.

Folgendes Beispiel zeigt wie Sie eine Verbindung zu Google IoT Core herstellen können.

Beispiel	Produkt	Beschreibung
lotMqttSampleGoogleIoT	TF6701	Dieses Beispiel demonstriert, wie Sie MQTT Funktionsbausteine von innerhalb der SPS Logik verwenden können, um eine Verbindung zu Google IoT Core herzustellen und Daten auszutauschen.

## 4.1.4 IBM Watson IoT

Bei IBM Watson IoT handelt es sich um eine IoT Suite der IBM Cloud, welche diverse Dienste anbietet, um IoT-Geräte mit Diensten auf IBM Bluemix zu verbinden, eingehende Nachrichten weiterzuverarbeiten oder Nachrichten an die Geräte zu versenden. Die Funktionalitäten in IBM Watson IoT erlauben aus Gerätesicht eine einfache und sichere Verbindung von IoT-Geräten mit IBM-Diensten, indem sie eine bidirektionale Kommunikation zwischen den Geräten und IBM Watson IoT ermöglicht.

**IBM Watson IoT und TwinCAT IoT**

Da IBM Watson IoT über das Transportprotokoll MQTT erreichbar ist, ist es möglich, TwinCAT IoT zu verwenden, um Nachrichten an IBM Watson IoT zu senden bzw. von dort zu empfangen.

Diverse Beispiele zeigen wie Sie eine Verbindung zu IBM Watson IoT herstellen können.

Beispiel	Produkt	Beschreibung
lotMqttSampleIbmWatsonIoT	TF6701	Dieses Beispiel demonstriert, wie Sie MQTT Funktionsbausteine verwenden können, um eine Verbindung zu IBM Watson IoT herzustellen und Daten auszutauschen.
<u>Veröffentlichung von Daten an IBM Watson IoT</u>	TF6720	Dieses Beispiel demonstriert, wie Sie den TwinCAT IoT Data Agent konfigurieren können, um eine Verbindung zu IBM Watson IoT herzustellen und Daten auszutauschen.

**4.1.5 MathWorks ThingSpeak**

ThingSpeak™ ist eine IoT-Plattform von The MathWorks®, bekannt unter anderem durch die Softwarelösungen MATLAB® und Simulink®.

Die Plattform bietet (neben einer REST API) eine MQTT-Schnittstelle, über die Daten aus der TwinCAT-Runtime nach ThingSpeak™ gesendet werden können. ThingSpeak™ ermöglicht das Sammeln, Speichern, Analysieren, Visualisierung und das Reagieren auf eingehende Daten. Ein wichtiger Abgrenzungspunkt zu anderen Plattformen ist die Möglichkeit im Web-Browser MATLAB®-Code zu schreiben und diesen zur Analyse und zur Visualisierung der Daten zu nutzen. Dabei ist es möglich, vorhandene Lizenzen für Toolboxes von der On-premis-Programmierungsumgebung ebenfalls auf ThingSpeak™ zu verwenden.

**Funktionsweise**

Der Data Ingest sowie das Speichern von Daten erfolgt auf Basis von sogenannten Channels. Jeder Channel besitzt 8 Felder, welche mit eingehenden Daten befüllt werden können. Neben den 8 Datenfeldern stehen noch weitere Meta-Felder zur Verfügung wie z. B. latitude, longitude, altitude oder auch ein Zeitstempel. Daten, die zu einem Channel gepublished werden, werden in einer Datenbank mit der Möglichkeit eines Datenexports (JSON, XML, CSV) abgelegt. Die Anzahl der Nachrichten pro Zeiteinheit, die an ein Channel gesendet werden können, ist abhängig von der hinterlegten ThingSpeak™-Lizenz. Aktuell basiert die MQTT-Schnittstelle auf dem Senden von Strings, welche vom ThingSpeak Channel interpretiert werden.

Beispiele für mögliche Aktionen auf ThingSpeak™:

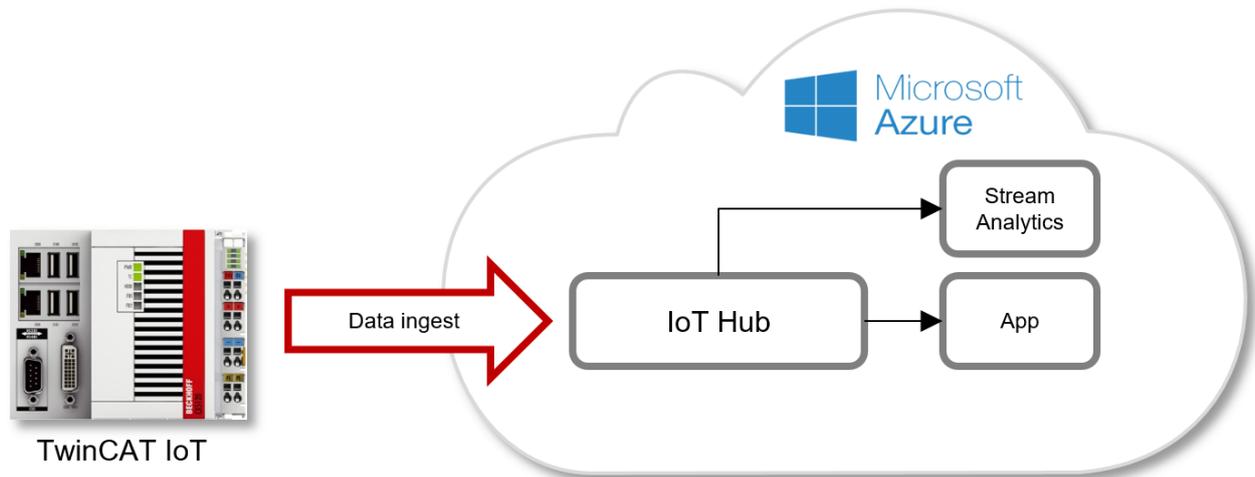
- Sende Nachricht an Twitter
- Zyklisches Ausführen von Matlab Analysis scripts
- Ausführen eines Matlab Analysis scripts, getriggert durch channel conditions

**Anwendungen**

Aufgrund der vom Cloud-Dienst limitierten Datenrate, die von der Steuerung aktuell an ThingSpeak™ gesendet werden kann, ist ein ausgeprägter Edge-Computing-Ansatz eine zielführende Strategie. Über das Beckhoff Produkt TE1400 können MATLAB®/Simulink®-Modelle in die TwinCAT-Runtime integriert werden und somit, neben den verschiedenen TwinCAT-Funktionen (Condition Monitoring, Filter, ...), Algorithmen zur Informationsverdichtung in der Echtzeit ausgeführt werden. Des Weiteren sind Prozesse mit großen Zeitkonstanten gut mit ThingSpeak™ zu handhaben, z. B. Energiedatenmanagement, Building Automation, etc.

## 4.1.6 Microsoft Azure IoT Hub

Bei dem Microsoft Azure IoT Hub handelt es sich um eine IoT Suite in der Azure Cloud, welche diverse Dienste anbietet, um IoT-Geräte mit Diensten auf Azure zu verbinden, eingehende Nachrichten weiterzuverarbeiten oder Nachrichten an die Geräte zu versenden. Die Funktionalitäten im Microsoft Azure IoT Hub erlauben aus Gerätesicht eine einfache und sichere Verbindung von IoT-Geräten mit Azure-Diensten, indem sie eine bidirektionale Kommunikation zwischen den Geräten und dem Azure IoT Hub ermöglichen.



### Microsoft Azure IoT Hub und TwinCAT IoT

Der Microsoft Azure IoT Hub bietet mehrere Kommunikationsschnittstellen an, um Nachrichten entgegenzunehmen oder zu verschicken - unter anderem auch MQTT. Daher ist es möglich TwinCAT IoT zu verwenden, um Nachrichten an den Microsoft Azure IoT Hub zu senden bzw. von diesem zu empfangen oder auch den Device Twin anzusteuern bzw. Methodenaufrufe an das Device auszuführen..

Diverse Beispiele zeigen wie Sie eine Verbindung zum Microsoft Azure IoT Hub herstellen können.

Beispiel	Produkt	Beschreibung
lotMqttSampleAzureIoT Hub	TF6701	Dieses Beispiel demonstriert, wie Sie MQTT Funktionsbausteine innerhalb der SPS Logik verwenden können, um eine Verbindung zum Microsoft Azure IoT Hub herzustellen und Daten auszutauschen.
<u>Data Agent</u>	TF6720	Dieses Beispiel demonstriert, wie Sie den TwinCAT IoT Data Agent konfigurieren können, um eine Verbindung zum Microsoft Azure IoT Hub herzustellen und Daten auszutauschen.
<u>Data Agent Device Twin</u>	TF6720	Dieses Beispiel demonstriert, wie Sie den TwinCAT IoT Data Agent konfigurieren können, um eine Verbindung zum Microsoft Azure IoT Hub herzustellen und Daten mit dem Device Twin auszutauschen.

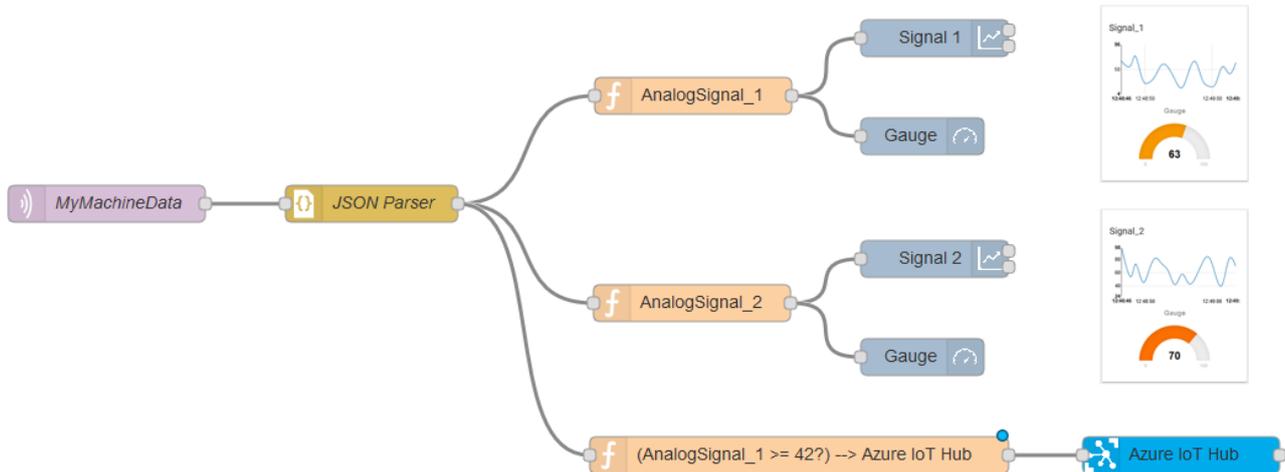
## 4.1.7 Node-RED

Node-RED ist ein beliebtes Tool, das eine workflowbasierte Programmierung ermöglicht, um Hardwaregeräte, APIs und Online-Dienste auf neue und interessante Weise miteinander zu verbinden. Das Tool läuft auf verschiedenen Plattformen, z. B. SPS-Steuerungen, Edge-Geräten, virtuellen Maschinen oder auch nativen Diensten in der Cloud. Es handelt sich um einen einfachen visuellen Open-Source-Editor, um Dienste und Geräte des Internet-of-Things miteinander zu verbinden.

Das System enthält sogenannte „Nodes“ (Knoten), die wie Icons aussehen, die auf der leeren Fläche platziert und per Drag-and-drop miteinander verbunden werden können. Jeder Node bietet eine andere Funktionalität. Beispielsweise gibt es einfache Debug-Nodes, um den Datenfluss zu analysieren/prüfen, oder MQTT-Nodes, die Daten bei einem MQTT-Message-Broker abonnieren und diese Daten in den Workflow einfügen. Um den Funktionsumfang von Node-RED zu erweitern, können Plugins hinzugefügt werden.

Mit Hilfe von TwinCAT IoT können Maschinendaten an Node-RED gesendet und Befehle vom Node-RED-Workflow empfangen werden. Zudem können auch andere TwinCAT-Funktionen verwendet werden, z. B. der TwinCAT TCP/IP Server.

Node-RED unterstützt mehrere Kommunikationsprotokolle, z. B. MQTT, HTTP, Raw TCP/IP usw.



Um TwinCAT IoT über MQTT mit einem Node-RED-Workflow zu verbinden, ist ein Message-Broker als Zwischenstation erforderlich. Eine weitere Alternative wäre die Verwendung von TwinCAT IoT HTTPS/REST mit dem integrierten HTTPS/REST-Webserver von Node-RED.

**Quellen**

- <https://nodered.org/>
- <https://learn.adafruit.com/raspberry-pi-hosting-node-red/what-is-node-red>

**4.2 Eigenschaften**

Die folgende Tabelle gibt einen Überblick über die Eigenschaften des TC3 IoT Data Agent. Die Eigenschaften lassen sich wie folgt kategorisieren:

- Allgemeines: Grundeigenschaften des TC3 IoT Data Agent, z. B. Abtastmuster usw.
- Gates: Unterstützte Cloud-Dienste und Protokolle, z. B. ADS, MQTT, OPC UA usw.
- Data Format: Unterstützte Datenformate beim Senden/Empfangen von Daten an einen/von einem Cloud-Dienst
- Symbol: Unterstützte Datentypen und weitere symbolbezogene Eigenschaften

**Gates**

	<b>Beschreibung</b>	<b>Version</b>
ADS	ADS-Geräte können entweder über symbolischen Zugriff oder die Igrp/Ioff-Kombination verwendet werden. Über ADS kann auf TwinCAT 2/3 I/O-Kanäle sowie TwinCAT 2/3 PLC Laufzeiten, TwinCAT 3 TcCOM-Module und BC Controller zugegriffen werden.	0.7.0.1 und höher
OPC UA	Der TC3 IoT Data Agent kann sich mit jedem OPC UA-Server verbinden, der den Transportkanal opc.tcp unterstützt. Darüber hinaus kann der Kommunikationskanal zum Server durch OPC UA-Zertifikate und Benutzername/Passwort-Authentifizierung gesichert werden.	0.7.0.1 und höher
MQTT	Zum Senden/Empfangen von Daten können allgemeine MQTT-Message-Broker genutzt werden. Der Message-Broker muss den MQTT TCP-Transport unterstützen (derzeit werden keine Web-Socket-Verbindungen unterstützt). TLS 1.2 kann verwendet werden, um den Kommunikationskanal entweder über ein CA-Zertifikat, ein Client-Zertifikat oder über einen PSK zu sichern.	0.7.0.1 und höher
Microsoft Azure IoT Hub	Öffentlicher Cloud-Dienst von Microsoft. Kann zum Senden/Empfangen von Telemetriedaten verwendet werden. Der Kommunikationskanal wird über den primären oder sekundären Schlüssel des IoT Hub-Geräts gesichert. Das Gerät kann auf der Konfigurationswebsite von Azure IoT Hub erstellt werden.  Außerdem kann der Gerätezwillingdienst von Azure IoT Hub genutzt werden, um das Zwillingdokument zu aktualisieren oder gewünschte Updates durch Backend-Anwendungen abzurufen.	0.7.0.1 und höher 1.3.2.1 und höher für Gerätezwillingfunktionen
AWS IoT	Öffentlicher Cloud-Dienst von AWS. Kann zum Senden/Empfangen von Daten verwendet werden. Die Konfiguration basiert auf spezifischen Einstellungen im MQTT-Gate. TLS 1.2 mit einem Client-Zertifikat ist zwingend erforderlich, um den Kommunikationskanal zu sichern. Das Zertifikat ist auf der Konfigurationswebsite von AWS IoT erhältlich.	0.16.2.2 und höher

**Datenformate**

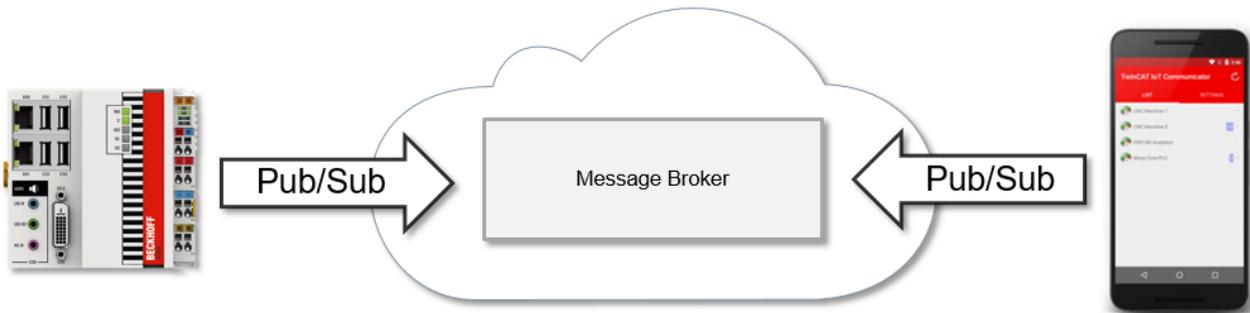
	<b>Beschreibung</b>	<b>Version</b>
ADS binary	Binäres ADS-Datenformat.  Kann zum Senden/Empfangen von Operationen verwendet werden.	0.7.0.1 und höher
Simple JSON	„Flaches“ JSON-Format, das aus einem Zeitstempel und Schlüssel/Wert-Paaren besteht, die die Variablen darstellen.  Kann zum Senden/Empfangen von Operationen verwendet werden.	0.16.2.2 und höher
TwinCAT JSON	Übliches JSON-Datenformat in allen TwinCAT IoT- und Analytics-Produkten sowie dem EK9160. Siehe separaten Dokumentationsartikel über die Spezifikation des TwinCAT IoT JSON-Datenformats.  Kann zum Senden/Empfangen von Operationen verwendet werden.	0.7.0.1 und höher
TwinCAT Analytics	Übliches binäres Datenformat in allen TwinCAT IoT Analytics-Produkten sowie dem EK9160.	0.26.2.8 und höher

**Symbole**

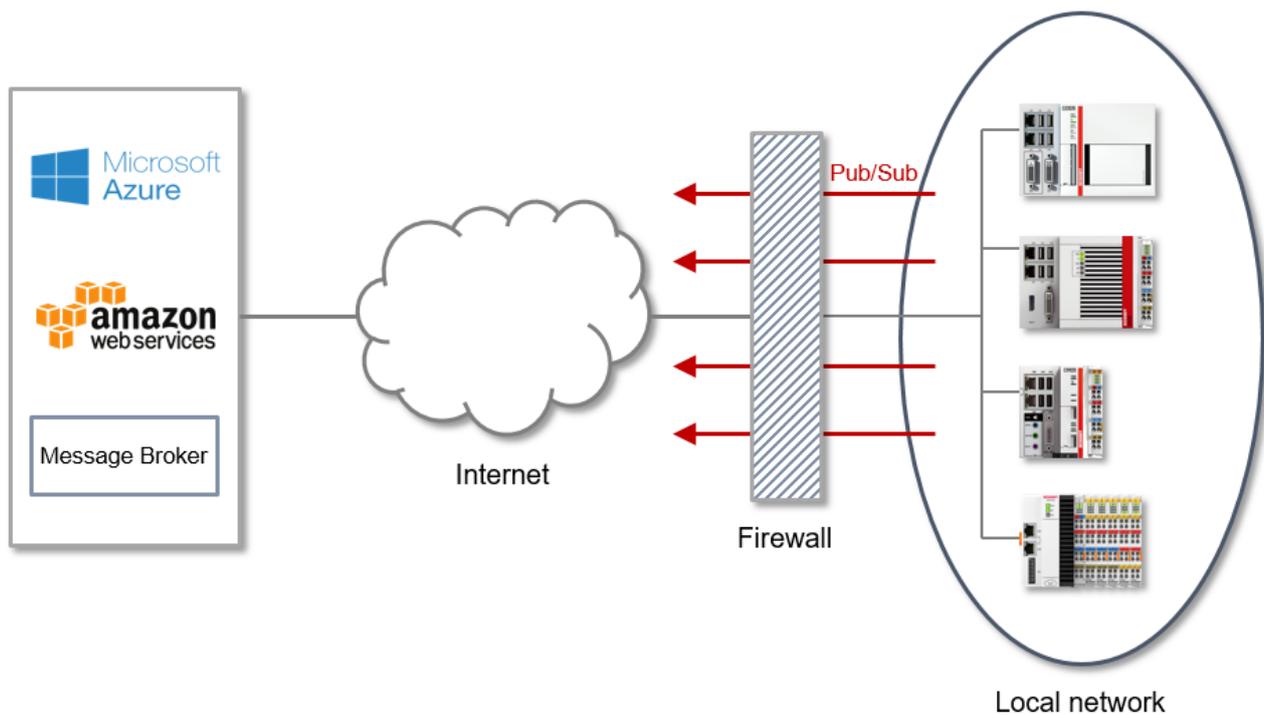
	<b>Beschreibung</b>	<b>Version</b>
Einfache Datentypen	Einfache Datentypen werden unterstützt: Int16, Int32, Int64, UInt16, UInt32, Float, Double, ...	0.7.0.1 und höher
Strings	Strings werden unterstützt	0.7.0.1 und höher
Arrays	Arrays mit einfachen und komplexen Datentypen werden für die Richtung Publish unterstützt	1.2.17.7 und höher
Strukturierte Typen	Strukturierte Typen werden für die Richtung Publish unterstützt	1.2.17.7 und höher
Trigger-Symbol	Trigger-Symbole können verwendet werden, um einen Anwendungsfall „Senden auf Anforderung“ zu erstellen	0.8.2.21 und höher
Typkonvertierung	Typkonvertierung kann verwendet werden, um zwischen verschiedenen Datentypen zu konvertieren	0.8.2.21 und höher
Multi-Link	Verbinden eines Subscriber-Symbols mit mehreren Publisher-Symbolen. (nicht anders herum)	0.24.1.4 und höher
Einheiten	Einheiten werden als Metadaten auf einem Symbol unterstützt. Diese Einstellung hängt vom Datenformat und Gate ab.	0.26.2.8 und höher
Runden von Float-Typen	Das Runden von Float-Typen auf eine bestimmte Anzahl von Dezimalstellen wird unterstützt	1.3.2.1 und höher

**4.3 Internetkonnektivität**

Einer der Hauptvorteile von TwinCAT IoT und dem EK9160 IoT Koppler besteht darin, dass die cloudbezogenen Kommunikationsprotokolle auf Publisher/Subscriber- („Pub/Sub“) und brokerbasierten Konnektivitätsgrundsätzen basieren.



Aus Sicht einer Firewall basiert Publisher/Subscriber nur auf ausgehender Datenkommunikation. Dies ermöglicht nicht nur eine einfache Integration in die vorhandene IT-Infrastruktur, sondern ist auch aus Sicherheitsgründen äußerst praktisch, da zum Senden und Empfangen von Daten keine eingehenden Firewall-Ports geöffnet werden müssen („Portweiterleitung“). Aufgrund dieses technischen Vorteils ist es nicht erforderlich bzw. wird nicht empfohlen, TwinCAT IoT- und EK9160-Produkte am Internet-Edge oder in einem nicht vertrauenswürdigen Netzwerk zu platzieren. Stattdessen können und sollten diese Produkte innerhalb lokaler, vertrauenswürdiger Netzwerkgrenzen – durch eine Firewall vor dem Internet gesichert – verwendet werden.



Die folgende Tabelle gibt einen Überblick über die am häufigsten verwendeten Kommunikationsprotokolle und ihre entsprechenden Netzwerkports. Zu beachten ist, dass diese Portübersicht von der jeweiligen Umgebung und dem Szenario abhängt und auf häufig verwendeten Standardeinstellungen basiert.

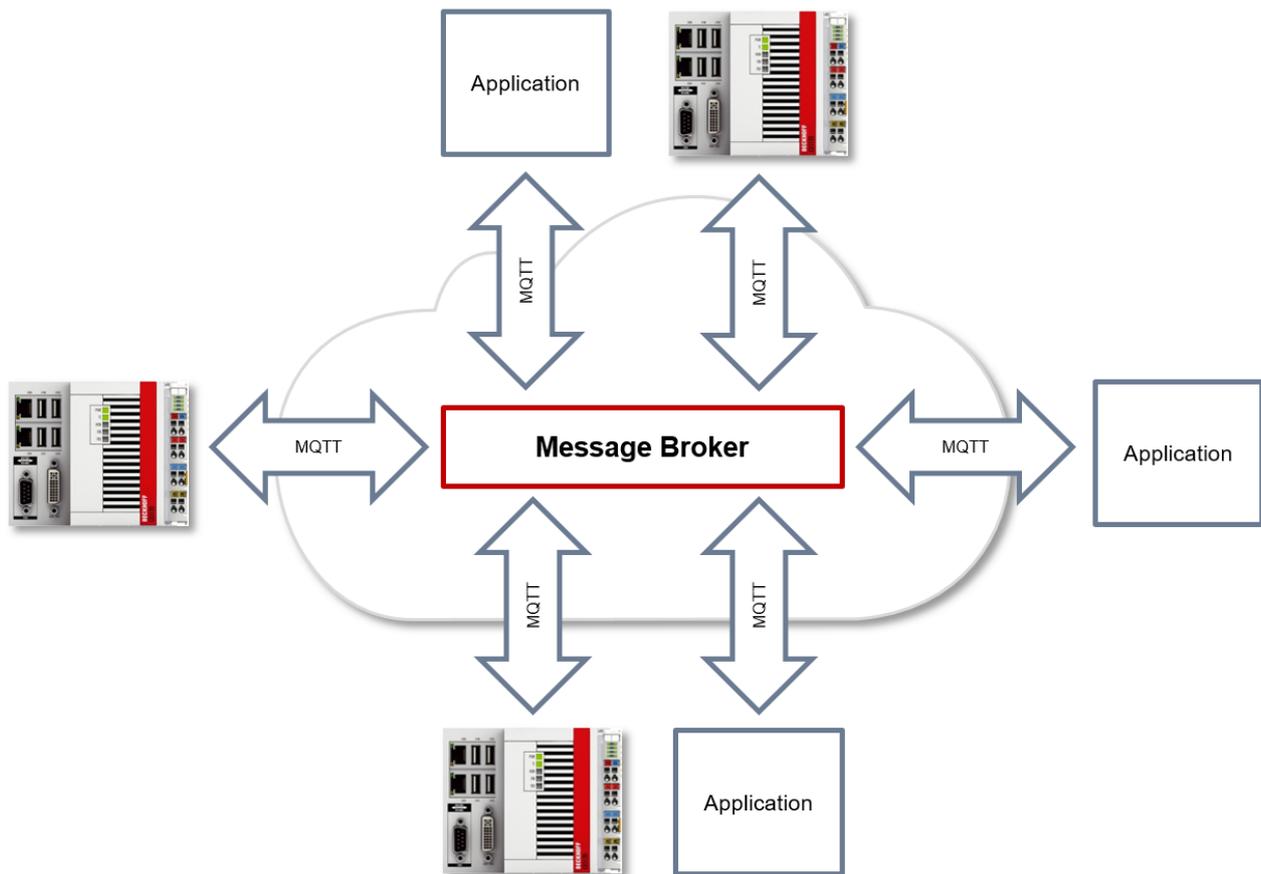
Service	Protokoll	Netzwerkport	Anmerkungen
MQTT-Message-Broker	MQTT	1883/tcp	
MQTT-Message-Broker	MQTT mit TLS	8883/tcp	
Microsoft Azure IoT Hub	AMQP	5671/tcp	
Microsoft Azure IoT Hub	MQTT mit TLS	8883/tcp	
AWS IoT	MQTT mit TLS	8883/tcp	
AWS Greengrass	MQTT mit TLS	8883/tcp	

**● Microsoft Azure IoT Hub**

**i** Der TC3 IoT Data Agent verwendet AMQP bei der Herstellung einer Verbindung mit dem Microsoft Azure IoT Hub.

## 4.4 MQTT Grundlagen

MQTT (Message Queueing Telemetry Transport) ist ein Publisher/Subscriber-basiertes Kommunikationsprotokoll, welches eine Nachrichten-basierte Übertragung zwischen Applikationen ermöglicht. Eine zentrale Komponente bei dieser Art der Übertragung ist der sogenannte Message Broker. Dieser hat die Aufgabe, Nachrichten zwischen den einzelnen Applikationen, bzw. dem Sender und Empfänger einer Nachricht, zu verteilen. Der Message Broker entkoppelt dabei Sender und Empfänger voneinander, sodass diese keine gegenseitigen Addressinformationen kennen und austauschen müssen. Alle Kommunikationsteilnehmer wenden sich beim Senden und Empfangen an den Message Broker und dieser übernimmt die Verteilung der Nachrichten.



