

# X20BC008T

## 1 Allgemeines

### 1.1 Mitgeltende Dokumente

Weiterführende und ergänzende Informationen sind den folgenden gelisteten Dokumenten zu entnehmen.

#### Mitgeltende Dokumente

Dokumentname	Titel
MAX20	<a href="#">X20 System Anwenderhandbuch</a>
MAEMV	<a href="#">Installations- / EMV-Guide</a>

### 1.2 Bestelldaten

Bestellnummer	Kurzbeschreibung	Abbildung
	<b>Bus Controller</b>	
X20BC008T	X20 Bus Controller, 1 OPC UA FX Ethernet Schnittstelle, integrierter 2-fach Switch, 2x RJ45, Busbasis, Einspeisemodul und Feldklemme gesondert bestellen!	
	<b>Erforderliches Zubehör</b>	
	<b>Feldklemmen</b>	
X20TB12	X20 Feldklemme, 12-polig, 24 VDC codiert	
	<b>Systemmodule für Bus Controller</b>	
X20BB80X	X20 Bus Controller Basis für X20BC008T und X20 Einspeisemodul, X20 Abschlussplatten links und rechts X20AC0SL1/ X20AC0SR1 beiliegend	
X20PS9400	X20 Einspeisemodul, für Bus Controller und interne I/O-Versorgung, X2X Link Versorgung	
X20PS9402	X20 Einspeisemodul, für Bus Controller und interne I/O-Versorgung, X2X Link Versorgung, Einspeisung galvanisch nicht getrennt	

Tabelle 1: X20BC008T - Bestelldaten

### 1.3 Modulbeschreibung

Der Bus Controller stellt OPC UA FX Funktionen zur Verfügung. Beliebige OPC UA Clients können damit auf die Daten der an den Bus Controller angeschlossenen I/O-Module lesend und schreibend zugreifen.

- Kommunikationstechnologie: OPC UA Field Exchange (FX)
- I/O-Konfiguration über OPC UA FX
- Minimale Zykluszeit 400 µs
- Integrierter Switch zur Daisy-Chain Verkabelung
- 2x 1 GBit/s Full Duplex Betrieb
- OPC UA Diagnose und Moduldiagnose zur Laufzeit über OPC UA Client