**ClientID**

Beim Herstellen einer Verbindung mit dem Message Broker übermittelt der Client eine sogenannte ClientID, welche zur eindeutigen Identifizierung des Clients auf dem Message Broker dient. Der MQTT-Kommunikationstreiber aus TwinCAT3 erzeugt automatisch eine eigene ClientID, welche sich an dem folgenden Namensschema orientiert:

*PlcProjectName-TcMqttClient%n*

%n ist hierbei ein inkrementeller Zähler für die Nummer der jeweiligen MQTT Client Instanz. Jede Instanz des Funktionsbausteins FB\_lotMqttClient erhöht hierbei diesen Zähler. In den meisten Fällen ist das Verwenden dieses ClientID Formats ausreichend. In speziellen Fällen, z.B. abhängig vom Message Broker oder auch durch die eigene MQTT Applikation bedingt, muss eine anwendungsspezifische ClientID vergeben werden. Dies kann über einen entsprechenden Eingang an den Funktionsbausteinen FB\_lotMqttClient und FB\_lotMqtt5Client erfolgen.

Soll eine eindeutige ClientID automatisch beim Start des SPS Projekts generiert werden, so bietet sich die Verwendung einer GUID an, welche über den Funktionsbaustein FB\_CreateGuid aus der Bibliothek Tc2\_System erzeugt werden kann. Der folgende Beispielcode verdeutlicht die Verwendung dieses Funktionsbausteins.

```

PROGRAM MAIN
VAR
  fbGuid : FB_CreateGUID;
  objGuid : GUID;
  sGuid : STRING;
  nState : UINT;
  bStart : BOOL; // set to TRUE to start this sample
END_VAR

CASE nState OF
  0 :
    IF bStart THEN
      bStart := FALSE;
      nState := nState + 1;
    END_IF

```

```

1 : // create GUID using FB_CreateGuid from Tc2_System library
fbGuid(bExecute := TRUE, pGuidBuffer := ADR(objGuid), nGuidBufferSize := SIZEOF(objGuid));
IF NOT fbGuid.bBusy THEN
  fbGuid(bExecute := FALSE);
  IF NOT fbGuid.bError THEN
    nState := nState + 1;
  ELSE
    nState := 255; // go to error state
  END_IF
END_IF

2: // GUID has been created, now convert to STRING
sGuid := GUID_TO_STRING(objGuid);
nState := nState + 1;

3: // done

255: // error state

END_CASE

```

Nach Ausführung dieser State Machine enthält die Variable sGuid die generierte GUID als STRING. Diese kann dann an den Funktionsbausteinen FB\_IotMqttClient und FB\_IotMqtt5Client als ClientID verwendet werden.

## Payload

Der Nachrichteninhalte einer MQTT-Nachricht wird als sogenannter Payload bezeichnet. Es können beliebige Daten übertragen werden, beispielsweise ein Text, ein einzelner Zahlenwert oder eine gesamte Informationsstruktur.

### ● Message-Payload-Formatierung

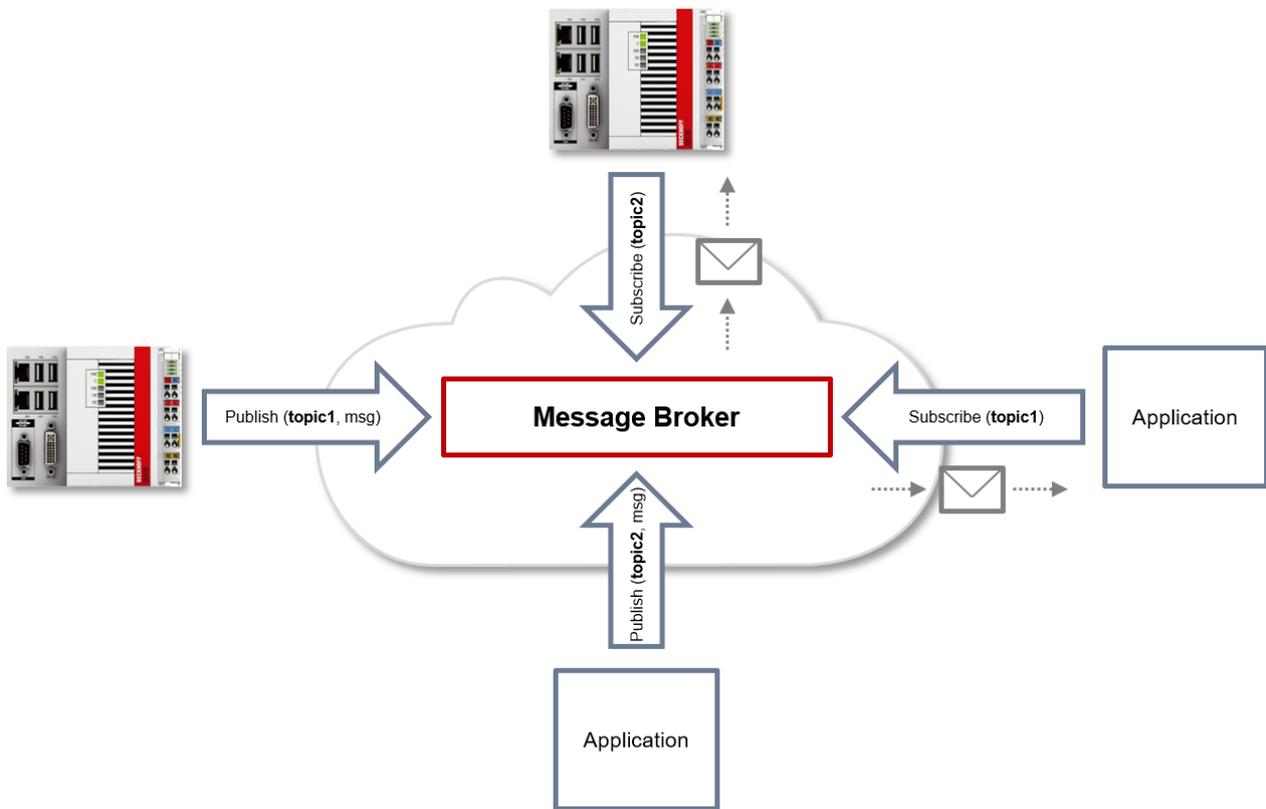
**I** Beachten Sie, dass der Datentyp und die Formatierung des Inhalts der Sender- und Empfängerseite bekannt sein müssen, insbesondere beim Versand von Binärinformationen (Alignment) oder Strings (mit/ohne Nullterminierung).

## Topics

Bei Verwendung eines Message Brokers, welcher auf dem Protokoll MQTT basiert, wird das Senden (Publish) und Abonnieren (Subscribe) von Nachrichten mithilfe sogenannter Topics organisiert. Der Message Broker filtert eingehende Nachrichten anhand dieser Topics für jeden verbundenen Client. Ein Topic kann hierbei auch aus mehreren Ebenen bestehen, wobei die Ebenen durch ein „/“ voneinander getrennt sind.

Beispiel: Campus / Building1 / Floor2 / Room3 / Temperature

Der Publisher einer Nachricht gibt beim Versand immer an, für welches Topic eine Nachricht gedacht ist. Ein Subscriber hingegen gibt an, für welches Topic er sich interessiert. Der Message Broker leitet dann die Nachricht entsprechend weiter.



Beispielkommunikation 1 in der Grafik oben:

- Eine Applikation subscribed sich auf das Topic mit dem Namen „topic1“.
- Ein Controller published eine Nachricht an das Topic mit dem Namen „topic1“.
- Der Message Broker leitet die Nachricht entsprechend an die Applikation weiter.

Beispielkommunikation 2 in der Grafik oben:

- Ein Controller subscribed sich auf das Topic mit dem Namen „topic2“.
- Eine Applikation published eine Nachricht an das Topic mit dem Namen „topic2“.
- Der Message Broker leitet die Nachricht entsprechend an den Controller weiter.

**Wildcards**

Bei der Verwendung von Topics können auch sogenannte „Wildcards“ benutzt werden. Eine Wildcard ersetzt einen Teil des Topics. Ein Subscriber erhält dann ggf. Nachrichten aus mehreren Topics. Es werden zwei Arten von Wildcards unterschieden:

- Single Level Wildcards
- Multi Level Wildcards

Beispiel „Single Level Wildcard“:

Das +-Symbol beschreibt eine Single Level Wildcard. Wird es z.B. vom Subscriber wie folgt verwendet, so werden entsprechende Nachrichten an die Topics entweder vom Subscriber empfangen oder nicht empfangen.

- Der Empfänger subscribed sich auf Campus/Building1/Floor2+/Temperature
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Temperature - OK
- Der Publisher sendet an Campus/Building1/Floor2/Room2/Temperature - OK
- Der Publisher sendet an Campus/Building42/Floor1/Room1/Temperature - NOK
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Fridge/Temperature - NOK

Beispiel „Multi Level Wildcard“:

Das #-Symbol beschreibt eine Multi Level Wildcard. Wird es z.B. vom Subscriber wie folgt verwendet, so werden entsprechende Nachrichten an die Topics entweder vom Subscriber empfangen oder nicht empfangen. Das #-Symbol muss hierbei immer als letztes Symbol im Topic-String verwendet werden.

- Der Empfänger subscribed sich auf Campus/Building1/Floor2/#
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Temperature - OK
- Der Publisher sendet an Campus/Building1/Floor2/Room2/Temperature - OK
- Der Publisher sendet an Campus/Building42/Floor1/Room1/Temperature - NOK
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Fridge/Temperature – OK
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Humidity - OK

### QoS (Quality of Service)

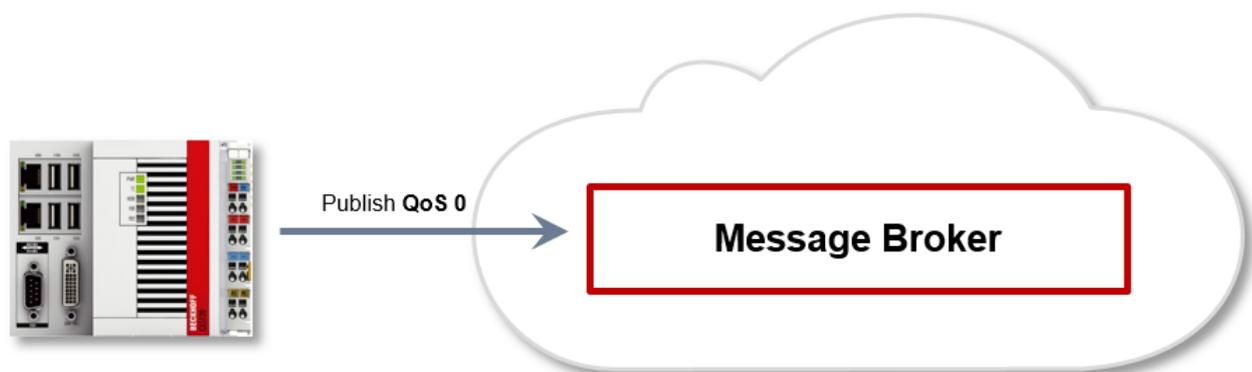
QoS ist eine Vereinbarung zwischen dem Sender und Empfänger einer Nachricht in Bezug auf das Garantieren der Nachrichtenübermittlung. Es existieren drei verschiedene Level in MQTT:

- 0 – höchstens einmal
- 1 – mindestens einmal
- 2 – genau einmal

Beide Kommunikationsarten (Publish/Subscribe) zum Message Broker müssen berücksichtigt und getrennt voneinander betrachtet werden. Das QoS-Level, welches ein Client beim Publishen einer Nachricht verwendet, wird vom jeweiligen Client gesetzt. Wenn der Broker nun die Nachricht an einen Client weiterleitet, der sich entsprechend auf das Topic subscribed hat, wird das QoS-Level vom Subscriber verwendet, welches beim Herstellen der Subscription angegeben wurde. Dies bedeutet, dass ein QoS-Level, welches vom Publisher vielleicht mit 2 angegeben wurde, vom Subscriber mit 0 „überschrieben“ werden kann.

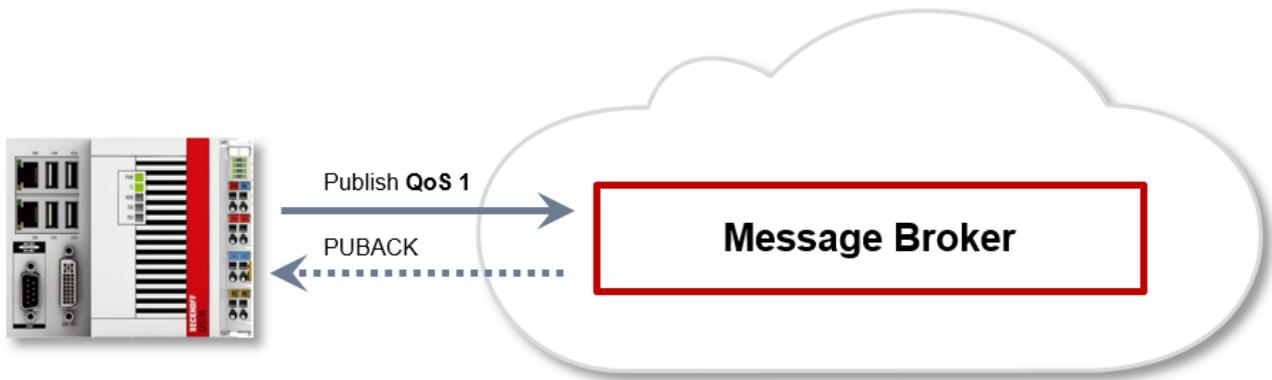
#### QoS-Level 0

Bei diesem QoS-Level erfolgt keine Bestätigung des Empfängers, ob die Nachrichten empfangen wurden oder nicht. In der Folge wird die Nachricht auch kein zweites Mal gesendet.



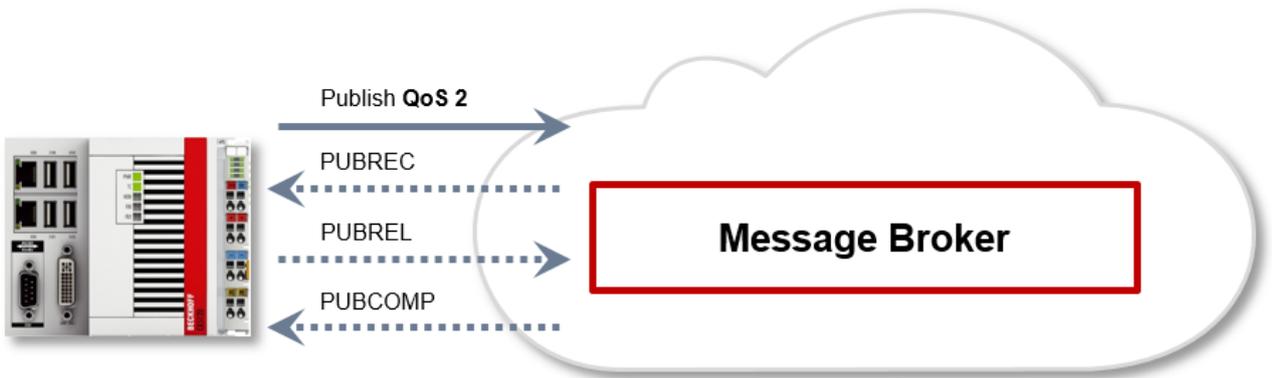
#### QoS-Level 1

Bei diesem QoS-Level wird garantiert, dass die Nachricht zumindest einmal beim Empfänger ankommt. Aber die Nachricht kann unter Umständen auch mehrfach beim Empfänger eintreffen. Der Sender speichert die Nachricht intern bis er eine Bestätigung in Form einer PUBACK-Nachricht vom Empfänger erhält. Wenn die PUBACK-Nachricht für eine bestimmte Zeit ausbleibt, wird die Nachricht erneut gesendet.



**QoS-Level 2**

Bei diesem QoS-Level wird garantiert, dass die Nachricht maximal einmal beim Empfänger ankommt. Dies wird MQTT-seitig durch einen Handshake-Mechanismus realisiert. QoS-Level 2 ist der sicherste (aus Sicht der Nachrichtenübermittlung), aber auch langsamste QoS-Level. Wenn ein Empfänger eine Nachricht mit QoS-Level 2 erhält, bestätigt er die Nachricht mit einem PUBREC. Der Absender der Nachricht merkt sich diese intern bis er ein PUBCOMP empfangen hat. Dieser zusätzliche Handshake (verglichen mit QoS 1) ist wichtig, damit die Nachricht nicht doppelt übertragen wird. Wenn der Absender der Nachricht ein PUBREC erhält, kann er den initialen Publish verwerfen, da er weiß, dass die Nachricht einmal vom Empfänger empfangen wurde. Er merkt sich somit intern den PUBREC und sendet seinerseits ein PUBREL. Nachdem der Empfänger ein PUBREL empfangen hat, kann er die sich zuvor gemerkten Zustände verwerfen und mit einem PUBCOMP antworten. Umgekehrt genauso. Immer dann, wenn ein Paket verloren geht, ist der jeweilige Kommunikationsteilnehmer dafür verantwortlich, die zuletzt gesendete Nachricht nach einer bestimmten Zeit noch einmal zu senden.



Der LastWill ist eine Nachricht, die im Falle eines irregulären Verbindungsabbruches vom Broker an alle Clients gesendet wird, die das passende Topic abonniert haben. Verliert der MQTT-Client in der SPS die Verbindung zum Broker und es wurde beim Verbindungsaufbau ein LastWill hinterlegt, so wird dieser LastWill vom Broker kommuniziert, ohne dass der Client sich darum kümmern muss.

Bei einem geplanten Disconnect wird der LastWill laut Spezifikation nicht zwingend übertragen. Aus Sicht des SPS-Programmierers kann dieser entscheiden, ob er vor Aufruf des Disconnects den LastWill publishen will. Dazu wird auf dem LastWill-Topic die LastWill-Nachricht noch einmal gepublished. Das ist notwendig, da der Broker aufgrund der regulären Verbindungsabbruches die Last Will-Nachricht nicht veröffentlichen würde.

Bei einem TwinCAT-Kontextwechsel und einem daraus folgenden Neustart der MQTT-Kommunikation sendet der IoT-Treiber den vorher spezifizierten LastWill an den Broker, weil in diesem Moment aus der SPS keine Möglichkeit mehr dazu besteht. Wenn bei Verbindungsherstellung kein LastWill definiert wurde, wird auch keine Nachricht vor dem Disconnect übertragen.

**Sicherheit**

Bei der Herstellung einer Verbindung zum Message Broker können hierbei auch Sicherheitsmechanismen, wie TLS eingesetzt werden, um die Kommunikationsverbindung zu verschlüsseln oder eine Authentifizierung zwischen Client und Message Broker zu realisieren.

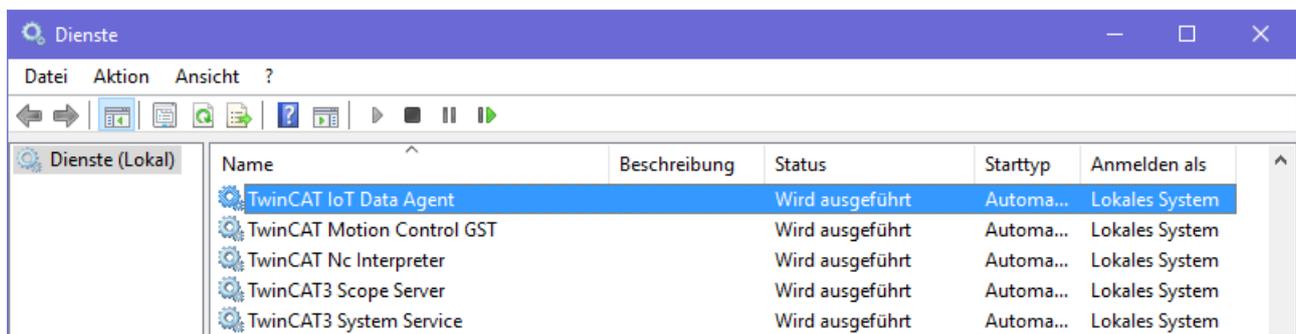
## Quellen

Für weitere und detailliertere Informationen zu MQTT empfehlen wir die folgenden Webseiten:

HiveMq Blog: <http://www.hivemq.com/blog/mqtt-essentials/> (Hauptgrundlage dieses Artikels)

## 4.5 Startmodi

Der TC3 IoT Data Agent wird automatisch als Windows-Dienst installiert, der im lokalen Systemkontext läuft (und daher vom angemeldeten Benutzerkonto unabhängig ist) und beim Hochfahren des Systems automatisch startet.



Nach dem Start des TC3 IoT Data Agent wartet die Kernanwendung auf einen Auslösebefehl, um die Kommunikation zu starten. Der Auslösebefehl wird üblicherweise über den Konfigurator oder die Taskleistanwendung gesendet (siehe [Konfigurator \[▶ 41\]](#) und [Taskleiste \[▶ 56\]](#)). Mit Hilfe des grafischen Konfigurator-Tools kann auch eine AutoStart-Option pro Konfigurationsdatei konfiguriert werden.

### Befehlszeile

Alternativ kann der TC3 IoT Data Agent als unabhängige Anwendung über die Windows-Befehlszeile ausgeführt werden. Starten Sie dazu die ausführbare Datei des TC3 IoT Data Agent (*TclotDataAgent.exe*) im Installationsverzeichnis. Sie können verschiedene Startoptionen einstellen, indem Sie die Parameter für den Prozessaufruf festlegen, z. B. einen Pfad zur Konfigurationsdatei oder Protokollierungseinstellungen. Verwenden Sie den Parameter „-h“ für eine Liste aller Startoptionen.

Beispiele:

Prozessaufruf	Beschreibung
TclotDataAgent.exe	Die Konfigurationsdatei <i>TclotDataAgentConfig.xml</i> aus dem Standard-Konfigurationsverzeichnis %TC_BOOTDIR% wird verwendet. Wartet auf einen Auslösebefehl, um die Kommunikation zu starten.
TclotDataAgent.exe -a	Startet die Kommunikation nach dem Start automatisch. Nimmt die Konfigurationsdatei automatisch aus %TC_BOOTDIR%.
TclotDataAgent.exe -c <i>PathToConfig</i>	Legt die zu verwendende Konfigurationsdatei fest. Wartet auf einen Auslösebefehl, um die Kommunikation zu starten.
TclotDataAgent.exe -a -c <i>PathToConfig</i>	Legt die zu verwendende Konfigurationsdatei fest. Startet die Kommunikation automatisch, ohne auf einen Auslösebefehl zu warten.
TclotDataAgent.exe -l VERBOSE	Die Konfigurationsdatei wird aus dem TwinCAT-Bootverzeichnis abgerufen und die Protokollierung wird im ausführlichen Modus aktiviert. Die Windows-Befehlszeile wird als Protokollausgabe verwendet.
TclotDataAgent.exe -l VERBOSE -o <i>PathToFile</i>	Die Konfigurationsdatei wird aus dem TwinCAT-Bootverzeichnis abgerufen und die Protokollierung wird im ausführlichen Modus aktiviert. Eine Datei wird als Protokollausgabe verwendet.

## 4.6 Sicherheit

Abhängig vom Funktionsumfang jedes Gates unterstützt der TC3 IoT Data Agent mehrere Sicherheitsmechanismen zur Sicherung der Datenübertragung, beispielsweise die Verschlüsselung von Nachrichten und Client/Server-Authentifizierung.

Die folgende Tabelle gibt einen Überblick über die verschiedenen Sicherheitsmechanismen in jedem unterstützten Gate.

Gate	Sicherheitsmechanismen	Beschreibung
ADS	keiner	ADS-Kommunikation sollte als nicht sicher behandelt werden.
OPC UA	Client/Server-Authentifizierung Benutzername/Passwort-Authentifizierung Autorisierung Datenverschlüsselung	Abhängig vom Funktionsumfang des OPC UA-Servers, mit dem sich der TC3 IoT Data Agent verbinden soll, können die folgenden Sicherheitsmechanismen verwendet werden: <ul style="list-style-type: none"> <li>• Client/Server-Authentifizierung kann über X.509-Zertifikate (öffentlicher/privater Schlüssel) und den Aufbau einer vertrauenswürdigen Beziehung zwischen den Kommunikationsteilnehmern verwendet werden.</li> <li>• Benutzername/Passwort-Authentifizierung kann von einem OPC UA-Client für die Verbindung mit einem OPC UA-Server verwendet werden.</li> <li>• Autorisierung kann auf dem Server verwendet werden, um Zugriffsebenen auf OPC UA-Nodes zu definieren, d. h. um zu konfigurieren, welche Benutzeridentität auf welche Nodes auf einem OPC UA-Server zugreifen darf.</li> <li>• Datenverschlüsselung kann für die Verschlüsselung von OPC UA-Nachrichten während der Übertragung verwendet werden.</li> </ul>
MQTT	Client/Server-Authentifizierung Benutzername/Passwort-Authentifizierung Autorisierung Datenverschlüsselung	Abhängig vom Funktionsumfang des MQTT-Message-Brokers, mit dem sich der TC3 IoT Data Agent verbinden soll, können die folgenden Sicherheitsmechanismen verwendet werden. <ul style="list-style-type: none"> <li>• Client/Server-Authentifizierung kann über X.509-Zertifikate (öffentlicher/privater Schlüssel) und den Aufbau einer vertrauenswürdigen Beziehung zwischen den Kommunikationsteilnehmern (Client und Message-Broker) verwendet werden</li> <li>• Benutzername/Passwort-Authentifizierung kann von einem Client für die Verbindung mit dem Message-Broker verwendet werden</li> <li>• Autorisierung kann auf dem Message-Broker verwendet werden, um Zugriffsebenen auf Topics zu definieren, d. h. um zu konfigurieren, welche Benutzeridentität auf welches Topic auf dem Message-Broker zugreifen darf</li> <li>• Datenverschlüsselung kann für die Verschlüsselung von Nachrichten während der Übertragung verwendet werden.</li> </ul> Der TC3 IoT Data Agent verwendet TLS Version 1.2 für die Sicherung des Kommunikationskanals.
Microsoft Azure IoT Hub	Geräteauthentifizierung Datenverschlüsselung	Jeder Azure IoT Hub-Client muss auf der IoT Hub-Instanz als Gerät registriert werden. Während der Geräteregistrierung werden eine „Deviceld“ und ein „SharedAccessKey“ generiert, die während der Verbindungsherstellung zur Authentifizierung des Geräts bei der IoT Hub-Instanz verwendet werden müssen. Darüber hinaus werden Nachrichten während der Übertragung verschlüsselt.
AWS IoT	Geräteauthentifizierung Autorisierung Datenverschlüsselung	Jeder AWS IoT-Client muss als „Thing“ (Objekt) registriert werden. Ein Objekt/Gerät wird mit einem X.509-Zertifikat konfiguriert, das das Gerät beim AWS IoT authentifiziert, und kann auch mit Sicherheitsrichtlinien verknüpft werden, die das Gerät dazu autorisieren, bestimmte Aktionen durchzuführen. Darüber hinaus werden Nachrichten während der Übertragung verschlüsselt. Für die Sicherung des Kommunikationskanals wird TLS Version 1.2 verwendet.

## 4.7 Puffern von Daten

Für kurzzeitige Netzwerkausfälle bietet der Data Agent eine Puffer-Funktionalität. Dabei ist dringend zu beachten, dass das Puffern ausschließlich im Arbeitsspeicher stattfindet. Dementsprechend sollte bei dieser Puffer -Funktionalität immer der Arbeitsspeicher im Auge behalten werden. Ist eine Verbindung zum Ziel wiederhergestellt, beginnt der Data Agent mit der Abarbeitung des Puffers.

Ein weiterer zu beachtender Punkt sind die Restriktionen eines Message Brokers, an den Daten gesendet werden. Es gibt Message Broker, die Limitationen auf eine gewisse Anzahl von Nachrichten pro Zeiteinheit haben. Es ist während des Konzipierens einer Data Agent-Anwendung darauf zu achten, dass diese Obergrenze beim Abarbeiten eines Puffers nicht überschritten wird.

### Konfiguration im Data Agent

Ein Publisher-Channel bietet die Möglichkeit, die Größe eines Puffers einzustellen. Diese Größe bezieht sich auf die Anzahl der zu puffernden Nachrichten.

An diesem Publisher-Channel gibt es desweiteren die Einstellungsmöglichkeit der InhibitTime. Diese besagt, mit welchem Zeitintervall die Nachrichten aus dem Puffer geholt werden sollen sobald die Verbindung zum Ziel wiederhergestellt wurde. Bei einem zyklischen Publisher-Channel **muss** die InhibitTime kleiner sein als die CycleTime, da der Puffer sonst nie geleert werden kann. Bei einem OnChange- oder Trigger-Channel gibt die InhibitTime die Zykluszeit an, mit der Daten aus dem Puffer geholt und an den Message Broker gepublished werden. Wird die InhibitTime mit 0 eingestellt, werden die Nachrichten vom Data Agent so schnell wie möglich an das Ziel versendet.

Wenn während der Puffer-Abarbeitung neue Nachrichten ankommen, werden diese an das Ende des Puffers gehängt. Dadurch wird die richtige Reihenfolge der Daten gewährleistet.

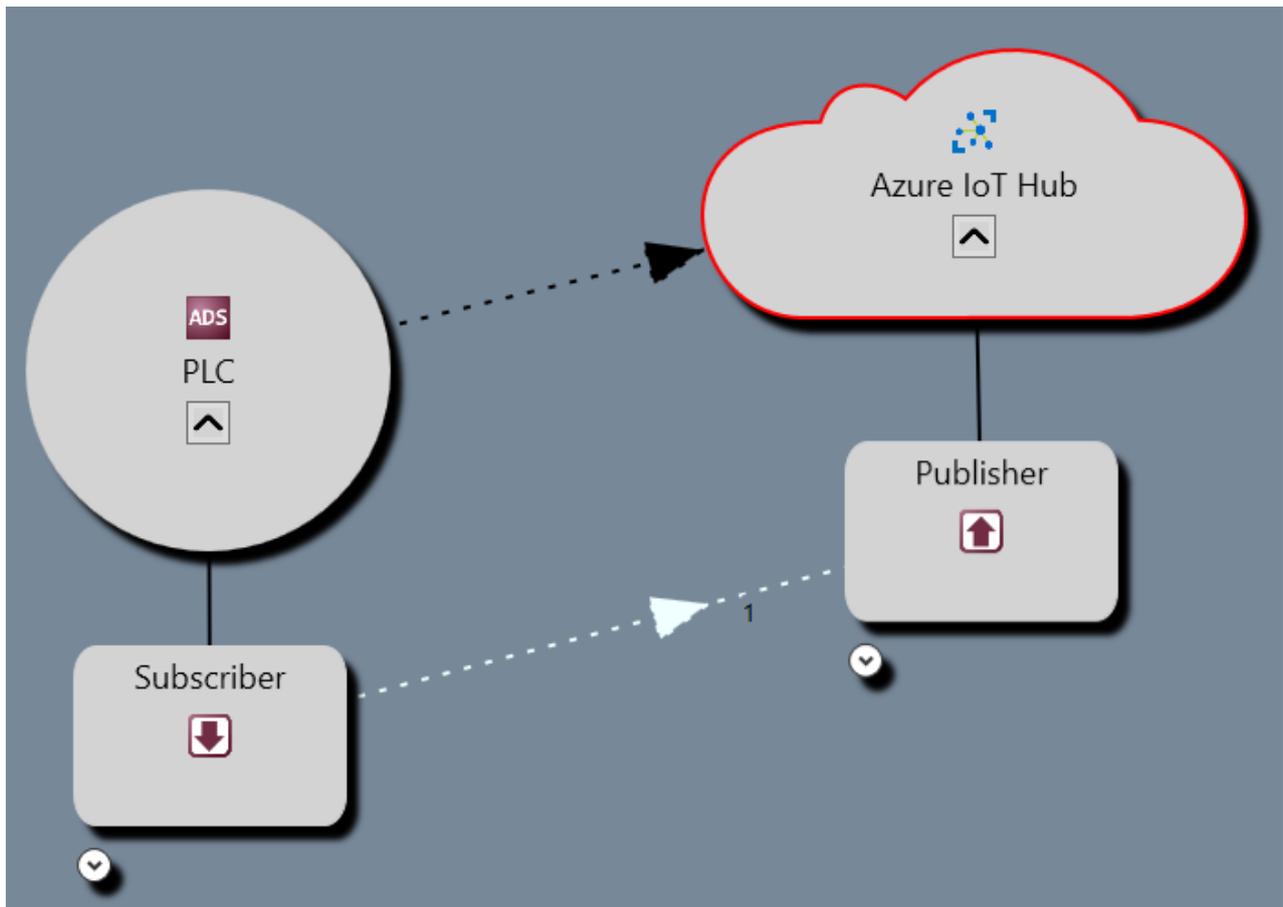
## 4.8 Unterstützte Protokolle und Dienste

### 4.8.1 Microsoft Azure IoT Hub

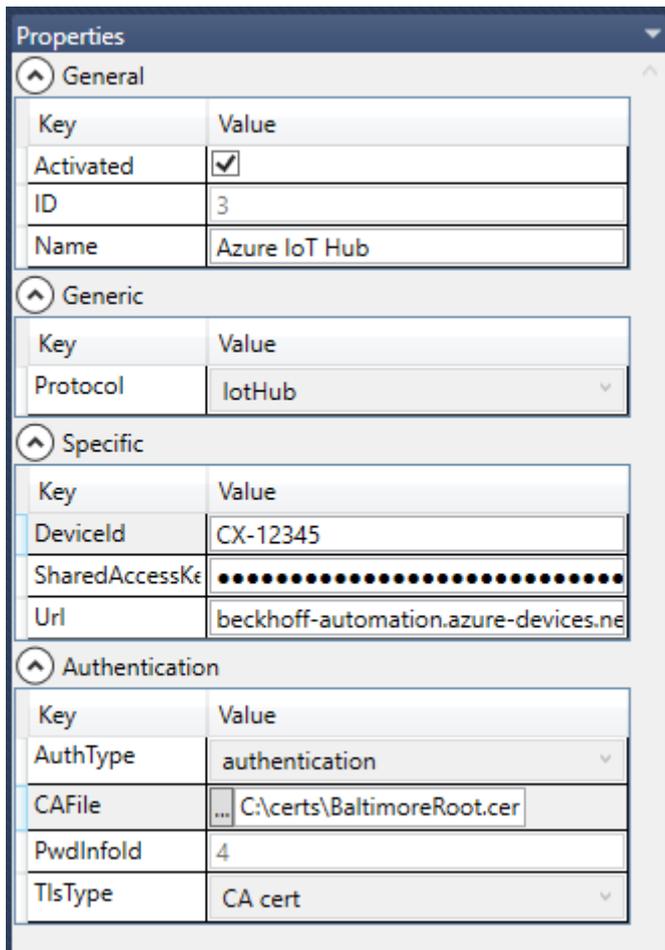
Bei dem Microsoft Azure IoT Hub handelt es sich um eine IoT Suite in der Azure Cloud, welche diverse Dienste anbietet, um IoT-Geräte mit Diensten auf Azure zu verbinden. Microsoft Azure IoT Hub ermöglicht es Geräten, eine sichere, bidirektionale Datenkommunikation aufzubauen, um Telemetriedaten zu senden, Befehle zu empfangen oder einen digitalen Zwilling des Devices zu erhalten.

### Konfiguration in TwinCAT IoT Data Agent

Verbindungen zum Microsoft Azure IoT Hub Dienst können einfach im TwinCAT IoT Data Agent Configurator konfiguriert werden und werden durch einen eigenen Gate-Typ dargestellt.



Die Gate-Konfiguration enthält alle erforderlichen Parameter, um eine Verbindung zum Azure IoT Hub herzustellen, z. B. DeviceId, PrimaryKey/SecondaryKey, Hostname und einen Link zu einem vertrauenswürdigen CA-Zertifikat für die Verbindung.

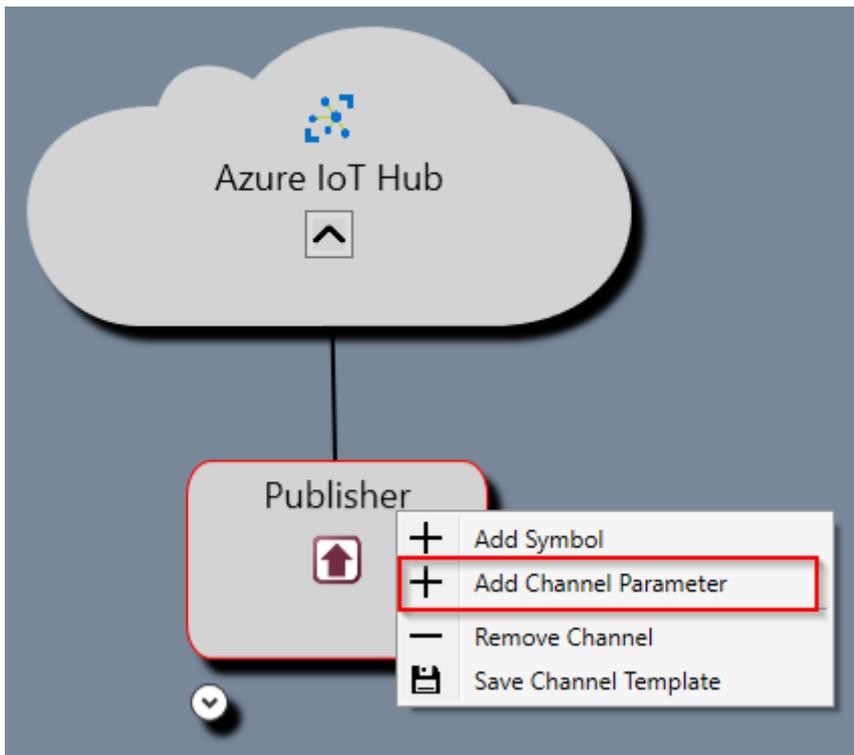


**Sendedaten**

**Empfangsdaten**

**Eigenschaften**

Microsoft Azure IoT Hub bietet die Möglichkeit, Telemetrie-Metadaten von Nachrichten zu senden. Solche Daten können als so genannte "Eigenschaften" gesendet werden. Im TwinCAT IoT Data Agent Configurator können Eigenschaften als Kanalparameter im Parameter Editor eingestellt werden.



Parameter Editor

**Publisher**

DataType	Name	Value
String	Company	Beckhoff Automation
String	Device	CX5130-0020
String	OS	Windows 10
String	OS Build	1909

Eine Cloud-Anwendung (z. B. Azure Stream Analytics) kann dann diese Parameter dann zur weiteren Verarbeitung übernehmen.

```
3/19/2021 3:54:51 PM> Device: [CX-12345], Data:
[{"Timestamp":"2021-03-19T14:54:51.1650000+01:00","GroupName":"Publisher","MAIN.nCounter":49510}] Properties:
'Company': 'Beckhoff Automation'
'Device': 'CX5130-0020'
'OS': 'Windows 10'
'OS Build': '1909'
```

**DeviceTwin**

## 5 Konfiguration

### 5.1 Schnellstart

In diesem Abschnitt wird beschrieben, wie man den TC3 IoT Data Agent schnell startet.

Das Kommunikationsszenario ist ein typischer Anwendungsfall für den Data Agent: Variablen von einem Southbound-Gerät (TwinCAT PLC-Laufzeit) werden abgetastet und an einen Northbound-Dienst (MQTT-Message-Broker) veröffentlicht. In diesem Beispiel läuft der Message-Broker auf demselben Computer wie die PLC-Laufzeit.

Für die Einrichtung dieses Szenarios sind folgende Schritte erforderlich. Alle Schritte werden auf demselben Computer durchgeführt.

- [Installieren der Software \[► 39\]](#)
- [Vorbereiten des SPS-Programms \[► 39\]](#)
- [Konfigurieren des TC3 IoT Data Agents \[► 40\]](#)
- [Anzeigen der veröffentlichten Nachrichten \[► 41\]](#)

#### Installieren der Software

1. Installieren Sie den TF6720 IoT Data Agent wie im Abschnitt [Installation \[► 11\]](#) beschrieben.
2. Installieren Sie den Mosquitto MQTT Message Broker mit seinen Standard-Setup-Einstellungen. Der Message-Broker kann überall installiert werden und wird über seinen Hostnamen oder seine IP-Adresse verbunden.
3. Öffnen Sie eine Windows-Konsole, navigieren Sie zum Installationsverzeichnis des Mosquitto MQTT-Message-Brokers (in der Regel *C:\Program Files (x86)\mosquitto*) und starten Sie den Broker im ausführlichen Modus, indem Sie den folgenden Befehl ausführen: `mosquitto.exe -v`

#### ● Mosquitto MQTT-Message-Broker



Der Mosquitto MQTT-Message-Broker kann von [www.mosquitto.org](http://www.mosquitto.org) heruntergeladen werden. Folgen Sie unbedingt den Installationsanweisungen des Mosquitto-Installers.

#### ● Start von Mosquitto



Der Mosquitto MQTT-Message-Broker wird standardmäßig als Windows-Dienst installiert. Sie können auch die Datei `mosquitto.exe` aus einer Befehlszeile starten und Parameter wie „-v“ für ausführliche Ausgabe festlegen. Wenn Sie den Broker aus einer Befehlszeile starten möchten, stellen Sie sicher, dass der Mosquitto Windows-Dienst nicht läuft.

#### Vorbereiten des SPS-Programms

Richten Sie ein TwinCAT 3 PLC-Projekt ein und aktivieren Sie Testlizenzen für TC1200 (PLC) und TF6720 (TC3 IoT Data Agent). In diesem Beispiel sollte das SPS-Programm die folgenden Variablen enthalten:

```
PROGRAM MAIN
VAR
    nCounter1 : INT;
    nCounter2 : LREAL;
END_VAR
```

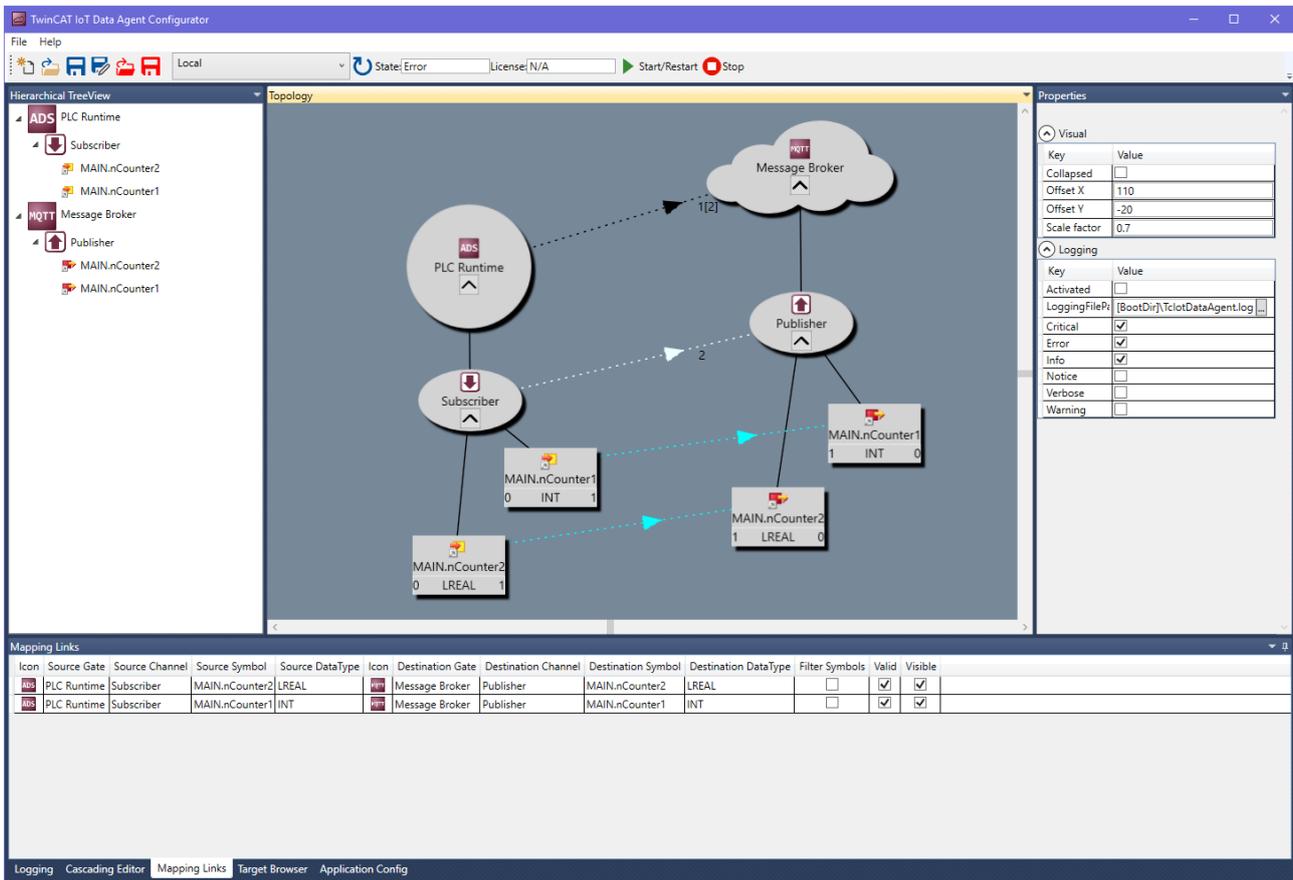
Diese zwei Variablen sollten zyklisch inkrementiert werden. Fügen Sie den folgenden Implementierungscode für das SPS-Programm hinzu:

```
nCounter1 := nCounter1 + 1;
nCounter2 := nCounter2 + 0.1;
```

Aktivieren Sie das Projekt und starten Sie TwinCAT im Run-Modus. Stellen Sie sicher, dass das SPS-Programm läuft, indem Sie sich anmelden.

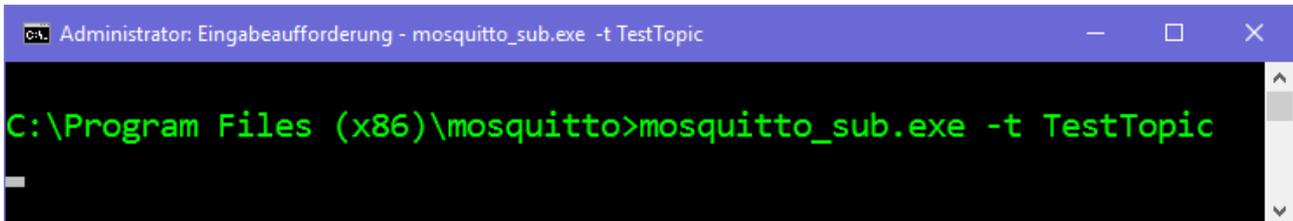
## Konfigurieren des TwinCAT IoT Data Agents

1. Starten Sie den TwinCAT IoT Data Agent-Konfigurator entweder über den Link im Windows-Startmenü oder über die Taskleistanwendung (siehe [Konfigurator \[► 41\]](#)).
2. Folgen Sie den nachstehenden Anweisungen Schritt für Schritt, um eine Data Agent-Konfiguration zu erstellen. Alternativ können Sie auch die Konfigurationsdatei „Sample\_QuickStart.xml“ öffnen, die im Installationsverzeichnis des Data Agent zu finden ist, und direkt mit Schritt (m) fortfahren.
3. Starten Sie eine neue Konfiguration, indem Sie auf das Icon **Neu** in der Konfigurator-Symbolleiste klicken.
4. Rechtsklicken Sie in der Topologieansicht auf einen leeren Bereich und wählen Sie **Add Gate (ADS)** aus.
5. Wählen Sie das neue Gate aus und stellen Sie sicher, dass die lokale AmsNetId und ADS-Port 851 im Eigenschaften-Fenster ausgewählt sind
6. Rechtsklicken Sie in der Topologieansicht auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
7. Wählen Sie den neuen Kanal aus und stellen Sie sicher, dass der Abtastmodus „cyclic“ und die Zykluszeit „1000“ im Eigenschaften-Fenster ausgewählt sind.
8. Öffnen Sie den Target Browser und navigieren Sie zu Ihrer lokalen PLC-Laufzeit (Port 851).
9. Wählen Sie die Variablen „nCounter1“ und „nCounter2“ aus und ziehen Sie sie zum Subscriber-Kanal in der Topologieansicht.
10. Rechtsklicken Sie in der Topologieansicht auf einen leeren Bereich und wählen Sie **Add Gate (MQTT)** aus.
11. Wählen Sie das neue Gate aus und stellen Sie sicher, dass die URL auf 127.0.0.1 gesetzt ist und Port 1883 verwendet wird.
12. Rechtsklicken Sie in der Topologieansicht auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
13. Wählen Sie den neuen Kanal aus und stellen Sie sicher, dass der Abtastmodus auf „cyclic“ eingestellt ist. Stellen Sie die Zykluszeit auf „1000“ und den Formatter auf „TwinCAT JSON“ ein. Das MQTT-Topic kann auf ein Topic Ihrer Wahl eingestellt werden, z. B. „TestTopic“.
14. Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal, um eine Zuordnung zwischen den Quell- und Zielsymbolen zu erstellen. Beachten Sie, dass die Zielsymbole automatisch erzeugt werden.
15. Starten Sie eine neue Konfiguration, indem Sie auf die Schaltfläche **Activate Configuration** in der Symbolleiste klicken.

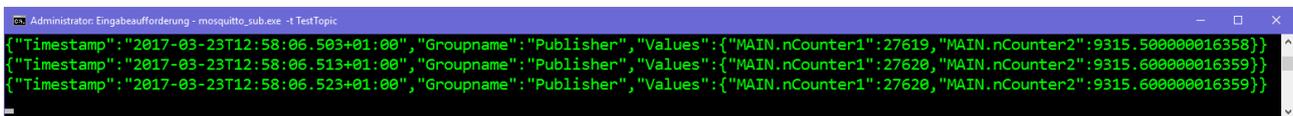


**Anzeigen der veröffentlichten Nachrichten**

Öffnen Sie ein Befehlszeilenfenster und navigieren Sie zum Installationsverzeichnis des Mosquitto MQTT-Message-Brokers (in der Regel „C:\Program Files\mosquitto“ oder „C:\Program Files (x86)\mosquitto“). Starten Sie das Mosquitto Subscriber-Tool, indem Sie den folgenden Befehl ausführen: `mosquitto_sub.exe -t TestTopic`.



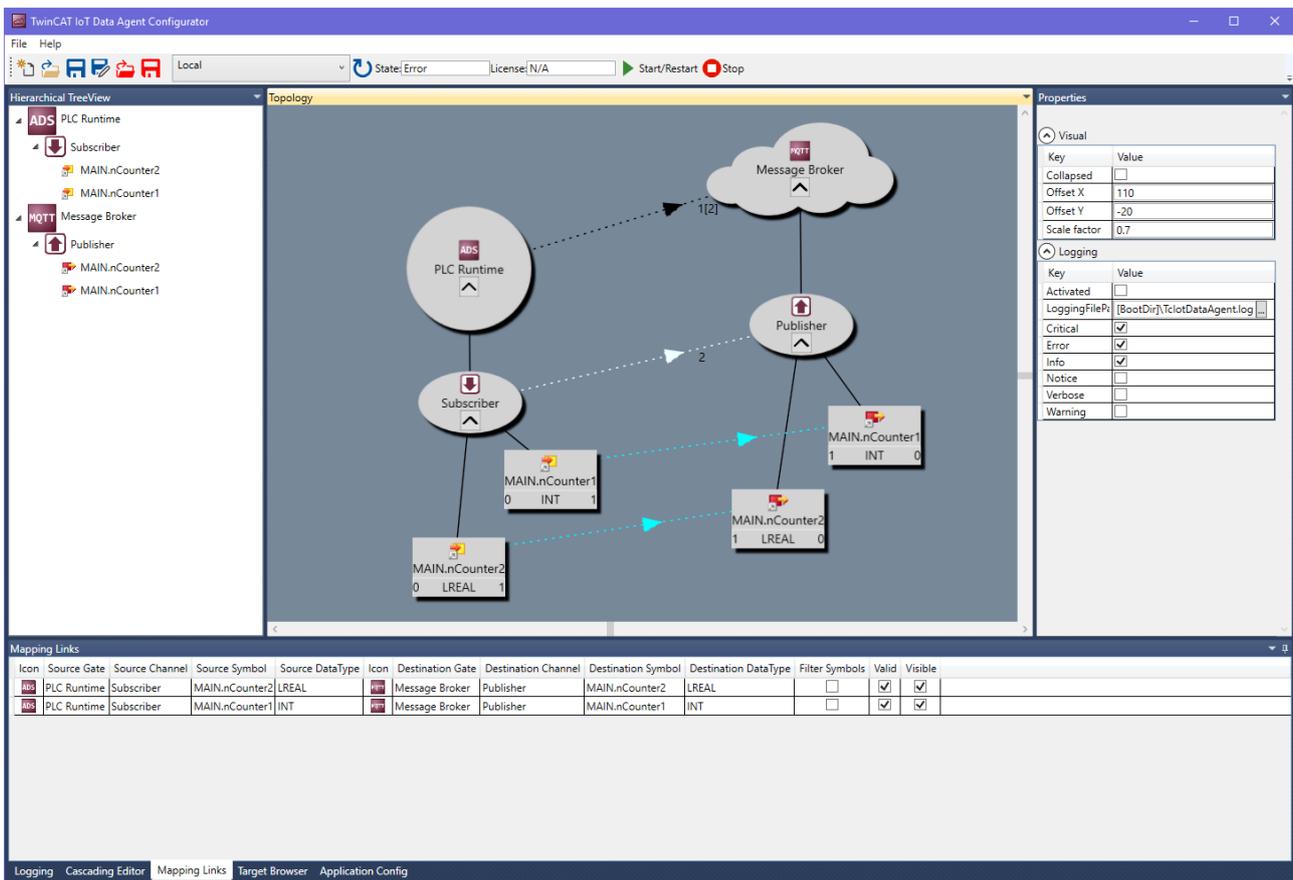
Das Mosquitto Subscriber-Tool erhält nun die Nachrichten, die vom TwinCAT IoT Data Agent veröffentlicht werden.



**5.2 Konfigurator**

Der TC3 IoT Data Agent-Konfigurator ist eine benutzerfreundliche grafische Benutzeroberfläche, die die XML-Konfigurationsdatei abstrahiert und eine moderne Schnittstelle bietet, die alle Funktionen zum einfachen Konfigurieren von Symbolen enthält, die an einem Cloud-Dienst gesendet oder von einem solchen Dienst empfangen werden sollen.

Der Konfigurator wird auch für die Konfiguration von TC3 IoT-Funktionen verwendet.