## 2 Technische Beschreibung

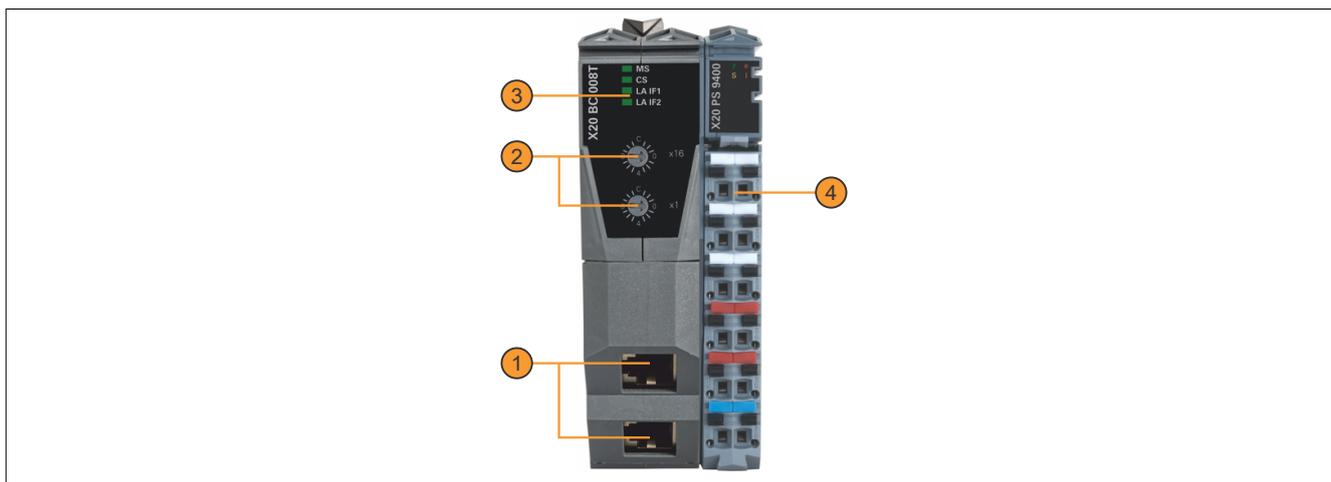
### 2.1 Technische Daten

<b>Bestellnummer</b>	<b>X20BC008T</b>
<b>Kurzbeschreibung</b>	
Bus Controller	OPC UA FX
<b>Allgemeines</b>	
B&R ID-Code	0xF629
Statusanzeigen	Modulstatus, Busfunktion
Diagnose	
Modulstatus	Ja, per Status-LED und SW-Status
Busfunktion	Ja, per Status-LED und SW-Status
Leistungsaufnahme	
Bus	3,5 W
Zusätzliche Verlustleistung durch Aktoren (ohmsch) [W]	-
Zulassungen	
CE	Ja
UKCA	Ja
EAC	Ja
<b>Schnittstellen</b>	
Feldbus	OPC UA Field Exchange (FX)
Ausführung	2x RJ45 geschirmt (Switch)
Leitungslänge	max. 100 m zwischen 2 Stationen (Segmentlänge)
Übertragungsrate	1 GBit/s
Übertragung	
Physik	100BASE-TX/1000BASE-T
Halbduplex	Nein
Voll duplex	Ja
Autonegotiation	Ja
Auto-MDI/MDIX	Ja
Min. Zykluszeit <sup>1)</sup>	
Feldbus	400 µs
X2X Link	400 µs
Synchronisation zw. Bussen möglich	Ja
<b>Elektrische Eigenschaften</b>	
Potenzialtrennung	OPC UA FX zu Bus und I/O getrennt
<b>Einsatzbedingungen</b>	
Einbaulage	
waagrecht	Ja
senkrecht	Ja
Aufstellungshöhe über NN (Meeresspiegel)	
0 bis 2000 m	Keine Einschränkung
>2000 m	Reduktion der Umgebungstemperatur um 0,5°C pro 100 m
Schutzart nach EN 60529	IP20
<b>Umgebungsbedingungen</b>	
Temperatur	
Betrieb	
waagrechte Einbaulage	-25 bis 45 °C
senkrechte Einbaulage	-25 bis 40 °C
Derating	Siehe Abschnitt "Derating"
Lagerung	-40 bis 85°C
Transport	-40 bis 85°C
Luftfeuchtigkeit	
Betrieb	5 bis 95%, nicht kondensierend
Lagerung	5 bis 95%, nicht kondensierend
Transport	5 bis 95%, nicht kondensierend
<b>Mechanische Eigenschaften</b>	
Anmerkung	Feldklemme 1x X20TB12 gesondert bestellen Einspeisemodul 1x X20PS9400 oder X20PS9402 gesondert bestellen Busbasis 1x X20BB80X gesondert bestellen
Rastermaß <sup>2)</sup>	37,5 <sup>+0,2</sup> mm

Tabelle 2: X20BC008T - Technische Daten

- Die minimale Zykluszeit gibt an, bis zu welcher Zeit der Buszyklus heruntergefahren werden kann, ohne dass Kommunikationsfehler auftreten.
- Das Rastermaß bezieht sich auf die Breite der Busbasis X20BB80X. Zum Bus Controller wird immer auch ein Einspeisemodul X20PS9400 oder X20PS9402 benötigt.

## 2.2 Bedien- und Anschlusselemente



1	OPC UA FX Anschluss mit 2 x RJ45 zur einfachen Verdrahtung	2	Nummernschalter
3	LED-Statusanzeige	4	Feldklemme für Bus Controller und I/O-Einspeisung