### Standardkomponenten

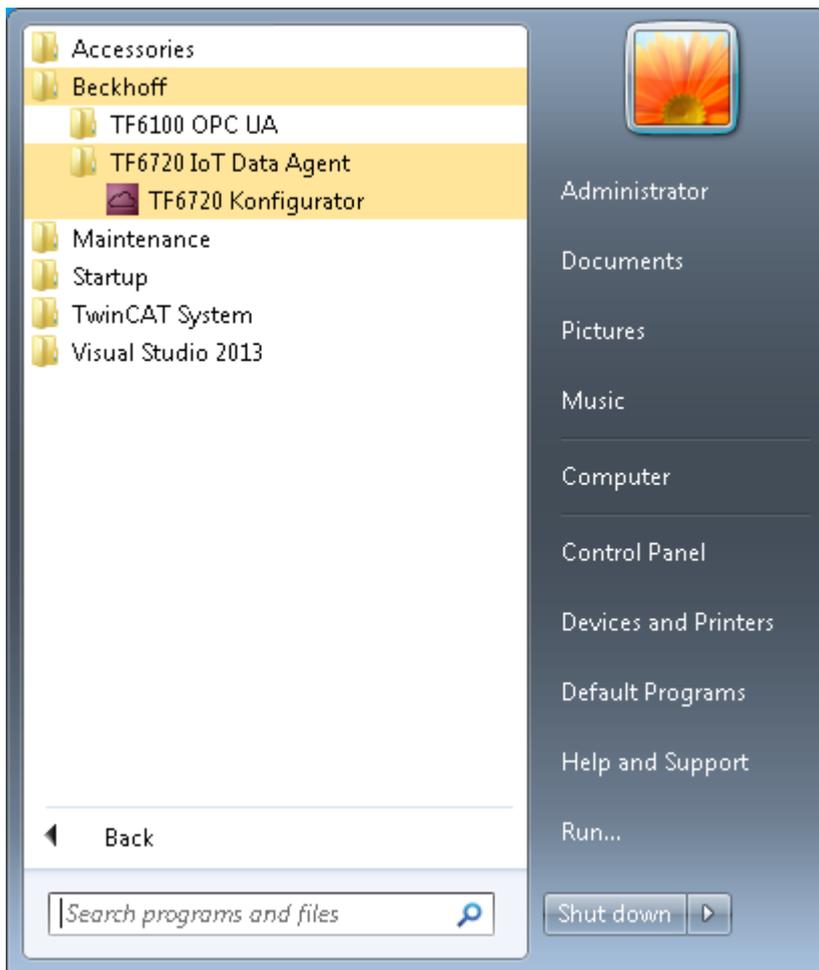
Der TC3 IoT Data Agent-Konfigurator besteht aus den folgenden Bereichen:

<b>Menü und Symbolleiste</b>	Enthält Befehle zum Speichern und Öffnen von Dateien und zum Starten und Stoppen der Anwendung
<b>Hierarchische Baumansicht</b>	Gibt einen hierarchischen Überblick über die Konfiguration, um eine Konfiguration zu erstellen und zu bearbeiten
<b>Topologieansicht</b>	Gibt einen grafischen Überblick über die Konfiguration, um eine Konfiguration zu erstellen und zu bearbeiten
<b>Eigenschaften-Fenster</b>	Zeigt die Eigenschaften einer aktivierten Komponente in der Topologie- oder Baumansicht an.
<b>Logging</b>	Stellt Protokollinformationen vom Konfigurator bereit.
<b>Cascading Editor</b>	Hilft bei großen und komplexen Navigationen, indem Filtermechanismen für Symbole bereitgestellt werden
<b>Mapping Links</b>	Gibt einen Überblick über alle Verbindungen zwischen Symbolen in der Konfiguration
<b>Target Browser</b>	Dient für den symbolischen Zugriff für ADS- und OPC UA-Ziellaufzeiten

### Installation

Der Konfigurator wird vom Setup automatisch installiert und ist als Verknüpfung im Windows-Startmenü verfügbar.

Beim ersten Starten des Konfigurators bittet dieser um die Erstellung eines OPC UA-Client-Zertifikats. Dieses Zertifikat wird vom Konfigurator in seinem integrierten OPC UA Target Browser verwendet, um sich mit einem Server zu verbinden und dessen Namensraum zu durchsuchen. Nachdem das Zertifikat erstellt wurde, wird die Konfigurator-Benutzeroberfläche angezeigt.



## 5.2.1 Topologieansicht

Die Topologieansicht (oder „leere Fläche“) ist der zentrale grafische Konfigurationsbereich des TC3 IoT Data Agent-Konfigurators. Sie zeigt die folgenden Komponenten einer Konfiguration:

- Die konfigurierten Gates, Kanäle und Symbole
- Die Beziehung (Zuordnung bzw. Mapping) zwischen Gates, Kanälen und Symbolen
- Die Kardinalität jeder Beziehung

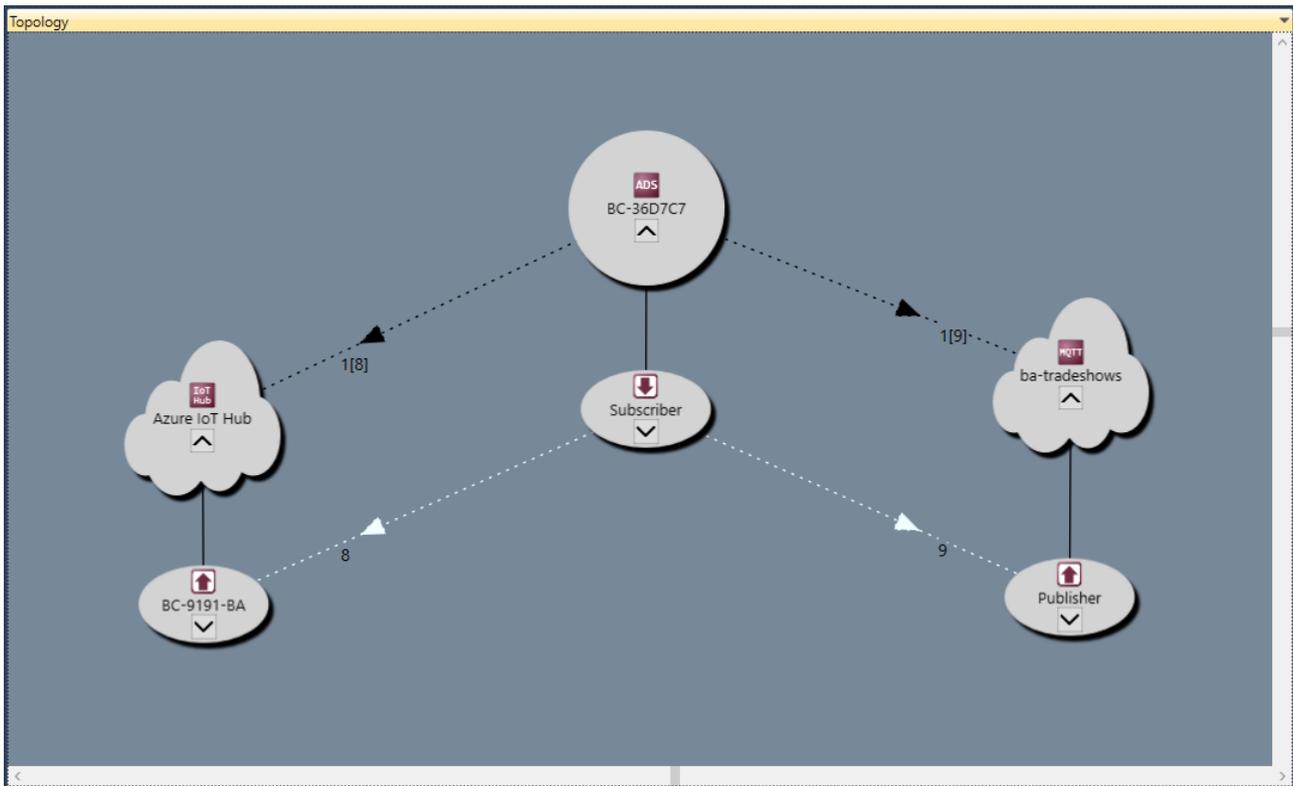
Die Topologieansicht kann zum Erstellen und Bearbeiten einer Konfiguration verwendet werden. Rechtsklicken Sie dazu einfach auf die leere Fläche, um das Kontextmenü zu öffnen, und wählen Sie eine der verschiedenen Konfigurationsoptionen aus, z. B. um ein neues Gate zu erstellen, einem Gate einen neuen Kanal anzufügen oder eine Komponente aus der Konfiguration zu entfernen.

Sie können jedes Objekt in der Topologieansicht verschieben, indem Sie es an eine neue Position ziehen. Jede angefügte Teilkomponente wird zusammen mit ihrer übergeordneten Komponente verschoben. Optional können Sie eine Teilkomponente auch ausblenden, indem Sie auf die Schaltfläche Erweitern ihrer übergeordneten Komponente klicken. Beim Speichern einer Konfiguration wird die Position jedes Objekts in der Konfigurationsdatei gespeichert.

Um die Navigation ein wenig einfacher zu gestalten, unterstützt die Topologieansicht die folgenden Funktionen:

- Bildlaufleisten für vertikale und horizontale Navigation
- Vertikale Navigation über Mausrad
- Horizontale Navigation über Mausrad und SHIFT-Taste (gedrückt halten)
- Heranzoomen/Herauszoomen über Mausrad und STRG-Taste (gedrückt halten)

Anstelle der Topologieansicht kann auch die Baumansicht für die Konfiguration verwendet werden, aber die Topologieansicht bietet einen besseren grafischen Überblick über die aktuelle Konfiguration (siehe [Baumansicht \[► 44\]](#))

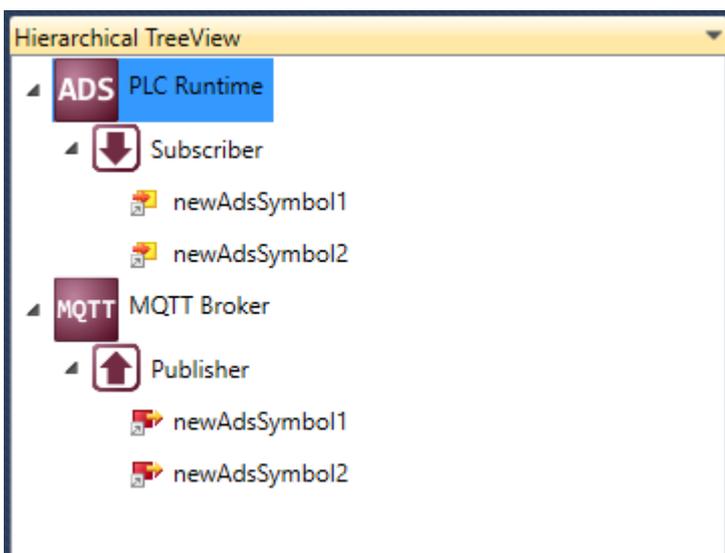


## 5.2.2 Baumansicht

Die Baumansicht ist eine hierarchische Ansicht der derzeit geöffneten Konfiguration. Sie zeigt die folgenden Komponenten einer Konfiguration:

- Die konfigurierten Gates, Kanäle und Symbole
- Das Vorhandensein einer Beziehung (Zuordnung bzw. Mapping) zwischen Symbolen

Die Baumansicht kann auch zur Bearbeitung der Konfiguration verwendet werden, aber es ist womöglich einfacher, stattdessen die Topologieansicht zu verwenden (siehe [Topologieansicht \[► 43\]](#)).



### 5.2.3 Zuordnungen (Mappings)

Das Fenster mit den Zuordnungen gibt einen Überblick über alle Verbindungen zwischen Symbolen in der derzeit geöffneten Konfiguration. Wenn eine Verbindung ausgewählt wird, wird sie in der Topologieansicht automatisch hervorgehoben.

Icon	Source Gate	Source Channel	Source Symbol	Source DataType	Icon	Destination Gate	Destination Channel	Destination Symbol	Destination DataType	Filter Symbols	Valid	Visible
ADS	BC-36D7C7	Subscriber	TemperatureActual	REAL	ba-tradeshows	Publisher	TemperatureActual	TemperatureActual	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	TemperatureActual	REAL	ba-tradeshows	Publisher	TemperatureActual	TemperatureActual	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	TemperatureSetPointShift_AI00	REAL	ba-tradeshows	Publisher	TemperatureSetPointShift_AI00	TemperatureSetPointShift_AI00	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	TemperatureSetPoint_AI01	REAL	ba-tradeshows	Publisher	TemperatureSetPoint_AI01	TemperatureSetPoint_AI01	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	CoolingControlValue_AI02	REAL	ba-tradeshows	Publisher	CoolingControlValue_AI02	CoolingControlValue_AI02	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	HeatingControlValue_AI03	REAL	ba-tradeshows	Publisher	HeatingControlValue_AI03	HeatingControlValue_AI03	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	KeyCardDetected	UINT	ba-tradeshows	Publisher	KeyCardDetected	KeyCardDetected	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	BalconyDoorOpened	UINT	ba-tradeshows	Publisher	BalconyDoorOpened	BalconyDoorOpened	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	PresentDetected	UINT	ba-tradeshows	Publisher	PresentDetected	PresentDetected	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	TemperatureActual	REAL	Azure IoT Hub	BC-9191-BA	TemperatureActual	TemperatureActual	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	TemperatureSetPointShift_AI00	REAL	Azure IoT Hub	BC-9191-BA	TemperatureSetPointShift_AI00	TemperatureSetPointShift_AI00	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	TemperatureSetPoint_AI01	REAL	Azure IoT Hub	BC-9191-BA	TemperatureSetPoint_AI01	TemperatureSetPoint_AI01	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	CoolingControlValue_AI02	REAL	Azure IoT Hub	BC-9191-BA	CoolingControlValue_AI02	CoolingControlValue_AI02	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	HeatingControlValue_AI03	REAL	Azure IoT Hub	BC-9191-BA	HeatingControlValue_AI03	HeatingControlValue_AI03	REAL	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	KeyCardDetected	UINT	Azure IoT Hub	BC-9191-BA	KeyCardDetected	KeyCardDetected	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	BalconyDoorOpened	UINT	Azure IoT Hub	BC-9191-BA	BalconyDoorOpened	BalconyDoorOpened	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ADS	BC-36D7C7	Subscriber	PresentDetected	UINT	Azure IoT Hub	BC-9191-BA	PresentDetected	PresentDetected	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Filtered View!

Logging Cascading Editor Mapping Links Target Browser Application Config

### 5.2.4 Target Browser

Der Target Browser dient für den symbolischen Zugriff für ADS- und OPC UA-Ziellaufzeiten. Er kann verwendet werden, um Symbole per Drag-and-drop für eine Ziellaufzeit zu konfigurieren.

Name	Type	Size	Category	Comment	Subitems	Unit	Context-Mask	Index-Group	Index-Offset	Attributes (Instance)	Attributes (Type)
GVL_static_...	0	Struct			84	0	0	0	0	none	
MAIN	0	Struct			9	0	0	0	0	none	
complex	ST_96	Struct			4	0	4040	7D53C		none	none
fbTest1	FB_11	Struct			9	0	4040	7D454		<OPC.UA.DA: 1>	none
fbTest2	FB_11	Struct			9	0	4040	7D4C8		none	none
i	INT_2	Primitiv			0	0	4040	7D452		none	none

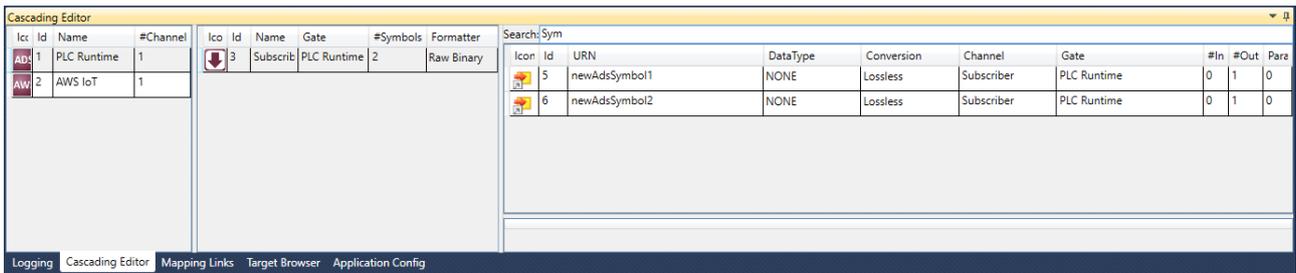
Logging Cascading Editor Mapping Links Target Browser Application Config

Abb. 1:

Name	Type	Size	Category	Full-Name	Comment	Subitems	NodeClass	Identifier	NamespaceIndex
Views	0	Struct	Views			0	Object	87	0
Objects	0	Struct	Objects			5	Object	85	0
Server	0	Struct	Objects.Server			14	Object	2253	0
Configuration	0	Struct	Objects.Configuration			5	Object	16	7
PLC1	0	Struct	Objects.PLC1			12	Object	PLC1	1
AlarmsConditions	0	Struct	Objects.AlarmsConditions			0	Object	AlarmsConditions	6
DeviceSet	0	Struct	Objects.DeviceSet			1	Object	5001	2
Types	0	Struct	Types			5	Object	86	0

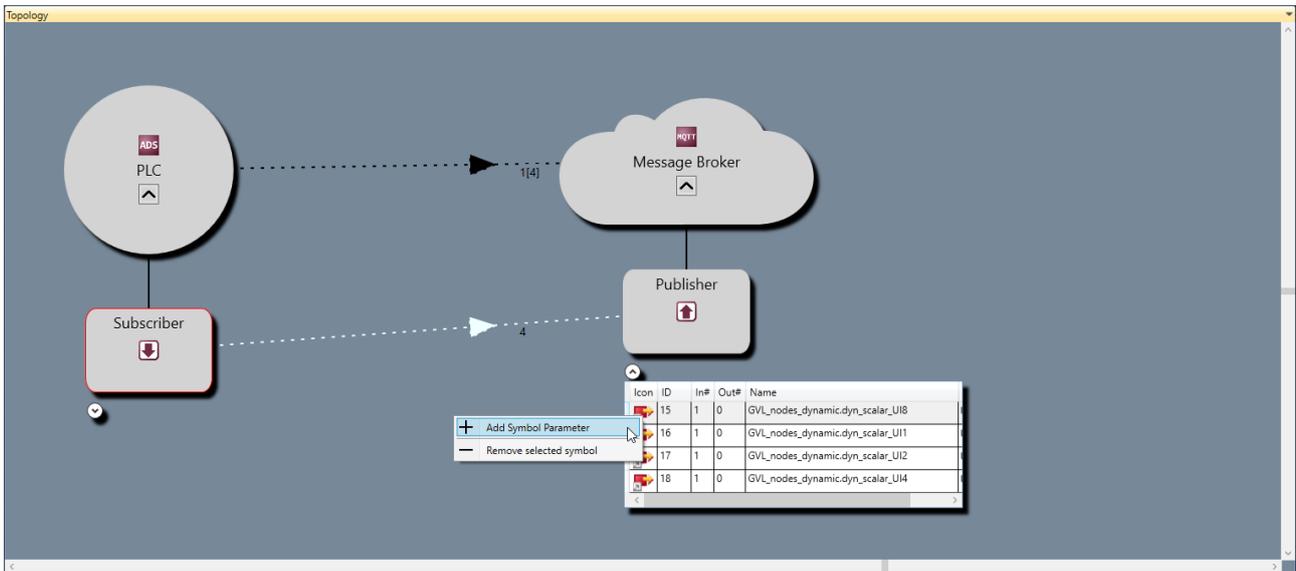
### 5.2.5 Cascading Editor

Der Cascading Editor hilft beim Navigieren durch große und komplexe Konfigurationen, indem Filtermechanismen für Symbole bereitgestellt werden. Von links nach rechts können Gates und Kanäle ausgewählt werden, um die entsprechenden Symbole anzuzeigen. Außerdem kann mit freiem Text nach Symbolnamen gesucht werden. Wenn eine Komponente ausgewählt wird, wird sie in der Topologieansicht automatisch hervorgehoben.

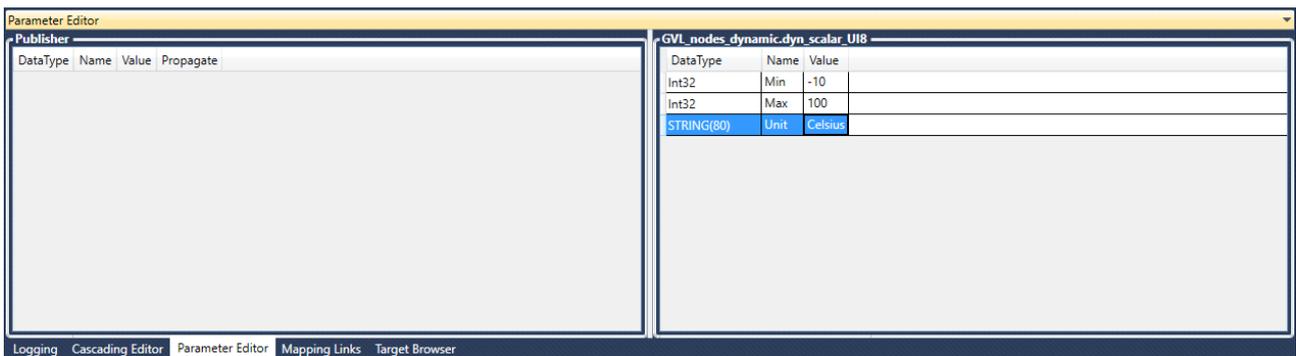


### 5.2.6 Parameter Editor

Der Parameter Editor ermöglicht die Konfiguration von Symbolmetadaten für Symbole in einem Publisher-Kanal. Als Voraussetzung muss der Publisher-Kanal für die Verwendung des Datenformats „TwinCAT JSON“ konfiguriert werden. Anschließend können durch Rechtsklick auf ein Symbol und Auswahl von **Add Symbol Parameter** neue Symbolparameter hinzugefügt werden.



Dadurch wird dem Parameter Editor ein neuer Symbolparametereintrag hinzugefügt. Danach können der Datentyp, Name und Wert des neuen Parameters festgelegt werden.



Ein Ergebnis der obigen Konfiguration kann bei Verwendung des Datenformats TwinCAT JSON wie folgt aussehen. In dieser Konfiguration wurde die Variable „GVL\_nodes\_dynamic.dyn\_scalar\_UI8“ mit den Symbolparametern Min, Max und Unit konfiguriert. Diese Parameter werden dem Abschnitt „MetaData“ der TwinCAT JSON-formatierten Nachricht hinzugefügt.

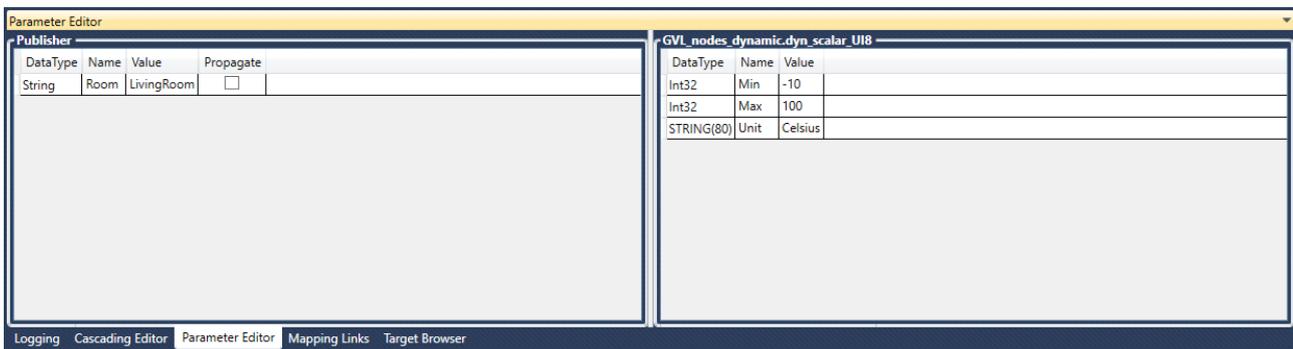
```
{
  "Timestamp": "2020-09-28T12:32:49.0370000+02:00",
  "GroupName": "Publisher",
  "Values": {
    "GVL_nodes_dynamic.dyn_scalar_UI8": 259096147,
    "GVL_nodes_dynamic.dyn_scalar_UI1": 83,
    "GVL_nodes_dynamic.dyn_scalar_UI2": 32339,
    "GVL_nodes_dynamic.dyn_scalar_UI4": 259096147
  },
}
```

```

"MetaData": {
  "GVL_nodes_dynamic.dyn_scalar_UI8": {
    "Timestamp": "2020-09-28T12:32:49.0350000+02:00",
    "Min": -10,
    "Max": 100,
    "Unit": "Celsius"
  },
  "GVL_nodes_dynamic.dyn_scalar_UI1": {
    "Timestamp": "2020-09-28T12:32:49.0350000+02:00"
  },
  "GVL_nodes_dynamic.dyn_scalar_UI2": {
    "Timestamp": "2020-09-28T12:32:49.0350000+02:00"
  },
  "GVL_nodes_dynamic.dyn_scalar_UI4": {
    "Timestamp": "2020-09-28T12:32:49.0350000+02:00"
  }
}
}

```

Darüber hinaus können Parameter auch einem Kanal hinzugefügt werden, wodurch die Root-Nachricht um Metadateneigenschaften ergänzt wird. Beispiel:



In der Folge enthält die Root-Nachricht nun eine statische Metadateneigenschaft namens „Room“, die den Wert „LivingRoom“ hat:

```

{
  "Timestamp": "2020-09-30T09:55:58.1160000+02:00",
  "GroupName": "Publisher",
  "Room": "LivingRoom",
  "Values": {
    "GVL_nodes_dynamic.dyn_scalar_UI8": 267266470,
    "GVL_nodes_dynamic.dyn_scalar_UI1": 166,
    "GVL_nodes_dynamic.dyn_scalar_UI2": 10662,
    "GVL_nodes_dynamic.dyn_scalar_UI4": 267266470
  },
  "MetaData": {
    "GVL_nodes_dynamic.dyn_scalar_UI8": {
      "Timestamp": "2020-09-30T09:55:58.1140000+02:00",
      "Min": -10,
      "Max": 100,
      "Unit": "Celsius"
    },
    "GVL_nodes_dynamic.dyn_scalar_UI1": {
      "Timestamp": "2020-09-30T09:55:58.1140000+02:00"
    },
    "GVL_nodes_dynamic.dyn_scalar_UI2": {
      "Timestamp": "2020-09-30T09:55:58.1140000+02:00"
    },
    "GVL_nodes_dynamic.dyn_scalar_UI4": {
      "Timestamp": "2020-09-30T09:55:58.1140000+02:00"
    }
  }
}

```

## 5.2.7 Settings

Dieser Abschnitt enthält ausführliche Informationen über die verschiedenen Konfigurationsparameter, die für Gates, Kanäle und Symbole eingestellt werden können.

### 5.2.7.1 Gates

Ein Gate steht für ein Kommunikationsprotokoll oder einen bestimmten Verbindungsdienst, z. B. ADS, OPC UA, MQTT, AWS IoT oder Microsoft Azure IoT Hub. Jedes Gate wird mit Parametern konfiguriert, die für den entsprechenden Gate-Typ spezifisch sind.

#### ADS

Ein Beckhoff ADS-Gerät ist entweder ein TwinCAT 2/3-Gerät oder ein Beckhoff BC-Gerät. ADS ist das übliche Kommunikationsprotokoll von Beckhoff und kann für den Zugriff auf viele Teile des TwinCAT-Systems verwendet werden. Die gängigsten Anwendungsszenarios für den TC3 IoT Data Agent beinhalten den Zugriff auf eine TwinCAT 2/3 PLC, C++, TcCOM-Module oder das I/O-Prozessabbild, der über ADS erfolgt.

Beim Konfigurieren eines ADS-Gates müssen die folgenden Einstellungen vom TC3 IoT Data Agent vorgenommen werden, um auf das zugrunde liegende ADS-Gerät zuzugreifen:

Einstellung	Beschreibung
AmsNetId	AmsNetId des Zielgeräts, z. B. 127.0.0.1.1.1 für das lokale Gerät.
AdsPort	AdsPort des Zielgeräts, z. B. 801 (TwinCAT 2 PLC) oder 851 (TwinCAT 3 PLC)
IoMode	Legt fest, wie der TC3 IoT Data Agent mit dem ADS-Gerät kommunizieren soll. Folgende Optionen können eingestellt werden: <ul style="list-style-type: none"> <li>• Direct: greift auf jedes Symbol mit einem separaten ADS-Befehl zu. Vorgeschrieben für BC Controller, erhöht jedoch den ADS-Verkehr</li> <li>• Batched: greift auf in einem ADS-Summenbefehl zusammengefasste Symbole zu, wodurch der ADS-Verkehr optimiert wird, und kann für TwinCAT 2 und 3 PLC oder C++-Laufzeiten verwendet werden</li> </ul>

#### OPC UA

OPC UA ist ein standardisiertes, industrielles Client/Server-Kommunikationsprotokoll, das von vielen Anbietern für verschiedene Anwendungsfälle genutzt wird. Der TC3 IoT Data Agent kann auf OPC UA-Servergeräte zugreifen, um Variablen (sogenannte „Nodes“) auf diesen Geräten mit IoT-Diensten zu verbinden.

Beim Konfigurieren eines OPC UA-Gates müssen die folgenden Einstellungen vom TC3 IoT Data Agent vorgenommen werden, um auf das zugrunde liegende OPC UA-Gerät zuzugreifen:

Einstellung	Beschreibung
Server URL	Die OPC UA-Server-URL, z. B. opc.tcp://localhost:4840
Security policy	Die OPC UA-Sicherheitsrichtlinie, die der TC3 IoT Data Agent während der Verbindungsherstellung mit dem OPC UA-Server verwenden sollte
Security mode	Der OPC UA-Nachrichtensicherheitsmodus, den der TC3 IoT Data Agent während der Verbindungsherstellung mit dem OPC UA-Server verwenden sollte
Authentication mode	Der OPC UA-Authentifizierungsmodus, den der TC3 IoT Data Agent während der Verbindungsherstellung mit dem OPC UA-Server verwenden sollte

#### MQTT

MQTT kann für die Verbindung mit einem allgemeinen Message-Broker verwendet werden, z. B. Mosquitto, HiveMQ oder ähnliche Broker-Typen.

Einstellung	Beschreibung
Broker address	Die IP-Adresse oder der Hostname des MQTT-Message-Brokers
Port	MQTT spezifiziert Port 1883 für unverschlüsselte Kommunikation und 8883 für verschlüsselte Kommunikation
ClientId	Ein numerischer oder stringbasierter Wert, der den Client identifiziert. Je nach Message-Broker-Typ sollte diese ID eindeutig sein
Authentication mode	Legt fest, ob sich der TC3 IoT Data Agent beim Broker durch eine Benutzername/Passwort-Kombination authentifizieren soll
TLS mode	Legt fest, ob TLS zur Sicherung des Kommunikationskanals zum Message-Broker verwendet werden soll. Zu beachten ist, dass auch der Message-Broker TLS verwenden muss, damit dies funktioniert. Für TLS stehen verschiedene Optionen zur Auswahl: <ul style="list-style-type: none"> <li>• CA certificate: Verwendet nur das CA-Zertifikat für die Serverauthentifizierung</li> <li>• Client certificate: Verwendet ein Client-Zertifikat für die gegenseitige Client/Server-Authentifizierung</li> <li>• PSK: Verwendet eine gemeinsame PSK-Identität und einen gemeinsamen PSK-Schlüssel, die dem Message-Broker und dem Client bekannt sind</li> </ul>

**Microsoft Azure IoT Hub**

Mit Azure IoT Hub bietet die Cloud-Plattform Microsoft Azure einen Verbindungsdienst in der Cloud an, der bidirektionale Kommunikation, Gerätesicherheit und automatische Skalierbarkeit bietet. Im TC3 IoT Data Agent kann der IoT Hub als spezieller Gate-Typ konfiguriert werden.

Einstellung	Beschreibung
HostName	URL der Azure IoT Hub-Instanz
Deviceld	Deviceld des Geräts, das auf der IoT Hub-Konfigurationswebsite erstellt wurde
SharedAccessKey	Entweder der primäre oder der sekundäre Geräteschlüssel, der zusammen mit dem Gerät auf der IoT Hub-Konfigurationswebsite generiert wurde
CA file	Die CA-Datei, die für die Serverauthentifizierung verwendet wird. Zum Zeitpunkt der Verfassung dieses Artikels spielen bei der Serverauthentifizierung drei Zertifikate eine Rolle, die zur Zertifikatkette gehören und miteinander verbunden sind. <ul style="list-style-type: none"> <li>• Root CA (Stammzertifikat): Baltimore CyberTrust Root</li> <li>• Intermediate CA (Zwischenzertifikat): Baltimore CyberTrust</li> <li>• Wildcard-Zertifikat</li> </ul> <p>Während des anfänglichen TLS-Handshakes werden nur die ersten beiden vom Server an den Client gesendet. Der Client validiert normalerweise nur das Root CA der Kette und bestimmt, ob es vertrauenswürdig ist.</p> <p>Um die CA-Datei für das Root CA zu erwerben, können Sie den Windows-Zertifikatspeicher (certmgr.msc) öffnen, zu den vertrauenswürdigen Stammzertifizierungsstellen browsen und das Zertifikat Baltimore CyberTrust Root exportieren.</p>

**AWS IoT**

Mit AWS IoT bietet die Cloud-Plattform Amazon Web Services einen Message-Broker-Dienst in der Cloud an, der bidirektionale Kommunikation, Gerätesicherheit und automatische Skalierbarkeit bietet. Im TC3 IoT Data Agent wird AWS IoT als regulärer MQTT-Gate konfiguriert. Für eine erfolgreiche Verbindung mit einer AWS IoT-Instanz müssen jedoch einige spezielle MQTT-Einstellungen konfiguriert werden.

Einstellung	Beschreibung
Broker address	Die URL der AWS IoT-Instanz
Port	Port 8883 für verschlüsselte TLS-Kommunikation ist vorgeschrieben
ClientId	Kann beliebig festgelegt werden, muss jedoch eindeutig sein. In der Regel könnte dies der AWS IoT-Objektname sein.
Authentication mode	Eingestellt auf „No authentication“. Die Authentifizierung bei AWS IoT erfolgt über das TLS-Client-Zertifikat.
TLS mode	Wird vom Konfigurator automatisch auf „Client certificate“ eingestellt. Wählen Sie den Pfad zur CA-Datei, Client-Zertifikatsdatei und Client-Schlüsseldatei aus. Dies sind die Dateien, die auf der Konfigurationswebsite von AWS IoT erzeugt und heruntergeladen werden können.

### 5.2.7.2 Kanäle

Kanäle werden auf einem Gate zum Senden („Publisher“) oder Empfangen („Subscriber“) von Daten an ein/ von einem Gate konfiguriert.

- Source: Dieses Gate ist die Quelle der Daten, d. h. der TC3 IoT Data Agent verbindet sich mit dem Gate und fragt Daten von ihm ab, um diese Daten woandershin (an ein „Destination Gate“) zu senden. Technisch wird dies auch als „Subscriber-Kanal“ bezeichnet.
- Destination: Dieses Gate ist das Ziel der Daten, d. h. der TC3 IoT Data Agent verbindet sich mit dem Gate und sendet ihm Daten, die er von einem anderen Gate (von einem „Source Gate“) erhalten hat. Technisch wird dies auch als „Publisher-Kanal“ bezeichnet.

Jeder Kanal hat andere Einstellungen, die entweder das Datenformat beschreiben, das für diesen Kanal verwendet werden soll, oder die Abtasteinstellungen, die der TC3 IoT Data Agent zum Sammeln der Daten verwenden soll. Diese Einstellungen können auch vom Gate-Typ abhängen, für den der Kanal konfiguriert wurde.

In der folgenden Tabelle sind alle vorhandenen Einstellungen aufgeführt.

Einstellung	Beschreibung	Anwendbar auf Gate-Typ
Direction	Legt fest, ob der Kanal entweder ein Publisher- (Sender-) oder Subscriber- (Empfänger-)Kanal sein soll. Je nach Auswahl und Gate-Typ sind weitere Einstellungen erforderlich oder werden vorausgewählt.	Alle Gates
Einstellung	Beschreibung	Anwendbar auf Gate-Typ
SamplingMode	Wählt aus, ob der Kanal beim Sammeln der Daten von einer Quelle entweder zyklische oder ereignisbasierte Abtastmechanismen verwenden soll. Zu beachten ist, dass je nach Richtung nicht alle Gates beide Arten unterstützen. Ein MQTT-Gate beispielsweise verwendet beim Empfangen von Daten immer den SamplingMode „event“ (wegen des Pub/Sub-Prinzips ist dies immer eventbasiert).	Alle Gates
CycleTime	Nur beim SamplingMode „cyclic“ anwendbar. Legt die Abtastrate in [ms] fest.	Alle Gates
Timeout	Das Timeout für eine Kommunikation mit dem Gate in [ms].	Alle Gates
PartialUpdate	Aktiviert/deaktiviert partielle Updates auf diesem Kanal. Bei Aktivierung (Standard) enthält ein Publish nur das aktualisierte Symbol. Bei Deaktivierung enthält ein Publish alle Symbole eines Kanals mit ihrem zuletzt bekannten Wert. Nur auf Publisher-Kanäle anwendbar.	Alle Gates
BufferSize	Legt die Größe (Anzahl der Nachrichten) des Ringpuffers bei Verbindungsverlust fest.	MQTT, AWS IoT, Azure IoT Hub
Einstellung	Beschreibung	Anwendbar auf Gate-Typ
Formatter	Legt das Datenformat fest, das für diesen Kanal verwendet werden soll, z. B. binär oder JSON. Zu beachten ist, dass einige Gates ihren Kanälen vorschreiben, ein vorgegebenes Datenformat zu verwenden, z. B. ADS- oder OPC UA-Gates, da die Kommunikation mit diesen Geräten ein spezifisches Format erfordert. In diesem Fall ist der Formatter vorgegeben und kann nicht über den Konfigurator geändert werden.	MQTT, AWS IoT, Azure IoT Hub
FormatterType	Legt den Formatter-Typ auf diesem Kanal fest. In den meisten Fällen ist der Formatter-Typ ein InOut-Typ. Für weitere Informationen über diese Einstellung sehen Sie sich unseren Dokumentationsartikel über das Schreiben benutzerdefinierter Plugins über die Formatter-Schnittstelle an.	MQTT, AWS IoT, Azure IoT Hub
Einstellung	Beschreibung	Anwendbar auf Gate-Typ
Topic	Legt das MQTT-Topic fest, das für Publishing oder Subscribing verwendet werden soll.	MQTT
QoS	Legt das QoS (Quality-of-Service)-Niveau fest, das bei Publishing oder Subscribing verwendet werden soll.	MQTT, AWS IoT
Retain	Legt fest, ob eine Nachricht als „Retain“ gesendet werden soll. (Nur für den Publisher-Kanal relevant)	MQTT

Einstellung	Beschreibung	Anwendbar auf Gate-Typ
SendStateInfo	<p>Bei Aktivierung veröffentlicht der TC3 IoT Data Agent seinen „OnlineState“ an das Subtopic /Desc/ und verwendet dieses Subtopic in seinem LastWill. Wenn sich der TC3 IoT Data Agent mit dem Message-Broker verbindet, wird eine JSON-Nachricht an dieses Topic veröffentlicht, die Folgendes enthält</p> <pre>{ "OnlineState" : true }</pre> <p>Wenn der TC3 IoT Data Agent die Verbindung zum Message-Broker ordnungsgemäß trennt, wird die folgende Nachricht an dieses Topic gesendet:</p> <pre>{ "OnlineState" : false }</pre> <p>Wenn der Message-Broker erkennt, dass der TC3 IoT Data Agent die Verbindung verloren hat, wird die folgende Nachricht an dieses Topic gesendet (LastWill):</p> <pre>{ "OnlineState" : false }</pre>	MQTT, AWS IoT

## Abtastmodi

Der TC3 IoT Data Agent beinhaltet verschiedene Abtastmodi, die beeinflussen, wie Daten von einer Quelle erfasst oder an ein Ziel geschrieben werden. Der Abtastmodus kann auf einem Kanal eingestellt werden. Gegenwärtig sind die folgenden Abtastmodi vorhanden:

- Cyclic
- OnChange
- TriggerSymbol

### Cyclic

Zyklische Abtastung bedeutet, dass der TC3 IoT Data Agent zyklisch das Gate auf Daten abtastet (Subscriber-Kanal) oder zyklisch Daten an ein Gate schreibt (Publisher-Kanal). Auf einem Subscriber-Kanal führt dies zu zyklischen Lesebefehlen, während es auf einem Publisher-Kanal zu zyklischen Schreibebefehlen führt, z. B. auf einem ADS- oder OPC UA-Gate. Auf Gate-Typen, die auf Publisher/Subscriber-Konzepten basieren, z. B. MQTT-, AWS IoT- und Azure IoT Hub-Gates, werden zyklische Anfragen auf einem Subscriber-Kanal automatisch durch Abonnements (Subscriptions) ersetzt, während dies auf einem Publisher-Kanal zu zyklischen Publish-Befehlen führt.

### OnChange

OnChange-Abtastung bedeutet, dass der TC3 IoT Data Agent nur Daten mit einem Gate austauscht, wenn sich der Wert einer Variablen geändert hat.

### Trigger-Symbole

Trigger-Symbole ermöglichen eine Abtastung „auf Anforderung“, z. B. wenn eine bestimmte Bedingung für ein bestimmtes Symbol (das sogenannte „Trigger-Symbol“) erfüllt ist. Es können verschiedene Arten von Bedingungen festgelegt werden. Sie werden als Teil eines Kanals konfiguriert und ermöglichen die Festlegung der folgenden Bedingungsarten.

Bedingungsart	Beschreibung
EQ	Wert des Trigger-Symbols ist gleich einem bestimmten Wert
NE	Wert des Trigger-Symbols weicht von einem bestimmten Wert ab
LE	Wert des Trigger-Symbols ist kleiner gleich einem bestimmten Wert
GE	Wert des Trigger-Symbols ist größer gleich einem bestimmten Wert
LT	Wert des Trigger-Symbols ist kleiner als ein bestimmter Wert
GT	Wert des Trigger-Symbols ist größer als ein bestimmter Wert

Wenn die Bedingung erfüllt ist, werden alle Symbole in diesem Kanal an das entsprechende Gate veröffentlicht. Außerdem kann festgelegt werden, wie oft die Symbolwerte gesendet werden sollen.

Sendeverhalten	Beschreibung
risingEdge	Die Symbole werden nur einmal gesendet, wenn die Bedingung erfüllt ist
continuous	Die Symbole werden gesendet, solange die Bedingung erfüllt ist

### ● Verwendung von Trigger-Symbolen

## I

Trigger-Symbole können nur für ADS- und OPC UA-Subscriber-Symbole konfiguriert werden.

- Fügen Sie das Symbol, das als Trigger-Symbol fungieren soll, dem ADS- oder OPC UA-Subscriber-Kanal hinzu.
- Konfigurieren Sie den verbundenen Publisher-Kanal mit dem SamplingMode „OnTrigger“, um das zuvor hinzugefügte Symbol als Trigger-Symbol festzulegen.

### 5.2.7.3 Symbole

Symbole stehen für Variablen von einem Gate, z. B. eine TwinCAT SPS-Variable. Die Symbolkonfiguration enthält die Adressinformationen, die der Data Agent benötigt, um den Wert eines Symbols zu lesen oder zu schreiben. Diese Adressinformationen hängen daher vom Gate-Typ ab.

Auf Gates, die einen [Target Browser \[► 45\]](#) unterstützen, erkennt der Browser automatisch die richtigen Adressinformationen für ein Symbol. Bei allen anderen Gates müssen diese Informationen manuell eingegeben werden.

Gate-Typ	Einstellung	Beschreibung
ADS	URN	Symbolname-Adressinformation der ADS-Variable. Funktioniert nicht bei allen ADS-Geräten, z. B. unterstützen BC-Geräte keine Symbolnamen.
ADS	IndexGroup IndexOffset	Die IndexGroup/IndexOffset-Kombination kann verwendet werden, um auf Daten eines ADS-Geräts zuzugreifen, das Symbol-Adressinformationen nicht unterstützt. Im Fall von TwinCAT PLC steht die IndexGroup/IndexOffset-Kombination direkt für eine Speicheradresse, z. B. von einer SPS-Variablen, die sich nach einer Neukompilierung des TwinCAT-Projekts ändern kann. Daher ist es gängige Praxis, stattdessen den TwinCAT PLC-Symbolserver zu verwenden, der Symbolinformationen für seine SPS-Variablen bereitstellt, was bedeutet, dass auf eine Variable über ihren Symbolnamen zugegriffen werden kann, der auch nach einer Neukompilierung oder Online-Änderung gültig bleibt (falls das Symbol noch vorhanden ist).  Einige ADS-Dienste umfassen jedoch keinen solchen Symbolserver, z. B. kleine Beckhoff BC-Geräte. In diesen Fällen muss die IndexGroup/IndexOffset-Kombination verwendet werden.
ADS	DataType	Datentyp des Symbols
ADS	Conversion	Legt den Konvertierungsmodus für dieses Symbol fest.
OPC UA	Name	Beschreibender Name des OPC UA-Nodes. Wird nur im Konfigurator verwendet, stellt keine Online-Adressinformation dar.
OPC UA	Identifizier	Identifizierung des OPC UA-Symbols auf dem Server, z. B.: <ul style="list-style-type: none"> <li>• s = MAIN.nCounter (wenn der IdentifizierType „String“ ist)</li> <li>• n = 42 (wenn der IdentifizierType „Numeric“ ist)</li> </ul>
OPC UA	NsName	Name des Namensraums, in dem sich das Symbol befindet. Dieser entspricht dem Namensraum-Index, der Teil einer OPC UA Nodeld ist. Die Übersetzung kann über das NamespaceArray erfolgen.
OPC UA	Attributeld	Die Attributeld definiert das OPC UA-Attribut, das vom Data Agent beim Lesen eines Symbols verwendet werden soll. In den meisten Szenarios ist dies der „Wert“ eines Symbols.
OPC UA	Conversion	Legt den Konvertierungsmodus für dieses Symbol fest.
MQTT	URN	Name des MQTT-Symbols. Dieser entspricht dem Namen, der im JSON-Format als Schlüssel verwendet wird, auch beim Empfang von Daten vom MQTT-Gate.
MQTT	DataType	Datentyp des Symbols
MQTT	Conversion	Legt den Konvertierungsmodus für dieses Symbol fest.
IoT Hub	URN	Name des IoT Hub-Symbols. Dieser entspricht dem Namen, der im JSON-Format als Schlüssel verwendet wird, auch beim Empfang von Daten vom IoT Hub.
IoT Hub	DataType	Datentyp des Symbols
IoT Hub	Conversion	Legt den Konvertierungsmodus für dieses Symbol fest.

Beim Konfigurieren eines Kanals können Symbole über einen Target Browser oder manuell hinzugefügt werden, indem die richtigen Adressinformationen angegeben werden. Zu beachten ist, dass nicht alle Gate-Typen Target Browser-Funktionen umfassen. In diesem Fall müssen Symbole manuell konfiguriert werden.

### Manuelle Symbolkonfiguration

Wenn das Zielgerät nicht online ist oder keine Symbol-Adressinformationen bereitstellt (z. B. das BC9191), können Symbole durch Eingabe der Symboladresse auch manuell hinzugefügt werden. Die Tabellen am Anfang dieses Dokuments zeigen, welche Informationen in diesem Fall benötigt werden.

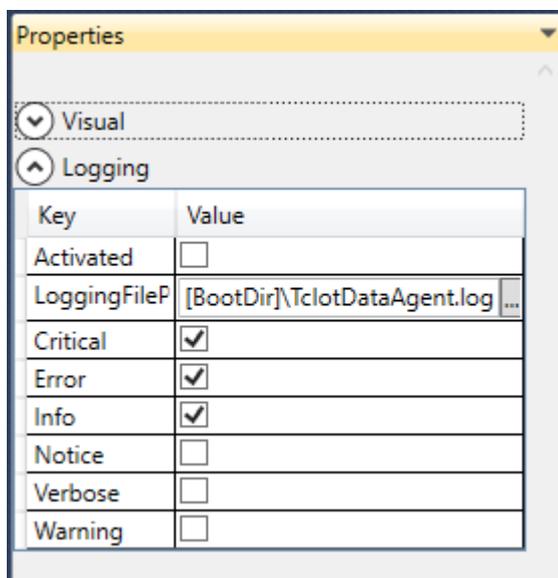
### Typkonvertierung

Der TC3 IoT Data Agent unterstützt die Konvertierung des Datentyps, bevor die Daten an ein Gate veröffentlicht werden. Die Typkonvertierung erfolgt auf Symbolebene, was bedeutet, dass verschiedene Symbole verschiedene Konvertierungsmodi verwenden können. Die folgende Tabelle zeigt die verschiedenen vorhandenen Konvertierungsmodi.

Konvertierungsmodus	Beschreibung
Losless	Standardeinstellung. „Kleinere“ Typen können in „größere“ Typen konvertiert werden. Beispielsweise kann ein Subscriber-Symbol vom Datentyp INT als ein Symbol vom Datentyp Int32 veröffentlicht werden (2 Byte in 4 Byte).
Lossy	Falls erforderlich, können auch „größere“ Symbole in „kleinere“ Symbole konvertiert werden. Beispielsweise kann ein Subscriber-Symbol vom Datentyp DINT als ein Symbol vom Datentyp Int16 veröffentlicht werden (4 Byte in 2 Byte). Je nach Wert des Subscriber-Symbols kann dies natürlich zu abgeschnittenen Werten führen.
Strict	Falls erforderlich, können Symbole auch so konfiguriert werden, dass sie den „strikten Modus“ verwenden. In diesem Konvertierungsmodus muss die Datentypgröße eines Subscriber-Symbols exakt mit dem Datentyp des zugeordneten Publisher-Symbols übereinstimmen. Beispielsweise kann ein Subscriber-Symbol vom Datentyp INT nur als ein Symbol vom Datentyp Int16 veröffentlicht werden (2 Byte in 2 Byte).

### 5.2.8 Fehlerprotokollierung

Zur Fehlersuche kann der TC3 IoT Data Agent eine Protokolldatei erzeugen, die auf der Grundlage verschiedener Protokollierungsebenen gefüllt werden kann. Alle erforderlichen Einstellungen können über das Eigenschaften-Fenster des Konfigurators konfiguriert werden. Klicken Sie dazu einfach auf eine leere Stelle in der Topologieansicht und konfigurieren Sie die Protokollierungseinstellungen im Eigenschaften-Fenster.



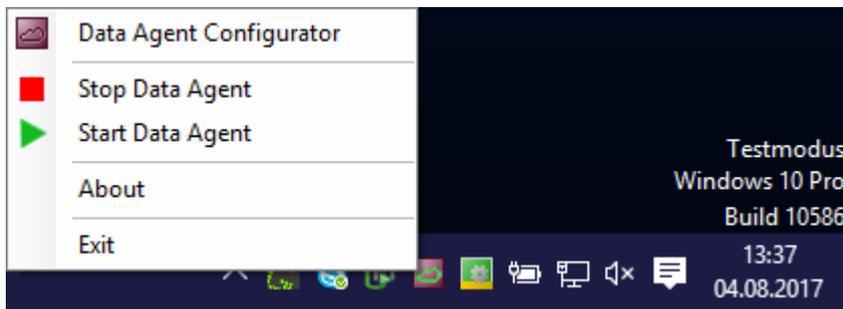
Zu beachten ist, dass alle Einstellungen mit der derzeit geöffneten Konfiguration verbunden sind und für jede Konfiguration individuell festgelegt werden müssen. Das Standardverzeichnis, in dem die Protokolldatei erstellt wird, ist das TwinCAT-Bootverzeichnis. Der Platzhalter [BootDir] wählt das TwinCAT-Bootverzeichnis automatisch aus.

#### ● Logging-Fenster

**i** Das Logging-Fenster im Konfigurator zeigt nur protokollierte Ereignisse an, die mit dem Konfigurator in Zusammenhang stehen. Um ein Protokoll für den TC3 IoT Data Agent-Hintergrunddienst zu erstellen, sind die vorstehenden Einstellungen erforderlich.

## 5.3 Taskleiste

Die lokalen Funktionen des TC3 IoT Data Agent sind auch über die Windows-Taskleiste verfügbar, z. B. das Starten oder Stoppen des lokal ausgeführten TC3 IoT Data Agent. Das Setup von TF6720 installiert automatisch eine Taskleistenanwendung, die diese Funktionen über das Kontextmenü bereitstellt.



Darüber hinaus zeigt das Symbol in der Taskleiste den aktuellen Status des TC3 IoT Data Agent an.

Modus	Symbol	Beschreibung
Ruhe-Modus		Derzeit ist keine Konfiguration geladen und aktiv
Run-Modus		Derzeit ist eine Konfiguration geladen und aktiv
Stopp-Modus		Eine Konfiguration geladen, aber derzeit nicht aktiv, z. B. aufgrund eines Fehlers

## 5.4 Support Information Report

Der Support Information Report ist ein Tool zum Sammeln von Produktinformationen, um mit dem technischen Beckhoff Support in Kontakt zu treten. Durch das Sammeln produktrelevanter Daten wie TwinCAT Version/Build, Produktversion, Image-Version und Gerätetyp wird der Email-Verkehr deutlich reduziert und eine schnellere Beratung ermöglicht.

### PlugIn-Mechanismus

Verschiedene Beckhoff Produkte integrieren sich in den Support Information Report über einen PlugIn-Mechanismus. Sie finden daher in den spezifischen Produkten wie z. B. dem TwinCAT Database Server im entsprechenden Produktmenü einen Eintrag zum Support Information Report.

### Support Information Report erstellen und abschicken

- ✓ Ein Support Information Report ist geöffnet.
- 1. Beschreiben Sie das aufgetretene Verhalten im Textfeld **Behavior** möglichst ausführlich.
- 2. Fügen Sie dem Report im Bereich **Attachment** optional über die Schaltfläche **Add Attachment** zusätzliche Dateien (Screenshots, ...) hinzu. Optional können Sie auch Dateien über den Fernzugriff auswählen. Wählen Sie dazu ein Target aus der Dropdown-Liste **Remote System** aus. Je nach ausgewähltem Target ist hier sogar ein Browsen von Windows-CE-Geräten möglich.
- 3. Geben Sie Ihre Kontaktdaten ein und wählen Sie eine Beckhoff Länder-Niederlassung aus. Diese Angabe ist erforderlich, um den Support Information Report abzuschicken.
- 4. Optional können Sie Ihre Kontaktdaten für die erneute Verwendung des Support Information Reports speichern. Aktivieren Sie dazu das Auswahlkästchen **Store personal data**.
- 5. Im unteren Bereich des Support Information Reports finden Sie die produktspezifischen PlugIns. Aktivieren Sie das Auswahlkästchen **Include in report**. Die für das Produkt notwendigen Informationen werden, soweit vorhanden, automatisch hinzugefügt. Im Screenshot ist beispielhaft die aktuelle Konfiguration von einem TwinCAT Database Server als XML-Datei dargestellt.
- 6. Schicken Sie den Support Information Report ab:

- Wenn das Gerät eine E-Mail-Anbindung hat, können Sie den Support Information Report direkt über die Schaltfläche **Send Report** an die Beckhoff Länder-Niederlassung schicken.
- Wenn das Gerät keine E-Mail -Anbindung hat, können Sie den Support Information Report über die Schaltfläche **Save .zip** zunächst lokal als .zip-Datei speichern und anschließend über FTP, USB etc. zugänglich machen und verschicken.