### 2.2.1 Status-LEDs

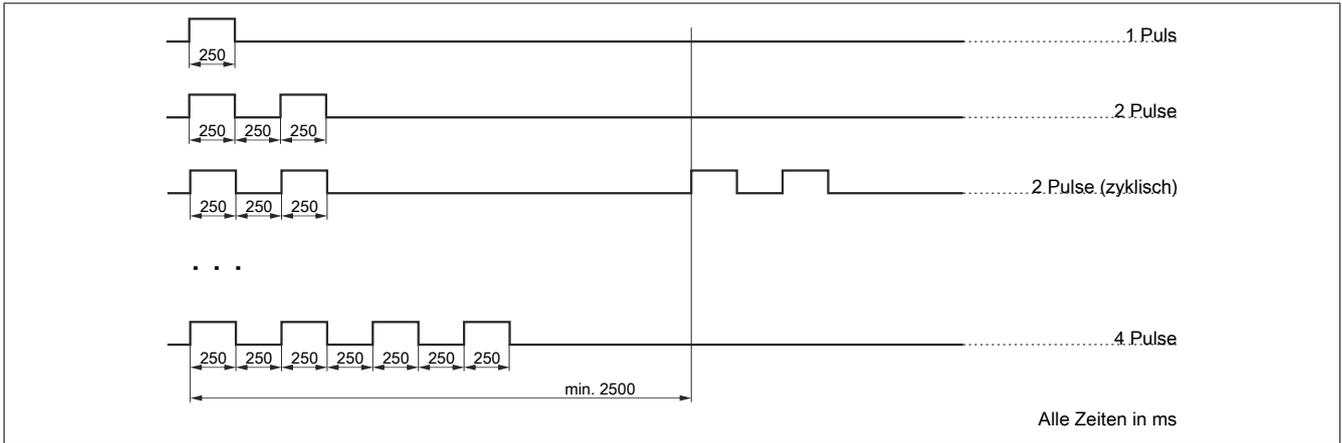
In der folgenden Tabelle sind die Status-LEDs des Bus Controllers beschrieben. Die genauen Blinkzeiten zeigt das Zeitdiagramm im nächsten Abschnitt.

Direkt nach dem Einschalten blitzen die LEDs rot auf. Dies ist keine Fehlermeldung.

Abbildung	LED	Farbe	Status	Beschreibung
	MS <sup>1)</sup>	-	Aus	Modul nicht versorgt oder Modus RESET <sup>2)</sup>
		Grün	2 Pulse	Firmware-Update
		Ein	Modul OK	
		Rot	1 Puls	Modus RESET: Neustart
		2 Pulse	Modus RESET: Konfiguration löschen	
		3 Pulse	Modus RESET: Sicherheit-Konfiguration löschen	
		4 Pulse	Modus RESET: Zurücksetzen auf Werkseinstellungen	
		Ein	Fehlerzustand	
		Grün + Rot	Ein	Modus RESET: Bestätigung des Löschvorgangs
		CS <sup>3)</sup>	Grün	1 Puls
	2 Pulse			Warten auf PTP-Synchronisation
	3 Pulse			Warten auf NTP-Synchronisation
	Ein			Netzwerk OK
	Rot		1 Puls	Zeitüberschreitung IP-Konfiguration <sup>4)</sup>
			2 Pulse	Zeitüberschreitung PTP-Synchronisation <sup>5)</sup>
			3 Pulse	Zeitüberschreitung NTP-Synchronisation
			4 Pulse	Fehler PTP-Status <sup>6)</sup>
	LA IFx	Grün	Aus	Kein Link zur Gegenstelle
			Ein	Der Link zur Gegenstelle ist aufgebaut
			Flackernd	Der Link zur Gegenstelle ist aufgebaut. Die LED flackert, wenn Ethernet Aktivität vorhanden ist.

- 1) Modul-Status "MS": Diese LED ist eine grün/rote Dual-LED.
- 2) Siehe "Nummernschalter" auf Seite 4.
- 3) LAN-Status "LS": Diese LED ist eine grün/rote Dual-LED.  
Die LED wechselt vom grün gepulsten Zustand in den rot gepulsten Zustand, wenn der aktuelle "Warten auf"-Status länger als 15 s ansteht. Bei Statuswechsel wird diese Zeit zurückgesetzt.
- 4) Dem Bus Controller wurde noch keine IP-Adresse zugewiesen.
- 5) Der Bus Controller ist noch nicht über PTP synchronisiert. Mögliche Ursachen:
  - Keine Verbindung zu einem PTP-Grandmaster
  - Der Synchronisationsoffset zum PTP-Grandmaster ist außerhalb der Vorgabe ( $\text{abs}(\text{OffsetFromMaster}) > \text{SyncOffsetNs}$ ).
  - PTP-Konfigurationsfehler
- 6) Mögliche Ursachen:
  - Der Bus Controller wurde als PTP-Grandmaster konfiguriert ( $\text{Priority1} < 128$ ), ist jedoch PTP-Slave.
  - Der Bus Controller wurde als PTP-Slave konfiguriert ( $\text{SlaveOnly} = \text{true}$ ), ist jedoch PTP-Grandmaster.

**Status-LEDs - Blinkzeiten**



**2.2.2 Nummernschalter**



Mittels der beiden Nummernschalter kann nur der Resetmodus aktiviert werden.

Schalterstellung	Beschreibung
0x00 - 0xFE	Resetmodus nicht aktiv
0xFF	Resetmodus aktiviert

**Reset während Hochlauf**

Anzeige des Hochlaufs: LED "MS" leuchtet noch nicht dauerhaft grün oder rot.

**Information:**

Während des Hochlaufs ist die Einstellung des Resetmodus nicht zulässig.

**Reset während Betrieb**

Während des Betriebs ist die ausgelöste Funktion von der Einstelldauer des Resetmodus abhängig.

Funktion	Einstelldauer	LED-Anzeige <sup>1)</sup>	Bestätigung
Temporäre IP-Adresse setzen <sup>2)</sup>	1 s	LED "MS": Aus	-
Neustart	5 s	LED "MS": Nach 5 Sekunden 1 Puls	-
Konfiguration löschen	10 s	LED "MS": Nach 10 Sekunden 2 Pulse	Werden die Nummernschalter innerhalb von 5 s erneut betätigt, wird die Aktion ausgeführt und anschließend der Bus Controller neu gestartet.
Sicherheitskonfiguration löschen	15 s	LED "MS": Nach 15 Sekunden 3 Pulse	
Zurücksetzen auf Werkseinstellungen	20 s	LED "MS": Nach 20 Sekunden 4 Pulse	

1) Siehe "Status-LEDs" auf Seite .

2) Temporäre IP-Adresse 192.168.1.1; siehe "IP-Adresse einstellen" auf Seite 6.

**Beispiel "Resetmodus - Temporäre IP-Adresse setzen"**

- 1) Nummernschalter auf 0xFF stellen
- 2) Nach 1 s erlischt die LED "MS"
- 3) Nummernschalter innerhalb 5 s ungleich 0xFF stellen, bevor die LED "MS" mit 1 Puls rot blinkt (Modus: Neustart)

**Information:**

Die temporäre IP-Adresse wird nur für den aktuellen Bootvorgang gesetzt und ist nach einem Neustart des Geräts nicht mehr vorhanden. Sie ermöglicht eine initiale Verbindung zum Gerät, um eine statische IP-Adresse zu konfigurieren.

### Beispiel "Resetmodus - Neustart"

- 1) Nummernschalter auf 0xFF stellen
- 2) Nach 1 s erlischt die LED "MS" (Modus: Temporäre IP-Adresse); nach 5 s blinkt die LED "MS" mit 1 Puls rot (Modus: Neustart)
- 3) Nummernschalter innerhalb 5 s ungleich 0xFF stellen, bevor die LED "MS" mit 2 Pulsen rot blinkt (Modus: Konfiguration löschen)
- 4) Gerät wird neu gestartet

### Beispiel "Resetmodus - Konfiguration löschen"

- 1) Nummernschalter auf 0xFF stellen
- 2) Nach 1s erlischt die LED "MS" (Modus: Temporäre IP-Adresse); nach 5s blinkt die LED "MS" mit 1 Puls rot und nach 10 s mit 2 Pulsen rot (Modus: Konfiguration löschen)
- 3) Nummernschalter innerhalb 5 s ungleich 0xFF verstellen, bevor die LED "MS" mit 3 Pulsen rot blinkt (Modus: Sicherheitskonfiguration löschen)
- 4) LED "MS" leuchtet für 5 s Grün und Rot (Modus: Bestätigung des Löschvorgangs). Innerhalb dieser Zeit Nummernschalter zum Bestätigen erneut auf 0xFF drehen und danach wieder ungleich 0xFF drehen. Erfolgt keine Bestätigung, dann wird die Konfiguration nicht gelöscht
- 5) Konfiguration wird gelöscht und Gerät neu gestartet

### Konfiguration löschen

Folgende Einstellungen werden gelöscht:

- Netzwerkkonfiguration
- X2X-Konfiguration
- Zeitsynchronisationskonfiguration
- TSN-Konfiguration

### Sicherheitskonfiguration löschen

Folgende Einstellungen werden gelöscht:

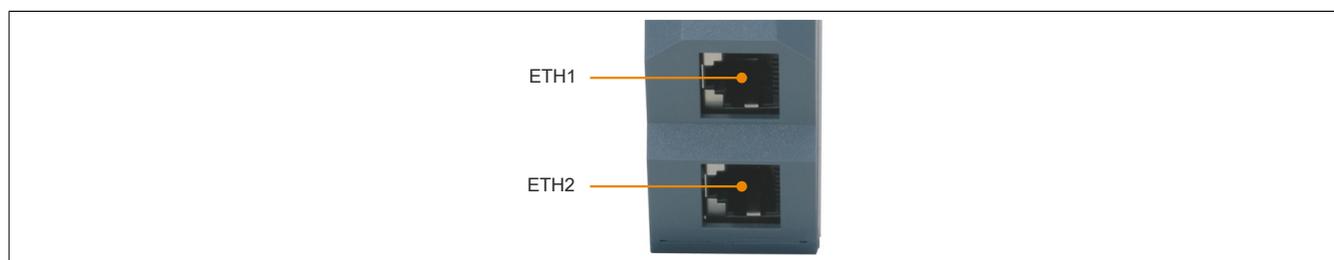
- Benutzer / Passwörter
- OPC UA Zertifikate
- Netconf Zertifikate / SSH Keys

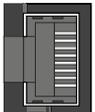
### Zurücksetzen auf Werkseinstellungen

Entspricht "Konfiguration löschen" und "Sicherheitskonfiguration löschen".

### 2.2.3 Ethernet-Schnittstelle

Hinweise für die Verkabelung von X20 Modulen mit Ethernet-Schnittstelle sind im X20 Anwenderhandbuch, Abschnitt "Mechanische und elektrische Konfiguration - Verkabelungsvorschrift für X20 Module mit Ethernet Kabel" zu finden.



Schnittstelle	Anschlussbelegung		
	Pin	Ethernet	
 RJ45 geschirmt	1	RXD	Empfange (Receive) Daten
	2	RXD\	Empfange (Receive) Daten\
	3	TXD	Sende (Transmit) Daten
	4	Termination	
	5	Termination	
	6	TXD\	Sende (Transmit) Daten\
	7	Termination	
	8	Termination	

## 2.3 IP-Adresse einstellen

Je nach verwendetem Einsatzgebiet kann eine IP-Adresse dem Bus Controller auf verschiedene Arten zugewiesen werden.

- Automatische Zuweisung per DHCP-Server  
Standardmäßig ist der Bus Controller für eine automatische IP-Adresszuweisung per DHCP-Server konfiguriert. In Maschinennetzwerken mit einer B&R-Steuerung wird die DHCP-Server-Funktion von der Automation Runtime bereitgestellt.  
PCs oder Laptops mit Desktop-Betriebssystemen, wie z. B. Windows oder Linux, bieten jedoch normalerweise keinen DHCP-Server an.
- Einstellen der temporären IP-Adresse (192.168.1.1) durch Betätigen der Nummernschalter. (Siehe Abschnitt "Nummernschalter" auf Seite 4)
- Konfiguration per OPC UA-Server (Siehe Abschnitt "Verbindungsaufbau" auf Seite 7)