Send Report Save .zip

**Information**

Behaviour

Remote System: Local - 172.17.214.70.1.1

Attachment Add Attachment

**Personal Data**

Name: Max

Lastname: Mustermann

Company: Beckhoff Automation GmbH

Your Country: Germany

City: Verl

Street: Hülshorstweg 20

Phone: +49 5246 9630

e-Mail: support@beckhoff.de

Beckhoff subsidiary country: Germany

Store personal data

---

TC DbSrv

Include in report:

**Attachments:**

C:\TwinCAT\3.1\Boot\CurrentConfigDataBase.xml

## 6 Beispiele

Dieser Abschnitt enthält Konfigurationsbeispiele, die zeigen, wie der TwinCAT IoT Data Agent für die Verbindung mit bestimmten Gerätetypen und Cloud-Diensten verwendet wird. Der Abschnitt ist in „Southbound“- und „Northbound“-Beispiele unterteilt.

- Southbound: Verbindungen mit zugrunde liegenden Geräten, z. B. über ADS mit TwinCAT oder OPC UA mit Geräten von Drittanbietern.
- Northbound: Verbindungen mit Cloud-Diensten

### 6.1 Schnellstart

In diesem Abschnitt wird beschrieben, wie man den TC3 IoT Data Agent schnell startet.

Das Kommunikationsszenario ist ein typischer Anwendungsfall für den Data Agent: Variablen von einem Southbound-Gerät (TwinCAT PLC-Laufzeit) werden abgetastet und an einen Northbound-Dienst (MQTT-Message-Broker) veröffentlicht. In diesem Beispiel läuft der Message-Broker auf demselben Computer wie die PLC-Laufzeit.

Für die Einrichtung dieses Szenarios sind folgende Schritte erforderlich. Alle Schritte werden auf demselben Computer durchgeführt.

- [Installieren der Software](#) [► 58]
- [Vorbereiten des SPS-Programms](#) [► 58]
- [Konfigurieren des TC3 IoT Data Agents](#) [► 59]
- [Anzeigen der veröffentlichten Nachrichten](#) [► 60]

#### Installieren der Software

1. Installieren Sie den TF6720 IoT Data Agent wie im Abschnitt [Installation](#) [► 11] beschrieben.
2. Installieren Sie den Mosquitto MQTT Message Broker mit seinen Standard-Setup-Einstellungen. Der Message-Broker kann überall installiert werden und wird über seinen Hostnamen oder seine IP-Adresse verbunden.
3. Öffnen Sie eine Windows-Konsole, navigieren Sie zum Installationsverzeichnis des Mosquitto MQTT-Message-Brokers (in der Regel *C:\Program Files (x86)\mosquitto*) und starten Sie den Broker im ausführlichen Modus, indem Sie den folgenden Befehl ausführen: `mosquitto.exe -v`

#### ● Mosquitto MQTT-Message-Broker



Der Mosquitto MQTT-Message-Broker kann von [www.mosquitto.org](http://www.mosquitto.org) heruntergeladen werden. Folgen Sie unbedingt den Installationsanweisungen des Mosquitto-Installers.

#### ● Start von Mosquitto



Der Mosquitto MQTT-Message-Broker wird standardmäßig als Windows-Dienst installiert. Sie können auch die Datei `mosquitto.exe` aus einer Befehlszeile starten und Parameter wie „-v“ für ausführliche Ausgabe festlegen. Wenn Sie den Broker aus einer Befehlszeile starten möchten, stellen Sie sicher, dass der Mosquitto Windows-Dienst nicht läuft.

#### Vorbereiten des SPS-Programms

Richten Sie ein TwinCAT 3 PLC-Projekt ein und aktivieren Sie Testlizenzen für TC1200 (PLC) und TF6720 (TC3 IoT Data Agent). In diesem Beispiel sollte das SPS-Programm die folgenden Variablen enthalten:

```
PROGRAM MAIN
VAR
    nCounter1 : INT;
    nCounter2 : LREAL;
END_VAR
```

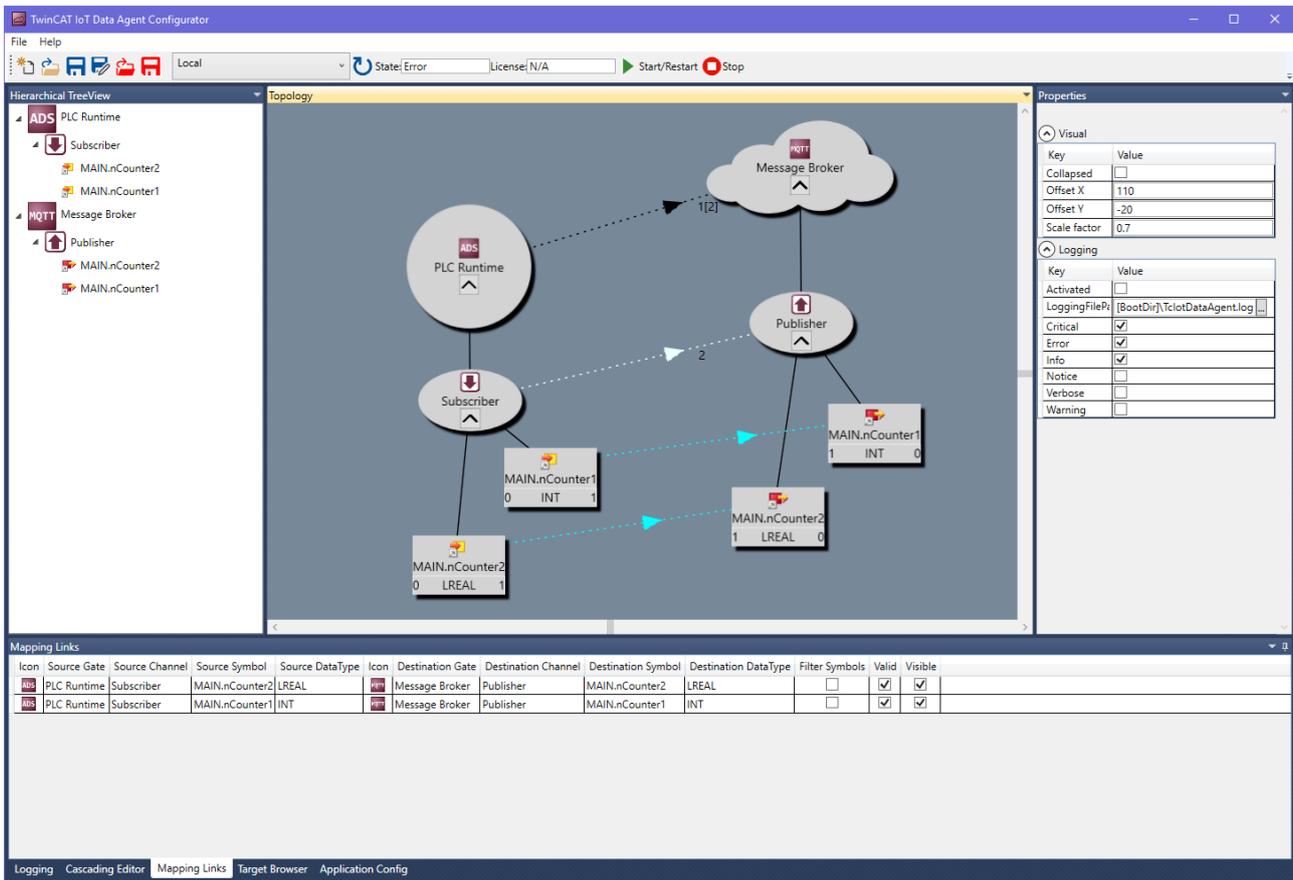
Diese zwei Variablen sollten zyklisch inkrementiert werden. Fügen Sie den folgenden Implementierungscode für das SPS-Programm hinzu:

```
nCounter1 := nCounter1 + 1;  
nCounter2 := nCounter2 + 0.1;
```

Aktivieren Sie das Projekt und starten Sie TwinCAT im Run-Modus. Stellen Sie sicher, dass das SPS-Programm läuft, indem Sie sich anmelden.

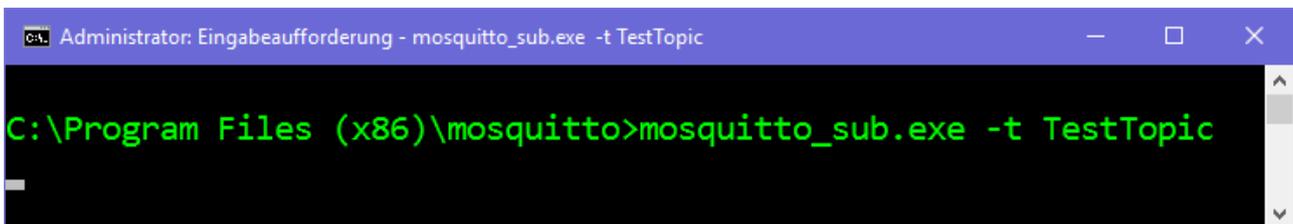
### Konfigurieren des TwinCAT IoT Data Agents

1. Starten Sie den TwinCAT IoT Data Agent-Konfigurator entweder über den Link im Windows-Startmenü oder über die Taskleistanwendung (siehe [Konfigurator \[► 41\]](#)).
2. Folgen Sie den nachstehenden Anweisungen Schritt für Schritt, um eine Data Agent-Konfiguration zu erstellen. Alternativ können Sie auch die Konfigurationsdatei „Sample\_QuickStart.xml“ öffnen, die im Installationsverzeichnis des Data Agent zu finden ist, und direkt mit Schritt (m) fortfahren.
3. Starten Sie eine neue Konfiguration, indem Sie auf das Icon **Neu** in der Konfigurator-Symbolleiste klicken.
4. Rechtsklicken Sie in der Topologieansicht auf einen leeren Bereich und wählen Sie **Add Gate (ADS)** aus.
5. Wählen Sie das neue Gate aus und stellen Sie sicher, dass die lokale AmsNetId und ADS-Port 851 im Eigenschaften-Fenster ausgewählt sind.
6. Rechtsklicken Sie in der Topologieansicht auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
7. Wählen Sie den neuen Kanal aus und stellen Sie sicher, dass der Abtastmodus „cyclic“ und die Zykluszeit „1000“ im Eigenschaften-Fenster ausgewählt sind.
8. Öffnen Sie den Target Browser und navigieren Sie zu Ihrer lokalen PLC-Laufzeit (Port 851).
9. Wählen Sie die Variablen „nCounter1“ und „nCounter2“ aus und ziehen Sie sie zum Subscriber-Kanal in der Topologieansicht.
10. Rechtsklicken Sie in der Topologieansicht auf einen leeren Bereich und wählen Sie **Add Gate (MQTT)** aus.
11. Wählen Sie das neue Gate aus und stellen Sie sicher, dass die URL auf 127.0.0.1 gesetzt ist und Port 1883 verwendet wird.
12. Rechtsklicken Sie in der Topologieansicht auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
13. Wählen Sie den neuen Kanal aus und stellen Sie sicher, dass der Abtastmodus auf „cyclic“ eingestellt ist. Stellen Sie die Zykluszeit auf „1000“ und den Formatter auf „TwinCAT JSON“ ein. Das MQTT-Topic kann auf ein Topic Ihrer Wahl eingestellt werden, z. B. „TestTopic“.
14. Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal, um eine Zuordnung zwischen den Quell- und Zielsymbolen zu erstellen. Beachten Sie, dass die Zielsymbole automatisch erzeugt werden.
15. Starten Sie eine neue Konfiguration, indem Sie auf die Schaltfläche **Activate Configuration** in der Symbolleiste klicken.

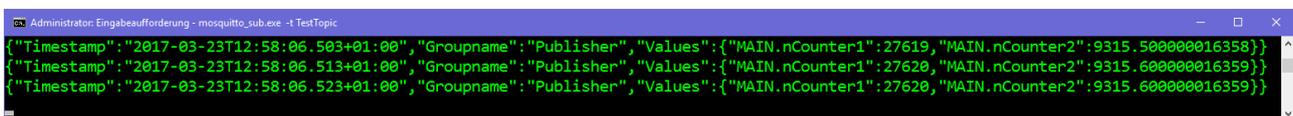


### Anzeigen der veröffentlichten Nachrichten

Öffnen Sie ein Befehlszeilenfenster und navigieren Sie zum Installationsverzeichnis des Mosquitto MQTT-Message-Brokers (in der Regel „C:\Program Files\mosquitto“ oder „C:\Program Files (x86)\mosquitto“). Starten Sie das Mosquitto Subscriber-Tool, indem Sie den folgenden Befehl ausführen: `mosquitto_sub.exe -t TestTopic`.



Das Mosquitto Subscriber-Tool erhält nun die Nachrichten, die vom TwinCAT IoT Data Agent veröffentlicht werden.



## 6.2 Southbound

### 6.2.1 Verbindung mit Geräten von Drittanbietern über OPC UA

Die meisten der schrittweisen Anweisungen in diesem Abschnitt mit Beispielen basieren auf einem OPC UA Subscriber-Kanal, aber dieselben Grundsätze können auch auf die Datenerfassung von einem anderen Gerätetyp angewendet werden.

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der TC3 IoT Data Agent konfiguriert wird, um Daten von einem OPC UA-Server abzutasten und mit einem JSON-Datenformat an einen MQTT-Message-Broker zu veröffentlichen.

### OPC UA-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein OPC UA-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals OPC UA-Nodes hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (OPC UA)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die OPC UA-spezifischen Einstellungen im Eigenschaften-Fenster (Server-URL, Sicherheitsrichtlinie, Nachrichtensicherheitsmodus, ...).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  - 5. Öffnen Sie den Target Browser.
  - 6. Navigieren Sie zum OPC UA-Namensraum und fügen Sie der Konfiguration einige Symbole hinzu, indem Sie sie zum Subscriber-Kanal ziehen.
- ⇒ Ein OPC UA-Gate und ein Subscriber-Kanal wurden erstellt. OPC UA-Nodes wurden der Konfiguration hinzugefügt.



#### ServerURL

Stellen Sie sicher, dass die ServerURL, die Sie in den Gate-Einstellungen festgelegt haben, der ServerURL entspricht, die im Target Browser verwendet wird.

### MQTT-Gate und Kanal konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein MQTT-Gate und einen Publisher-Kanal zu erstellen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (MQTT)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die MQTT-bezogenen Einstellungen im Eigenschaften-Fenster (Broker-Adresse, ...).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“, „Formatter“ auf „Simple JSON“ und „Topic“ auf „TestTopic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
- ⇒ Ein MQTT-Gate und ein Publisher-Kanal wurden erstellt.

### Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

#### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

#### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

## Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
  - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
  - ⇒ Die Konfiguration ist aktiviert.

Die resultierende Konfiguration sieht wie folgt aus:

The screenshot displays the TwinCAT IoT Data Agent Configurator interface. The main area shows a topology diagram with the following components and connections:

- OPC UA Server** (left) connected to **MQTT Broker** (right) via a dashed arrow labeled "1[2]".
- Subscriber** (left) connected to **Publisher** (right) via a dashed arrow labeled "2".
- newOpcUASymbol1** (left) connected to **newOpcUASymbol1** (right) via a dashed arrow.
- newOpcUASymbol2** (left) connected to **newOpcUASymbol2** (right) via a dashed arrow.

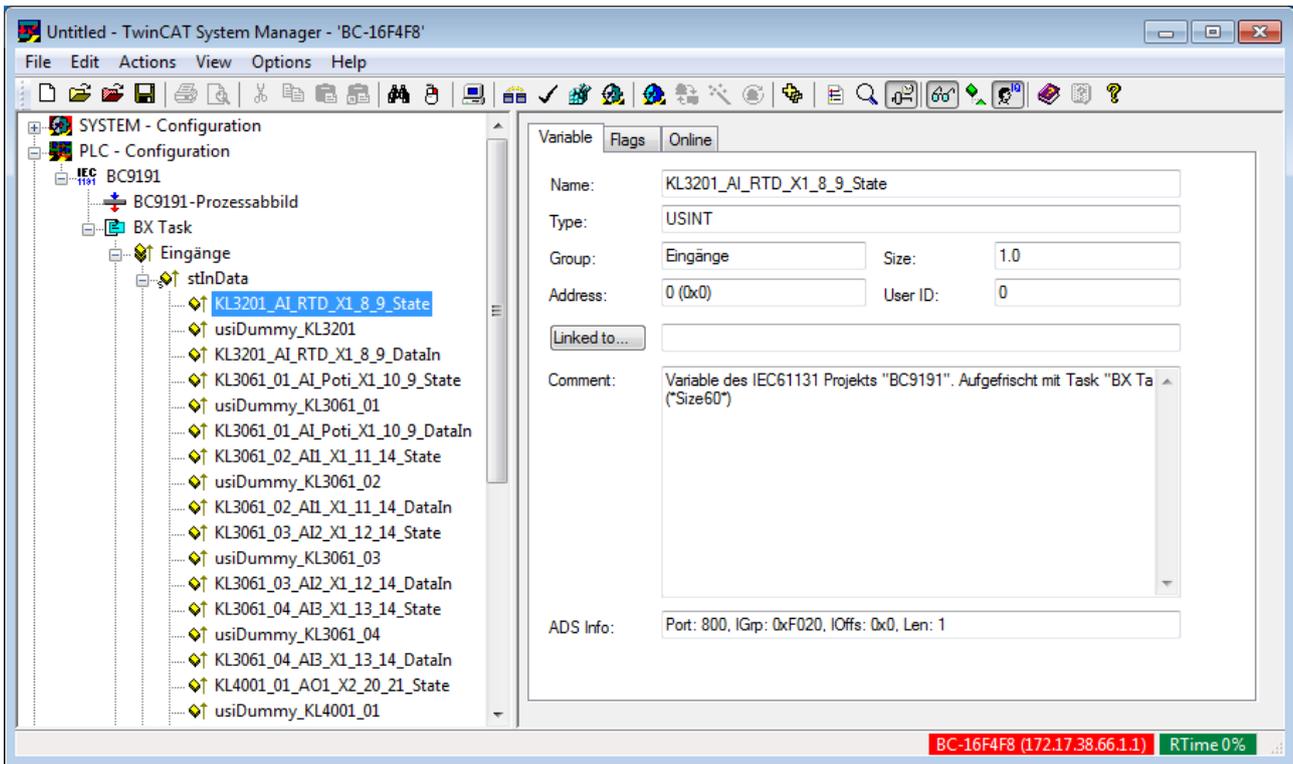
The logging table at the bottom shows the following entries:

LogDictionary	RelatedShapeltems	Timestamp	Summary	Description	Priority	OriginCallerMemberName
(Auflistung)	(Auflistung)	2/19/2018 9:07:30 AM	RegisterSymbol	newMqttSymbol_4 [DataAgentTopologyView.VMs.SymbolMqttVM]	INFO	RegisterSymbolVM
(Auflistung)	(Auflistung)	2/19/2018 9:07:30 AM	RegisterMappingLink	newOpcUASymbol2 to newOpcUASymbol2	INFO	RegisterMappingLinkVM
(Auflistung)	(Auflistung)	2/19/2018 9:07:30 AM	RegisterGateLinkVM		INFO	RegisterGateLinkVM
(Auflistung)	(Auflistung)	2/19/2018 9:07:30 AM	RegisterChannelLinkVM		INFO	RegisterChannelLinkVM
(Auflistung)	(Auflistung)	2/19/2018 9:07:30 AM	RegisterMappingLink	newOpcUASymbol1 to newOpcUASymbol1	INFO	RegisterMappingLinkVM
(Auflistung)	(Auflistung)	2/19/2018 9:07:30 AM	RegisterSymbol	newMqttSymbol_3 [DataAgentTopologyView.VMs.SymbolMqttVM]	INFO	RegisterSymbolVM

## 6.2.2 Verbindung mit einem BC9191 über ADS

Um den TC3 IoT Data Agent für den Zugriff auf einen BC9191 Controller über ADS zu konfigurieren, müssen die Symbole im TC3 IoT Data Agent-Konfigurator manuell konfiguriert werden, da der BC9191 Controller keine Symbolinformationen über ADS bereitstellt. Der Target Browser-Mechanismus kann nicht verwendet werden. Für jedes Symbol, das für die IoT-Kommunikation konfiguriert werden soll, müssen die IndexGroup/IndexOffset-Informationen konfiguriert werden.

Bei Eingangs-/Ausgangsvariablen sind diese Informationen im TwinCAT System Manager im Teil „ADS Info“ zu finden.



Im Fall von internen Variablen können diese Variablen auf verschiedene Weise für den ADS-Zugriff konfiguriert werden. Weitere Informationen finden Sie in der BC9191-Dokumentation.

Der folgende SPS-Code-Ausschnitt ist ein Beispiel:

```
VAR_GLOBAL
  iCounter1 AT%MB100 : INT;
  iCounter2 AT%MB200 : INT;
  iCounter3 AT%IB100 : INT;
  iCounter4 AT%IB200 : INT;
END_VAR
```

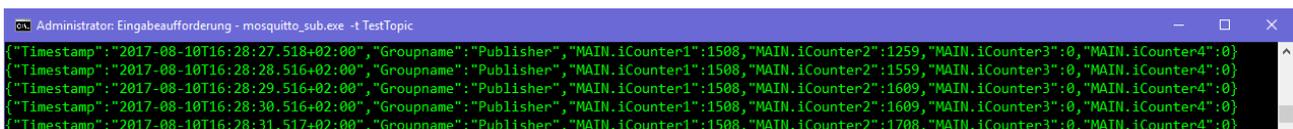
Die entsprechenden IndexGroup/IndexOffset-Kombinationen für diese beiden Variablen lauten wie folgt:

Variable	IndexGroup	IndexOffset
iCounter1	0x4020 (16416)	0x64 (100)
iCounter2	0x4020 (16416)	0xC8 (200)
iCounter3	0xF020 (61472)	0x64 (100)
iCounter4	0xF020 (61472)	0xC8 (200)

Der BC9191 kann der TC3 IoT Data Agent-Konfiguration als reguläres ADS-Gate hinzugefügt werden. Der ADS-Port muss auf 800 gesetzt werden, woraufhin der IoMode automatisch auf „direct“ gesetzt wird.

Wie beschrieben, müssen die Symbole mit Hilfe der entsprechenden IndexGroup/IndexOffset-Informationen manuell hinzugefügt werden. Als URN kann ein individueller Name Ihrer Wahl festgelegt werden.

Nach der Zuordnung dieser Subscriber-Symbole zu den entsprechenden Publisher-Symbolen, z. B. auf einem MQTT-Gate, werden die Variablen auf der Publisher-Seite angezeigt.



### 6.2.3 Verbindung mit einem TwinCAT I/O Task über ADS

Um den TC3 IoT Data Agent für den Zugriff auf einen TwinCAT I/O Task (mit Prozessabbild) über ADS zu konfigurieren, können die Symbole im TC3 IoT Data Agent-Konfigurator manuell, wenn der I/O Task nicht für die Erstellung von Symbolinformationen konfiguriert wurde, oder über den Target Browser konfiguriert werden.

#### TwinCAT I/O Task

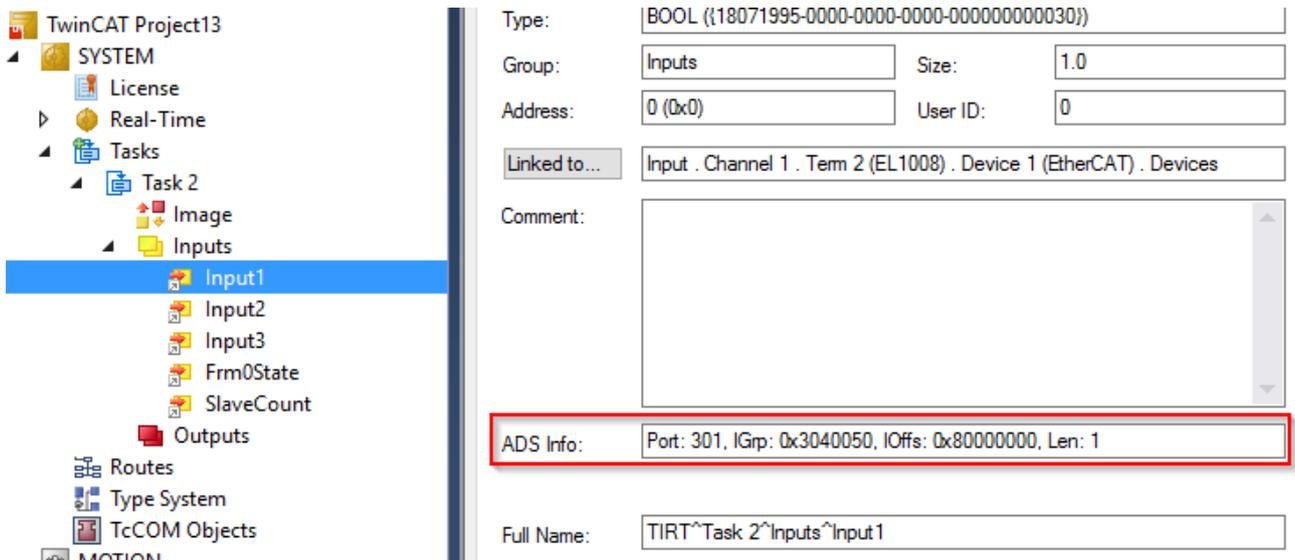
Ein TwinCAT I/O Task kann mit einem Prozessabbild konfiguriert werden, um Variablen von diesem Abbild direkt mit I/Os zu verbinden. Der TC3 IoT Data Agent kann dann über seinen ADS-Port auf den I/O Task zugreifen.

The screenshot displays the configuration interface for a TwinCAT I/O Task. The left pane shows a tree view of the project structure: TwinCAT Project13 > SYSTEM > License > Real-Time > Tasks > Task 2 > Image > Inputs > Input1, Input2, Input3, Frm0State, SlaveCount, and Outputs. The right pane shows the configuration for 'Task 2' with the following settings:

- Name: Task 2
- Port: 301 (highlighted with a red box)
- Object Id: 0x02010020
- Options:
  - Disable
  - Create symbols
  - Include external symbols
  - Extern sync
- Auto start:
- Auto Priority Management:
- Priority: 1
- Cycle ticks: 10 (ms)
- Start tick (modulo): 0
- Separate input update:
- Pre ticks: 0

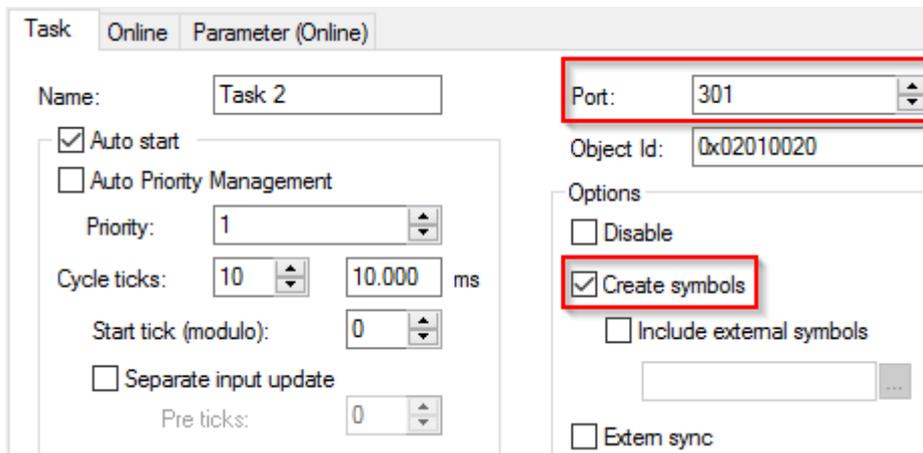
#### Manuelle Symbolkonfiguration

Auf die Prozessabbildvariablen kann über ihre IndexGroup/IndexOffset-Kombination zugegriffen werden. Diese Informationen müssen in den TC3 IoT Data Agent eingegeben werden.

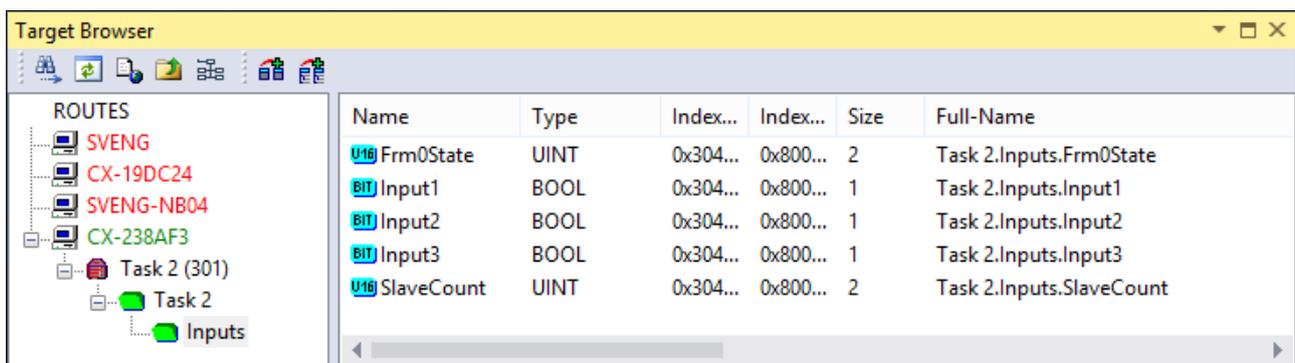


**Konfiguration über Target Browser**

Für einen TwinCAT I/O Task können Symbolinformationen erzeugt werden, indem die entsprechende Option in der I/O Task-Konfiguration aktiviert wird.



Nach Aktivierung dieser Konfiguration kann über den Target Browser auf die Symbolinformationen zugegriffen werden, und die Symbole können der TC3 IoT Data Agent-Konfiguration hinzugefügt werden.



**6.2.4 Verbindung mit einem TwinCAT TcCOM-Modul über ADS**

Um den TC3 IoT Data Agent für den Zugriff auf ein TwinCAT 3 TcCOM-Modul über ADS zu konfigurieren, können die Symbole im TC3 IoT Data Agent-Konfigurator manuell, wenn das TcCOM-Modul nicht für die Erstellung von Symbolinformationen konfiguriert wurde, oder über den Target Browser konfiguriert werden.

### TwinCAT 3 TcCOM-Modul

Auf ein TwinCAT 3 TcCOM-Modul kann über ADS über seinen entsprechenden ADS-Port zugegriffen werden. Dies ist in der Regel der Port des Task.

The screenshot shows the project tree on the left with 'Object1 (Module1)' expanded to show 'Inputs' and 'Outputs'. Below the tree is a table with the following data:

ID	Task	Name	Priority	Cycle Time (µs)	Task Port	Symbol Port
1	02010020	Task 2	1	10000	350	350

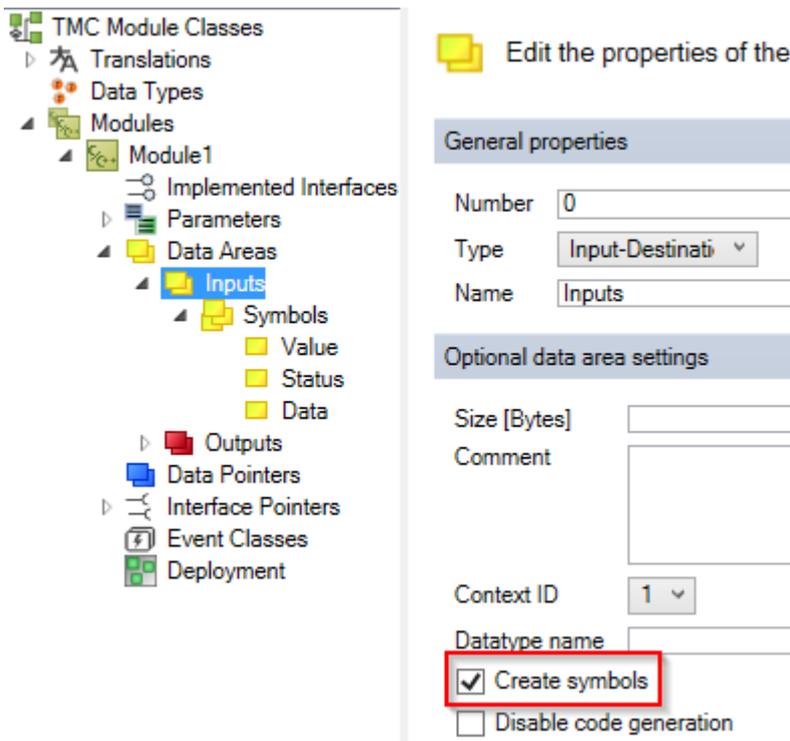
### Manuelle Symbolkonfiguration

Auf die Prozessabbildvariablen des TcCOM-Moduls kann über ihre IndexGroup/IndexOffset-Kombination zugegriffen werden. Diese Informationen müssen in den TC3 IoT Data Agent eingegeben werden.

The screenshot shows the configuration window for a variable. The 'ADS Info' field is highlighted with a red box and contains the text: 'Port: 350, IGrp: 0x1010020, IOffs: 0x80000000, Len: 4'. The 'Full Name' field contains: 'TIRC^TcCOM Objects^Object1 (Module1)^Inputs^Value'.

### Konfiguration über Target Browser

Symbolinformationen können für ein TwinCAT 3 TcCOM-Modul über den TMC Code Generator erzeugt werden.



Nach Aktivierung dieser Konfiguration kann über den Target Browser auf die Symbolinformationen zugegriffen werden, und die Symbole können der TC3 IoT Data Agent-Konfiguration hinzugefügt werden.

## 6.3 Northbound

### 6.3.1 Veröffentlichung von Daten an AWS IoT Core

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der Konfigurator verwendet wird, um Daten von einer TwinCAT 3 PLC-Laufzeit abzutasten und mit einem JSON-Datenformat an AWS IoT zu veröffentlichen.

#### ADS-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein ADS-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals Symbole hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (ADS)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die ADS-spezifischen Einstellungen im Eigenschaften-Fenster (AmsNetId, AdsPort, ...).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „Cyclic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  - 5. Öffnen Sie den Target Browser.
  - 6. Navigieren Sie zum ADS-Gerät und fügen Sie der Konfiguration einige Symbole hinzu, indem Sie sie zum Subscriber-Kanal ziehen.
- ⇒ Ein ADS-Gate und ein Subscriber-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

#### ● AMS Net ID

**i** Stellen Sie sicher, dass die AMS Net ID, die auf dem ADS-Gate konfiguriert wurde, dem ADS-Gerät entspricht, das Sie mit dem Target Browser ausgewählt haben.

## AWS IoT-Gate und Kanal konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein AWS IoT-Gate und einen Publisher-Kanal zu erstellen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (AWS IoT)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die AWS IoT-bezogenen Einstellungen im Eigenschaften-Fenster (Broker-Adresse, TLS-Einstellungen).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“, „Formatter“ auf „Simple JSON“ und „Topic“ auf „TestTopic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
- ⇒ Ein AWS IoT-Gate und ein Publisher-Kanal wurden erstellt.

## Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

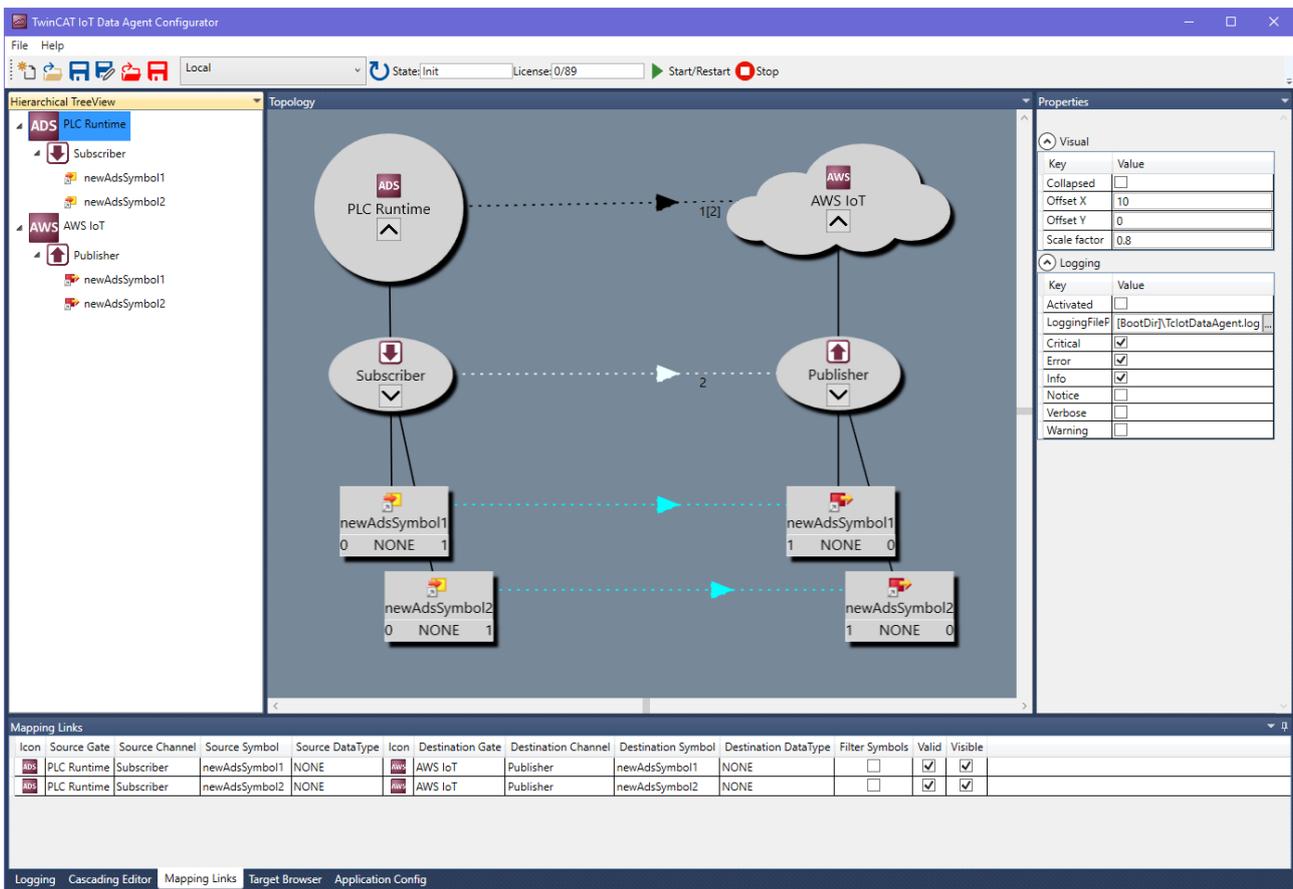
Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

## Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
  - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
  - ⇒ Die Konfiguration ist aktiviert.

Die resultierende Konfiguration sieht wie folgt aus:



### 6.3.2 Veröffentlichung von Daten an AWS Greengrass

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der Konfigurator verwendet wird, um Daten von einer TwinCAT 3 PLC-Laufzeit abzutasten und mit einem JSON-Datenformat an AWS Greengrass zu veröffentlichen.

#### ADS-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein ADS-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals Symbole hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (ADS)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die ADS-spezifischen Einstellungen im Eigenschaften-Fenster (AmsNetId, AdsPort, ...).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „Cyclic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  - 5. Öffnen Sie den Target Browser.
  - 6. Navigieren Sie zum ADS-Gerät und fügen Sie der Konfiguration einige Symbole hinzu, indem Sie sie zum Subscriber-Kanal ziehen.
- ⇒ Ein ADS-Gate und ein Subscriber-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

#### AMS Net ID

**i** Stellen Sie sicher, dass die AMS Net ID, die auf dem ADS-Gate konfiguriert wurde, dem ADS-Gerät entspricht, das Sie mit dem Target Browser ausgewählt haben.

## AWS IoT-Gate und Kanal konfigurieren

Eine Verbindung mit AWS Greengrass kann grundsätzlich wie eine Verbindung mit AWS IoT konfiguriert werden. Folgen Sie den Anweisungen Schritt für Schritt, um ein AWS Greengrass-Gate und einen Publisher-Kanal zu erstellen.

### ● CA certificate



Stellen Sie sicher, dass Sie das richtige CA-Zertifikat verwenden, das über die AWS CLI erhältlich ist.

✓ Die Topologieansicht ist aktiviert.

1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (AWS IoT)** aus.
2. Wählen Sie das neue Gate aus und konfigurieren Sie die AWS Greengrass-bezogenen Einstellungen im Eigenschaften-Fenster (IP-Adresse oder Hostname des Greengrass Core-Geräts und die TLS-Einstellungen).
3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“, „Formatter“ auf „Simple JSON“ und „Topic“ auf „TestTopic“. Stellen Sie sicher, dass „QoS“ auf „0“ gesetzt ist. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.

⇒ Ein AWS Greengrass-Gate und ein Publisher-Kanal wurden erstellt.

## Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

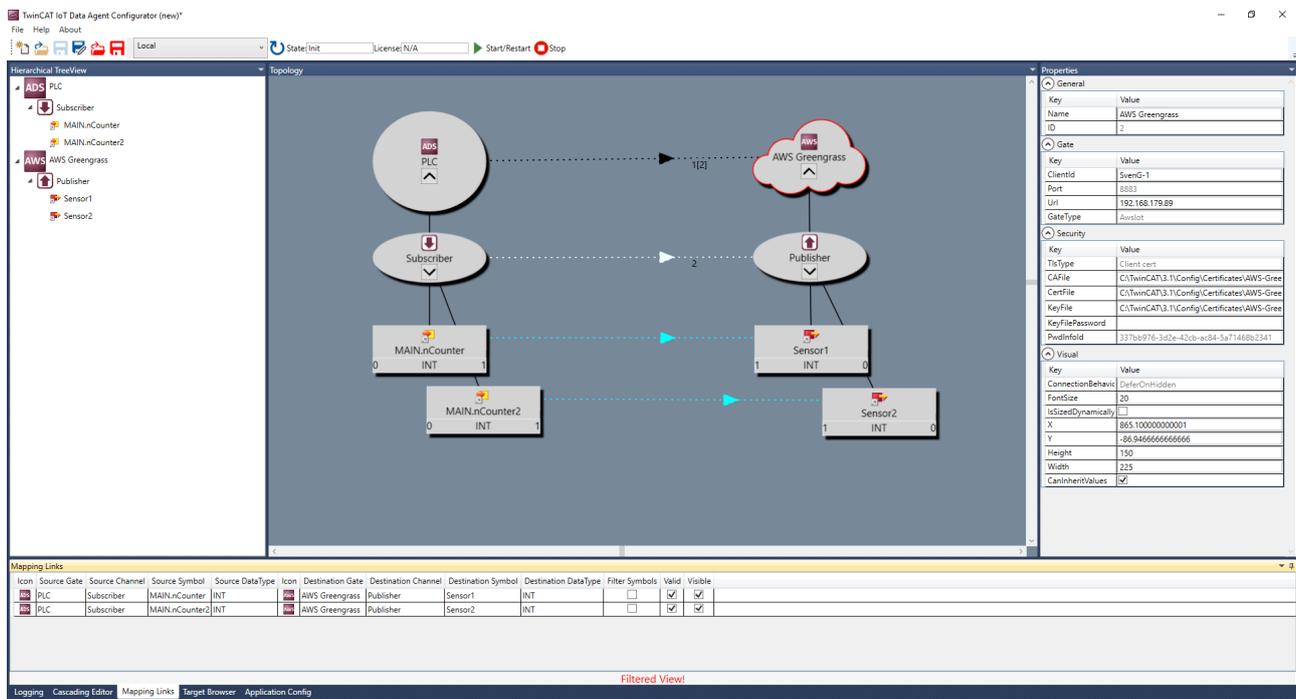
Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

## Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
  2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
    - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
  3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
- ⇒ Die Konfiguration ist aktiviert.

Die resultierende Konfiguration sieht wie folgt aus:



### 6.3.3 Veröffentlichung von Daten an IBM Watson IoT

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der Konfigurator verwendet wird, um Daten von einer TwinCAT 3 PLC-Laufzeit abzutasten und mit einem JSON-Datenformat an IBM Watson IoT zu veröffentlichen.

#### ● Einstellungen von IBM Watson IoT

**i** Für IBM-spezifische Konfigurationseinstellungen sehen Sie sich bitte die IBM Watson IoT-Dokumentation an. Ein guter Artikel über Erste Schritte ist hier zu finden: <https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/devices/mqtt.html>

#### ADS-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein ADS-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals Symbole hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (ADS)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die ADS-spezifischen Einstellungen im Eigenschaften-Fenster (AmsNetId, AdsPort, ...).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „Cyclic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  - 5. Öffnen Sie den Target Browser.
  - 6. Navigieren Sie zum ADS-Gerät und fügen Sie der Konfiguration einige Symbole hinzu, indem Sie sie zum Subscriber-Kanal ziehen.
- ⇒ Ein ADS-Gate und ein Subscriber-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

#### ● AMS Net ID

**i** Stellen Sie sicher, dass die AMS Net ID, die auf dem ADS-Gate konfiguriert wurde, dem ADS-Gerät entspricht, das Sie mit dem Target Browser ausgewählt haben.

## MQTT-Gate und Kanal konfigurieren

Das von IBM Watson IoT verwendete Transportprotokoll ist MQTT. Folgen Sie den Anweisungen Schritt für Schritt, um ein MQTT-Gate und einen Publisher-Kanal zu erstellen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (MQTT)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die IBM Watson IoT-bezogenen Einstellungen im Eigenschaften-Fenster (Broker-Adresse, TLS-Einstellungen, Benutzername/Passwort, ClientID) wie in der IBM Watson IoT-Dokumentation beschrieben.
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“, „Formatter“ auf „Simple JSON“ und „Topic“ auf „iot-2/evt/<eventId>/fmt/<format>“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen. Ersetzen Sie <eventId> und <format> wie in der IBM Watson IoT-Dokumentation beschrieben.
- ⇒ Ein AWS IoT-Gate und ein Publisher-Kanal wurden erstellt.

## Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

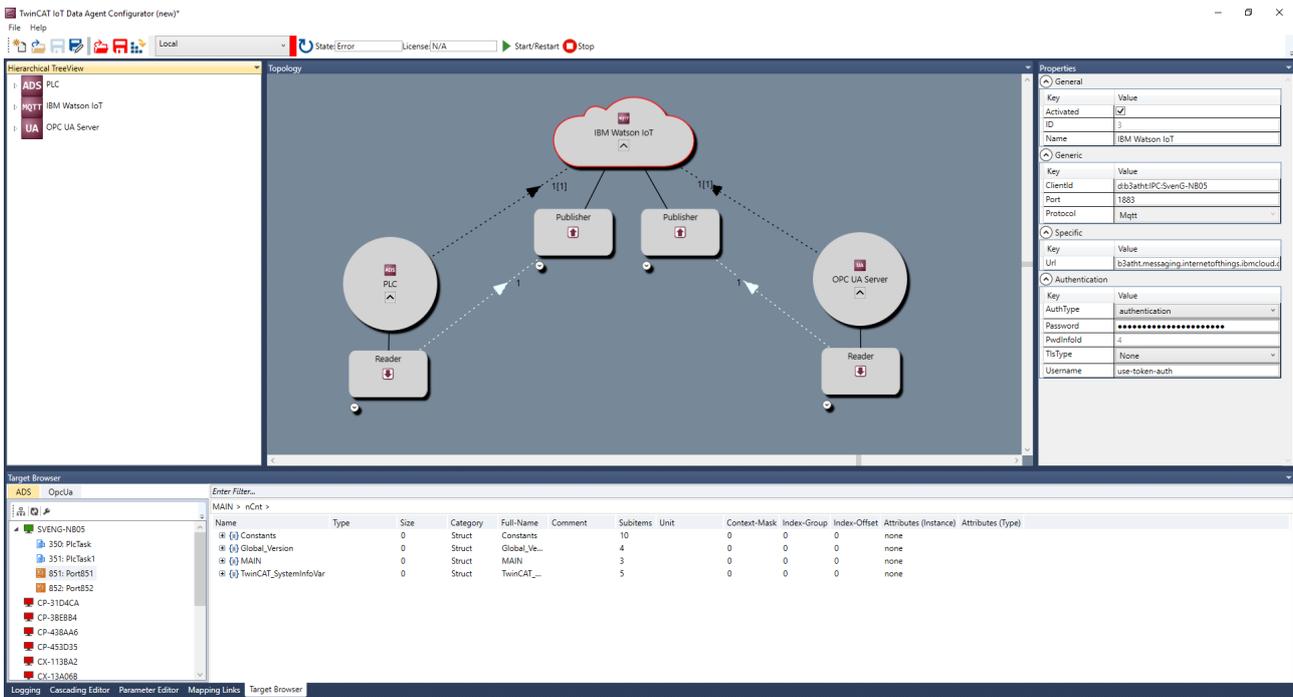
Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

## Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
  2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
    - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
  3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
- ⇒ Die Konfiguration ist aktiviert.

Die resultierende Konfiguration kann wie folgt aussehen:



### 6.3.4 Veröffentlichung von Daten an Microsoft Azure IoT Hub

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der Konfigurator verwendet wird, um Daten von einer TwinCAT 3 PLC-Laufzeit abzutasten und mit einem JSON-Datenformat an Microsoft Azure IoT Hub zu veröffentlichen.

#### ADS-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein ADS-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals Symbole hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.
- 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (ADS)** aus.
- 2. Wählen Sie das neue Gate aus und konfigurieren Sie die ADS-spezifischen Einstellungen im Eigenschaften-Fenster (AmsNetId, AdsPort, ...).
- 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
- 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „Cyclic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
- 5. Öffnen Sie den Target Browser.
- 6. Navigieren Sie zum ADS-Gerät und fügen Sie der Konfiguration einige Symbole hinzu, indem Sie sie zum Subscriber-Kanal ziehen.

⇒ Ein ADS-Gate und ein Subscriber-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

#### AMS Net ID

**I** Stellen Sie sicher, dass die AMS Net ID, die auf dem ADS-Gate konfiguriert wurde, dem ADS-Gerät entspricht, das Sie mit dem Target Browser ausgewählt haben.

#### Azure IoT Hub-Gate und Kanal konfigurieren

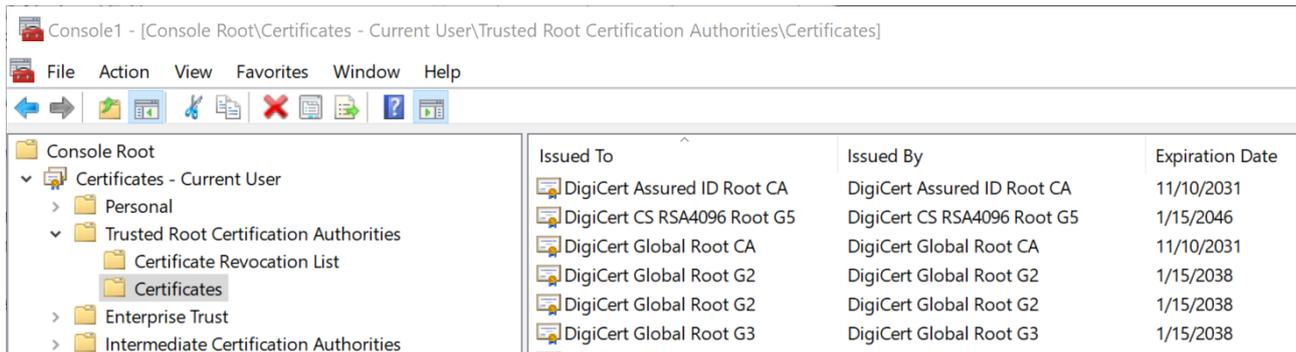
Folgen Sie den Anweisungen Schritt für Schritt, um ein Azure IoT Hub-Gate und einen Publisher-Kanal zu erstellen.

- ✓ Die Topologieansicht ist aktiviert.
- 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (IoT Hub)** aus.
- 2. Wählen Sie das neue Gate aus und konfigurieren Sie die IoT Hub-bezogenen Einstellungen im Eigenschaften-Fenster (Gerätename, primärer Schlüssel, CA-Zertifikat).

3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
  4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“ und „Formatter“ auf „Simple JSON“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
- ⇒ Ein Azure IoT Hub-Gate und ein Publisher-Kanal wurden erstellt.

## CA certificate

Bei der Herstellung einer Verbindung mit Microsoft Azure IoT Hub muss ein CA-Zertifikat festgelegt werden. Zum Zeitpunkt der Erstellung dieses Dokumentationsartikels migrierte Microsoft alle Azure IoT Hub Instanzen auf eine neue Root CA (Baltimore Cyber Trust Root CA -> DigiCert Global Root G2). Bitte konsultieren Sie daher immer auch die Microsoft Dokumentation, um die aktuell gültige Root CA zu erfahren. In den meisten Fällen der zugehörige Public Key der jeweiligen Root CA aus der Microsoft Windows Zertifikatkonsole extrahiert werden (Start > Run > mmc.exe, dann das Snapshot „Zertifikate“ hinzufügen). Siehe auch unten stehenden Screenshot als Vergleich.



## Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

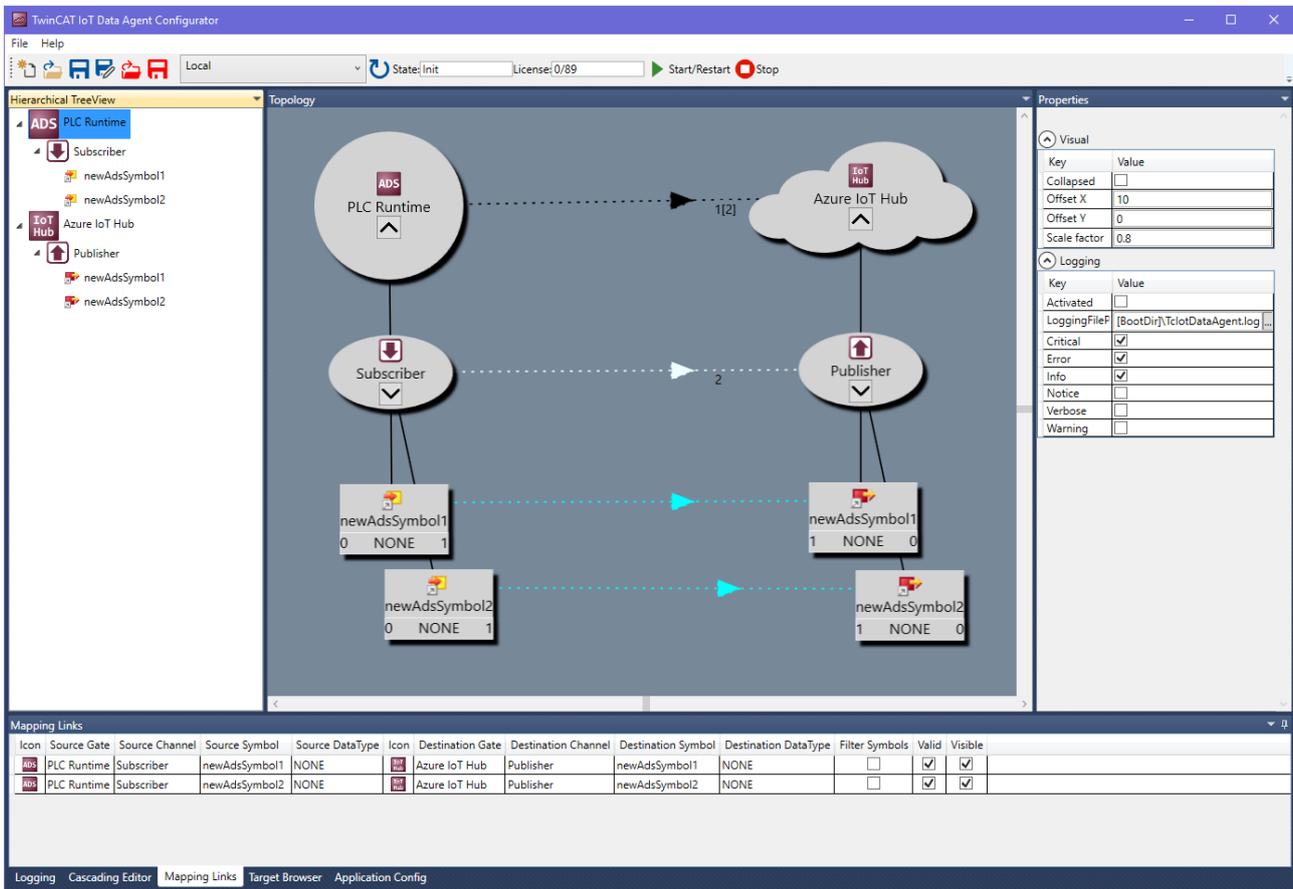
Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

## Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
  - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
  - ⇒ Die Konfiguration ist aktiviert.

Die resultierende Konfiguration sieht wie folgt aus:



### 6.3.5 Veröffentlichung von Daten an einen MQTT-Message-Broker

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der Konfigurator verwendet wird, um Daten von einer TwinCAT 3 PLC-Laufzeit abzutasten und mit einem JSON-Datenformat an einen MQTT-Message-Broker zu veröffentlichen.

#### ADS-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein ADS-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals Symbole hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (ADS)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die ADS-spezifischen Einstellungen im Eigenschaften-Fenster (AmsNetId, AdsPort, ...).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „Cyclic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  - 5. Öffnen Sie den Target Browser.
  - 6. Navigieren Sie zum ADS-Gerät und fügen Sie der Konfiguration einige Symbole hinzu, indem Sie sie zum Subscriber-Kanal ziehen.
- ⇒ Ein ADS-Gate und ein Subscriber-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

#### AMS Net ID

**i** Stellen Sie sicher, dass die AMS Net ID, die auf dem ADS-Gate konfiguriert wurde, dem ADS-Gerät entspricht, das Sie mit dem Target Browser ausgewählt haben.

#### MQTT-Gate und Kanal konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein MQTT-Gate und einen Publisher-Kanal zu erstellen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (MQTT)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die MQTT-bezogenen Einstellungen im Eigenschaften-Fenster (Broker-Adresse, ...).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“, „Formatter“ auf „Simple JSON“ und „Topic“ auf „TestTopic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
- ⇒ Ein MQTT-Gate und ein Publisher-Kanal wurden erstellt.

### Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

#### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

#### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

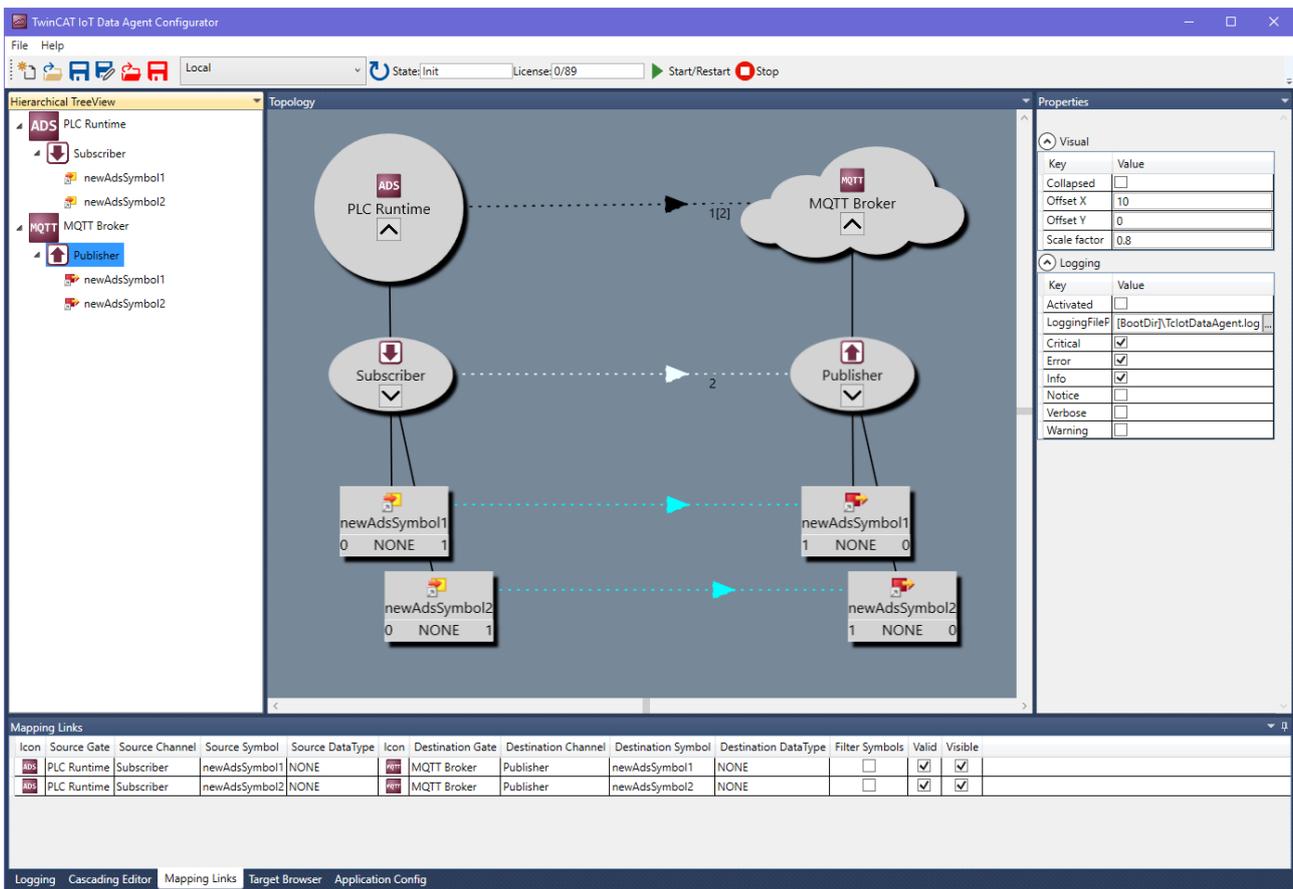
Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

### Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
  2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
    - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
  3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
- ⇒ Die Konfiguration ist aktiviert.

Die resultierende Konfiguration sieht wie folgt aus:



### 6.3.6 Veröffentlichung von Daten an die App TwinCAT IoT Communicator

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der Konfigurator verwendet wird, um Daten von einer TwinCAT 3 PLC-Laufzeit abzutasten und mit einem TwinCAT JSON-Datenformat an einen MQTT-Message-Broker zu veröffentlichen, damit sie von der Smartphone-App TwinCAT IoT Communicator visualisiert werden.

#### ADS-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein ADS-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals Symbole hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.
  - 1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (ADS)** aus.
  - 2. Wählen Sie das neue Gate aus und konfigurieren Sie die ADS-spezifischen Einstellungen im Eigenschaften-Fenster (AmsNetId, AdsPort, ...).
  - 3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
  - 4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „Cyclic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  - 5. Öffnen Sie den Target Browser.
  - 6. Navigieren Sie zum ADS-Gerät und fügen Sie der Konfiguration einige Symbole hinzu, indem Sie sie zum Subscriber-Kanal ziehen.
- ⇒ Ein ADS-Gate und ein Subscriber-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

#### **i** AMS Net ID

Stellen Sie sicher, dass die AMS Net ID, die auf dem ADS-Gate konfiguriert wurde, dem ADS-Gerät entspricht, das Sie mit dem Target Browser ausgewählt haben.

## MQTT-Gate und Kanal konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein MQTT-Gate und einen Publisher-Kanal zu erstellen. Der Kanal wird so konfiguriert, dass er die Topic-Struktur und das Datenformat für den TwinCAT IoT Communicator darstellt.

1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (MQTT)** aus.
  2. Wählen Sie das neue Gate aus und konfigurieren Sie die MQTT-bezogenen Einstellungen auf dem neuen Gate (Broker-Adresse, ...).
  3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add channel (Publisher)** aus.
  4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“ und „Formatter“ auf „TwinCAT JSON“. Setzen Sie „Topic“ auf <MainTopic>/<DeviceName>/TclotCommunicator/Json/Tx/Data. Die Teile <MainTopic> und <DeviceName> müssen gemäß Ihrem Communicator-Setup ersetzt werden.
  5. Setzen Sie „StatusInfo Topic“ auf <MainTopic>/<DeviceName>/TclotCommunicator/Desc. Die Teile <MainTopic> und <DeviceName> müssen gemäß Ihrem Communicator-Setup ersetzt werden.
- ⇒ Ein MQTT-Gate und ein Publisher-Kanal wurden erstellt.

## Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

## Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
  2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
    - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
  3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
- ⇒ Die Konfiguration ist aktiviert.

## 6.3.7 Abonnieren eines MQTT-Message-Brokers

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der Konfigurator verwendet wird, um einen MQTT-Message-Broker zu abonnieren und empfangene Daten in ADS-Variablen zu schreiben.

### MQTT-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein MQTT-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals Symbole hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.

1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (MQTT)** aus.
  2. Wählen Sie das neue Gate aus und konfigurieren Sie die MQTT-bezogenen Einstellungen im Eigenschaften-Fenster (Broker-Adresse, ...).
  3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
  4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“ und „Formatter“ auf „Simple JSON“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen. Stellen Sie sicher, dass das MQTT-Topic dem Topic-Namen entspricht, zu dem Sie Daten empfangen.
  5. Rechtsklicken Sie auf den Kanal und wählen Sie **Add Symbol** aus. Konfigurieren Sie die symbolbezogenen Einstellungen, z. B. den Symbolnamen. Bei einem JSON-Datenformat muss der Name mit einem JSON-Schlüssel in dem JSON-Dokument übereinstimmen, das der TC3 IoT Data Agent empfangen hat. Stellen Sie sicher, dass der richtige Datentyp eingestellt wird.
- ⇒ Ein MQTT-Gate und ein Subscriber-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

### ● Data Format

**I** Stellen Sie sicher, dass das Datenformat der Nutzlast der empfangenen Nachricht dem konfigurierten Datenformat entspricht. Anderenfalls kann der TC3 IoT Data Agent den Inhalt der Nachricht nicht parsen und die Daten nicht verarbeiten.

## ADS-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein ADS-Gate und einen Publisher-Kanal zu erstellen. Fügen Sie der Konfiguration des Kanals Symbole hinzu.

- ✓ Die Topologieansicht ist aktiviert.
1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (ADS)** aus.
  2. Wählen Sie das neue Gate aus und konfigurieren Sie die ADS-bezogenen Einstellungen im Eigenschaften-Fenster (AmsNetId, AdsPort, ...).
  3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
  4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen
  5. Öffnen Sie den Target Browser.
  6. Navigieren Sie zum ADS-Gerät und fügen Sie der Konfiguration die Symbole hinzu, indem Sie sie zum Publisher-Kanal ziehen.
- ⇒ Ein ADS-Gate und ein Publisher-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

### ● AMS Net ID

**I** Stellen Sie sicher, dass die AMS Net ID, die auf dem ADS-Gate konfiguriert wurde, dem ADS-Gerät entspricht, das Sie mit dem Target Browser ausgewählt haben.

## Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

## Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
  - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
  - ⇒ Die Konfiguration ist aktiviert.

## 6.3.8 Verwendung eines Microsoft Azure IoT Hub Gerätezwillingings

Im folgenden Abschnitt wird Schritt für Schritt beschrieben, wie der Konfigurator verwendet wird, um eine Verbindung zu Microsoft Azure IoT Hub und dem Gerätezwillingdienst von Microsoft Azure IoT Hub herzustellen.

### ADS-Gate, Kanal und Symbole konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein ADS-Gate und einen Subscriber-Kanal zu erstellen und der Konfiguration des Kanals Symbole hinzuzufügen.

- ✓ Die Topologieansicht ist aktiviert.
1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (ADS)** aus.
  2. Wählen Sie das neue Gate aus und konfigurieren Sie die ADS-spezifischen Einstellungen im Eigenschaften-Fenster (AmsNetId, AdsPort, ...).
  3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Subscriber)** aus.
  4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „Cyclic“. Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  5. Öffnen Sie den Target Browser.
  6. Navigieren Sie zum ADS-Gerät und fügen Sie der Konfiguration einige Symbole hinzu, indem Sie sie zum Subscriber-Kanal ziehen.
- ⇒ Ein ADS-Gate und ein Subscriber-Kanal wurden erstellt. Dem Kanal wurden Symbole hinzugefügt.

### ● AMS Net ID



Stellen Sie sicher, dass die AMS Net ID, die auf dem ADS-Gate konfiguriert wurde, dem ADS-Gerät entspricht, das Sie mit dem Target Browser ausgewählt haben.

### Azure IoT Hub-Gate und Gerätezwillingingskanal konfigurieren

Folgen Sie den Anweisungen Schritt für Schritt, um ein Azure IoT Hub-Gate und Kanäle für Telemetriedaten und den Gerätezwilling zu erstellen.

- ✓ Die Topologieansicht ist aktiviert
1. Rechtsklicken Sie auf die leere Fläche und wählen Sie **Add Gate (IoT Hub)** aus.
  2. Wählen Sie das neue Gate aus und konfigurieren Sie die IoT Hub-bezogenen Einstellungen im Eigenschaften-Fenster (Gerätename, primärer Schlüssel, CA-Zertifikat).
  3. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Publisher)** aus.
  4. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“, „Formatter“ auf „Simple JSON“ und „Name“ auf „Telemetry Data“ (optional). Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  5. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Device Twin Publisher)** aus.
  6. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“, „Formatter“ auf „Simple JSON“ und „Name“ auf „Device Twin“ (optional). Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
  7. Rechtsklicken Sie auf das neue Gate und wählen Sie **Add Channel (Device Twin Subscriber)** aus.

8. Wählen Sie den neuen Kanal aus und setzen Sie „SamplingMode“ auf „OnChange“, „Formatter“ auf „Simple JSON“ und „Name“ auf „Device Twin GET“ (optional). Belassen Sie bei allen anderen Feldern die Standardeinstellungen.
- ⇒ Sie haben eine Azure IoT Hub-Verbindung und drei Kanäle – einen für Telemetriedaten und zwei für den Gerätezwilling – konfiguriert. In dem Schritt müssen Sie entscheiden, welche Quellsymbole vom ADS-Gate als Telemetriedaten behandelt werden und welche Symbole über den Publisher-Kanal des Gerätezwillings an den Gerätezwilling gesendet werden sollen. Der Subscriber-Kanal des Gerätezwillings kann verwendet werden, um Updates für den Gerätezwilling von cloudbasierten Backend-Anwendungen (sogenannte „gewünschte“ Updates) abzurufen. Jede erwartete Eigenschaft muss diesem Kanal als Symbol hinzugefügt werden, wobei der Eigenschaftsname als Symbolname und URN zu verwenden ist.

### Zuordnung (Mapping) erstellen

Folgen Sie den Anweisungen Schritt für Schritt, um eine Zuordnung zwischen Quell- und Zielsymbolen zu erstellen. Zu beachten ist, dass eine Zuordnung nur zwischen Kanälen mit verschiedenen Rollen (z. B. zwischen einem Subscriber- und einem Publisher-Kanal und umgekehrt) möglich ist.

#### Drag-and-drop – Kanal:

Halten Sie die Strg-Taste gedrückt und ziehen Sie den Subscriber-Kanal zum Publisher-Kanal.

Der Konfigurator erstellt automatisch neue Symbole auf dem Publisher-Kanal und eine Zuordnung zwischen dem Quell- und Zielsymbol.

#### Drag-and-drop – Symbol:

Halten Sie die Strg-Taste gedrückt und ziehen Sie ein Symbol vom Subscriber-Kanal entweder zu einem vorhandenen Symbol auf dem Publisher-Kanal oder zum Publisher-Kanal selbst.

Der Konfigurator verbindet das Quell- und Zielsymbol miteinander oder, wenn das Symbol zu einem Kanalobjekt gezogen wurde, erstellt das Zielsymbol automatisch

### Konfiguration aktivieren

Folgen Sie den Anweisungen Schritt für Schritt, um die derzeit geöffnete Konfiguration zu aktivieren.

1. Wählen Sie das Zielsystem über die Symbolleiste aus.
2. Klicken Sie auf die Schaltfläche **Activate Configuration**.
  - ⇒ Es öffnet sich ein Dialog mit der Frage, ob der Ziel-TC3 IoT Data Agent in den Run-Modus wechseln soll.
3. Bestätigen Sie den Dialog mit **Yes**, wenn Sie den Modus wechseln möchten. Anderenfalls wählen Sie **No** aus.
  - ⇒ Die Konfiguration ist aktiviert.

Die resultierende Konfiguration kann wie folgt aussehen:

TwinCAT IoT Data Agent Configurator (new)

File Help Local State/Run License(1/244) Start/Restart Stop

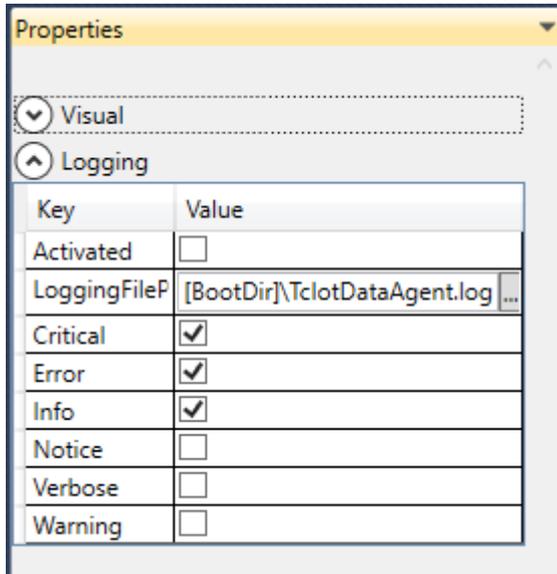
The screenshot displays the TwinCAT IoT Data Agent Configurator interface. The main window shows a topology diagram with components like PLC, C2D Writer, C2D Receiver, Telemetry Reader, Telemetry Publisher, Twin Reader, Twin Publisher, Twin Subscriber, and IoT Functions Twin. The diagram is connected to a Microsoft Azure IoT Hub. The left sidebar shows a Hierarchical TreeView with components like ADS PLC, Telemetry Reader, Twin Reader, Twin Writer, C2D Writer, Microsoft Azure IoT Hub, Telemetry Publisher, Twin Publisher, Twin Subscriber, IoT Functions Twin, and C2D Receiver. The right sidebar shows Properties for Logging, with a table of Key-Value pairs and checkboxes for various logging levels. The bottom window shows a Target Browser with a table of variables.

Name	Type	Size	Category	Full-Name	Comment	Subitems	Unit	Context-Mask	Index-Group	Index-Offset	Attributes (Instance)	Attributes (Type)
# (0) GVL_Audi_OpClea	Struct	0	Struct	GVL_Audi...		32		0	0	0	none	
# (0) MAIN	Struct	0	Struct	MAIN		5		0	0	0	none	
# bStart	BOOL	1	Primitive	MAIN.bSt...		0		0	4040	5DCBE	[Area: Local]	none
# bTrigger	BOOL	1	Primitive	MAIN.bTr...		0		0	4040	5EBAC	[Area: Local]	none
# fbTest	FB_Test	16	Struct	MAIN.fbT...		2		0	4040	5E120	[OPC-UA-DA: 1]	none
# nCounter1	INT	2	Primitive	MAIN.nC...		0		0	4040	5DCBC	[OPC-UA-DA: 1]	none
# nCounter2	LREAL	8	Primitive	MAIN.nC...		0		0	4040	5E118	[OPC-UA-DA: 1]	none
# TwinCAT_SystemInfoVar	Struct	0	Struct	TwinCAT...		5		0	0	0	none	

## 7 Anhang

### 7.1 Fehlerprotokollierung

Zur Fehlersuche kann der TC3 IoT Data Agent eine Protokolldatei erzeugen, die auf der Grundlage verschiedener Protokollierungsebenen gefüllt werden kann. Alle erforderlichen Einstellungen können über das Eigenschaften-Fenster des Konfigurators konfiguriert werden. Klicken Sie dazu einfach auf eine leere Stelle in der Topologieansicht und konfigurieren Sie die Protokollierungseinstellungen im Eigenschaften-Fenster.



Zu beachten ist, dass alle Einstellungen mit der derzeit geöffneten Konfiguration verbunden sind und für jede Konfiguration individuell festgelegt werden müssen. Das Standardverzeichnis, in dem die Protokolldatei erstellt wird, ist das TwinCAT-Bootverzeichnis. Der Platzhalter [BootDir] wählt das TwinCAT-Bootverzeichnis automatisch aus.

#### **i** Logging-Fenster

Das Logging-Fenster im Konfigurator zeigt nur protokollierte Ereignisse an, die mit dem Konfigurator in Zusammenhang stehen. Um ein Protokoll für den TC3 IoT Data Agent-Hintergrunddienst zu erstellen, sind die vorstehenden Einstellungen erforderlich.

### 7.2 Fehlerdiagnose

Dieser Dokumentationsartikel erläutert einige häufige Konfigurations- oder Setup-Fehler und könnte eine gute Möglichkeit bieten, um Fehler in Ihrem Setup zu beheben. Auch die Erstellung einer Fehlerprotokollierung [[▶ 83](#)] könnte hilfreich sein.

Verhalten	Beschreibung
Das Symbol in der Taskleiste wird nach dem Start ROT	<p>Prüfen Sie nochmals alle Einstellungen und aktivieren Sie die <a href="#">Fehlerprotokollierung</a> [► 83]. Prüfen Sie außerdem Folgendes:</p> <ul style="list-style-type: none"> <li>• Kann sich der TC3 IoT Data Agent mit allen konfigurierten Gates verbinden (siehe unten)?</li> <li>• Sind genug Lizenzen vorhanden?</li> </ul>
Verbindung zum MQTT-Message-Broker konnte nicht hergestellt werden	<p>Prüfen Sie Folgendes:</p> <ul style="list-style-type: none"> <li>• Stellen Sie sicher, dass der Message-Broker für das Gerät, auf dem der TC3 IoT Data Agent ausgeführt wird, erreichbar ist. Der MQTT-Standardport ist 1883/tcp für unverschlüsselte MQTT-Kommunikation und 8883/tcp, wenn MQTT-TLS verwendet wird. Stellen Sie sicher, dass diese Ports in Ihrer Firewall geöffnet sind (auf der Seite des Message-Brokers für eingehenden Zugriff und auf der Seite des TC3 IoT Data Agent für ausgehenden Zugriff).</li> <li>• Stellen Sie sicher, dass Sie die richtigen Zugriffsdaten (ggf.) für den Message-Broker verwenden</li> <li>• Prüfen Sie die Konnektivität mit dem Message-Broker, indem Sie einen anderen MQTT-Client verwenden, z. B. Mqtt.Fx, MqttSpy oder die Mosquitto Befehlszeilentools.</li> </ul>
Verbindung zum ADS-Gerät konnte nicht hergestellt werden	<p>Prüfen Sie Folgendes:</p> <ul style="list-style-type: none"> <li>• Stellen Sie sicher, dass Sie den richtigen ADS-Port und die richtige AmsNetID für das Remote-Gerät verwenden</li> <li>• Stellen Sie sicher, dass auf dem Remote-Gerät keine Firewall-Einstellungen vorhanden sind, die den ADS-Zugriff verhindern.</li> <li>• Stellen Sie sicher, dass der Computer, auf dem der TC3 IoT Data Agent ausgeführt wird, über eine ADS-Route verfügt, die für das entsprechende TwinCAT-Zielgerät konfiguriert ist. (Nur anwendbar, wenn sich der TC3 IoT Data Agent und die ADS-Ziellaufzeit auf verschiedenen Computern befinden.)</li> </ul>
Verbindung zum OPC UA-Server konnte nicht hergestellt werden	<p>Prüfen Sie Folgendes:</p> <ul style="list-style-type: none"> <li>• Stellen Sie sicher, dass auf dem Computer, auf dem der OPC UA-Server ausgeführt wird, keine Firewall-Einstellungen vorhanden sind, die den Remote-Zugriff auf den Server verhindern.</li> <li>• Prüfen Sie die lokalen Firewall-Einstellungen auf dem TC3 IoT Data Agent-Gerät.</li> </ul>
Verbindung zu Azure IoT Hub konnte nicht hergestellt werden	<p>Prüfen Sie Folgendes:</p> <ul style="list-style-type: none"> <li>• Stellen Sie sicher, dass Sie den richtigen IoT Hub-Hostnamen verwenden</li> <li>• Stellen Sie sicher, dass Sie auf Azure IoT Hub ein Gerät erstellt haben, z. B. mit Hilfe des Azure IoT Explorers, und dass Sie im TC3 IoT Data Agent-Konfigurator die richtigen Gerätedaten für dieses Gerät eingegeben haben</li> <li>• Stellen Sie sicher, dass Sie das richtige CA-Zertifikat für Azure IoT Hub verwendet haben und dass das CA-Zertifikat in einem Base64-codierten Dateiformat verwendet wird.</li> <li>• Stellen Sie sicher, dass Sie nicht dieselben Zugriffsdaten auf einem anderen Gerät verwenden. Mit demselben Satz Zugriffsdaten kann nur eine Verbindung zu IoT Hub hergestellt werden.</li> </ul>

Verhalten	Beschreibung
Verbindung zu AWS IoT konnte nicht hergestellt werden	<p>Prüfen Sie Folgendes:</p> <ul style="list-style-type: none"> <li>• Prüfen Sie die Konnektivität mit AWS IoT, indem Sie einen anderen MQTT-Client verwenden, z. B. Mqtt.Fx, MqttSpy oder die Mosquitto Befehlszeilentools.</li> <li>• Stellen Sie sicher, dass keine Firewalls den TCP-Port 8883 (ausgehend) blockieren, der für die Konnektivität mit AWS IoT erforderlich ist.</li> <li>• Stellen Sie sicher, dass Sie den richtigen Satz Zertifikate verwendet haben (wie auf der Managementwebsite von AWS IoT bereitgestellt)</li> </ul>
Die Fehlermeldung „InvalidServerCert“ wird angezeigt	<p>Diese Fehlermeldung bedeutet, dass der Data Agent bei der Überprüfung des Server-Zertifikats auf ein Problem gestoßen ist und die Überprüfung nicht erfolgreich war. Gründe dafür können beispielsweise sein:</p> <ul style="list-style-type: none"> <li>• Der CommonName oder der Subject Alternative Name (SAN) im Server-Zertifikat ist nicht gleich wie die Adresse, die für die Verbindung zum Server genutzt wird.</li> <li>• Der Server-Zertifikat ist abgelaufen.</li> </ul>
Das Serverzertifikat konnte nicht validiert werden, obwohl das richtige CA Zertifikat referenziert wurde.	<p>Einige Betriebsumgebungen verwenden eine sogenannte CA Chain, bestehend aus Root und einer oder mehrerer Intermediate CAs. Das im Data Agent referenzierte CA Zertifikat muss in diesem Fall die komplette CA Chain beinhalten, damit das Serverzertifikat korrekt überprüft werden kann.</p>

### 7.3 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

#### Downloadfinder

Unser Downloadfinder beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

#### Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: [www.beckhoff.com](http://www.beckhoff.com)

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

#### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157  
E-Mail: support@beckhoff.com

**Beckhoff Service**

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460  
E-Mail: service@beckhoff.com

**Beckhoff Unternehmenszentrale**

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49 5246 963-0  
E-Mail: info@beckhoff.com  
Internet: [www.beckhoff.com](http://www.beckhoff.com)



Mehr Informationen:  
**[www.beckhoff.com/tf6720](http://www.beckhoff.com/tf6720)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