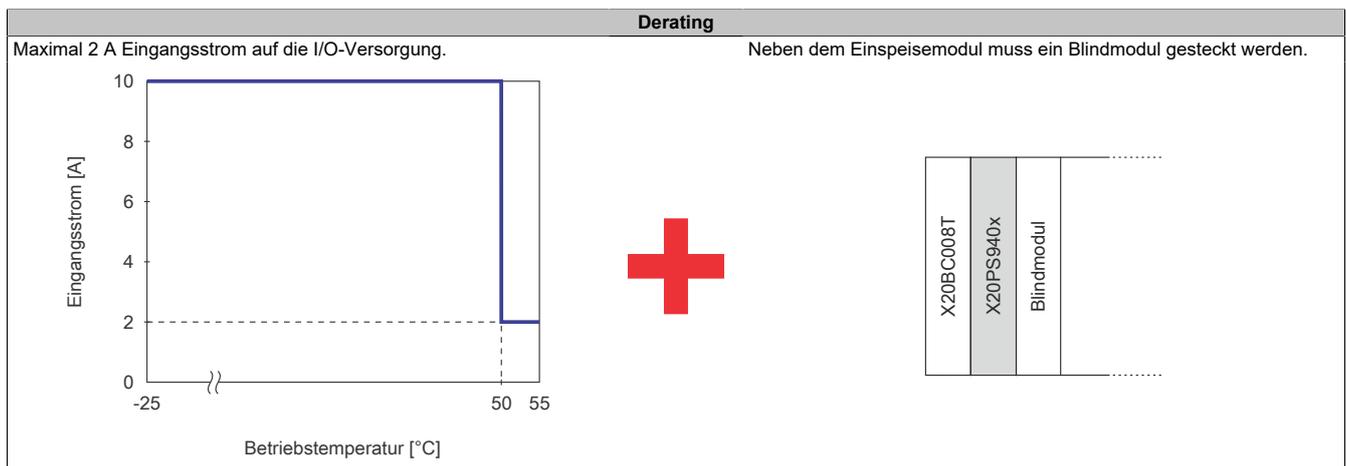
## 2.4 Derating

### Waagrechte Einbaulage

Im Temperaturbereich von -25 bis 50°C ist kein Derating erforderlich. Bei Temperaturen über 50°C sind folgende 2 Deratings zu beachten.

#### Information:

Es müssen immer beide Deratings durchgeführt werden!

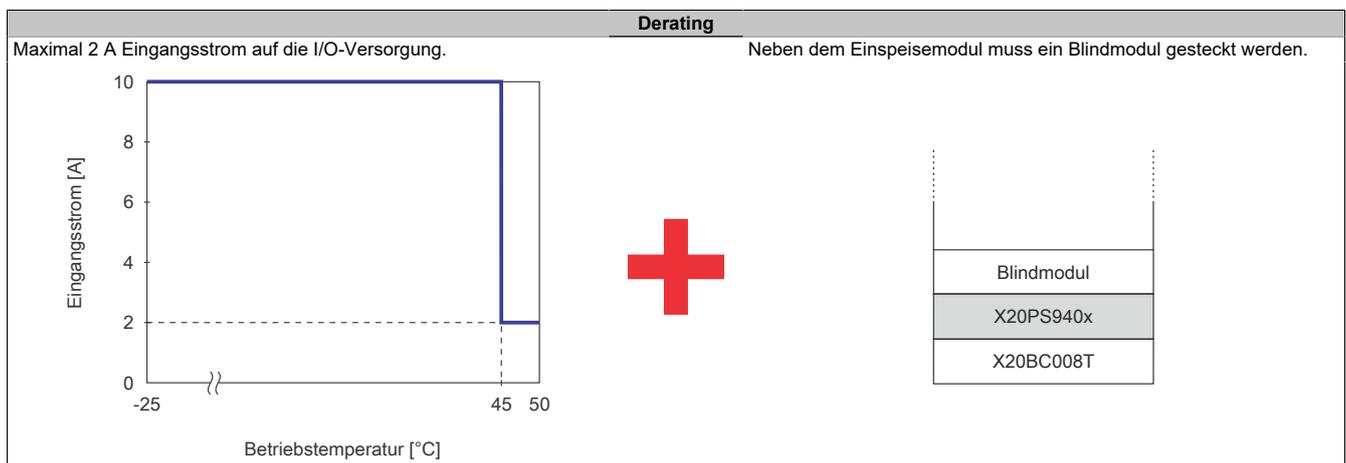


### Senkrechte Einbaulage

Im Temperaturbereich von -25 bis 40°C ist kein Derating erforderlich. Bei Temperaturen über 45°C sind folgende 2 Deratings zu beachten.

#### Information:

Es müssen immer beide Deratings durchgeführt werden!



## 3 Erste Schritte

Der Bus Controller wird mit Werkseinstellungen ausgeliefert. Das bedeutet, dass weder Gerätefunktionalität noch etwaige Sicherheitseinstellungen konfiguriert sind. Um die Inbetriebnahme sicher zu gestalten, soll dafür gesorgt werden, dass der Bus Controller vorerst nur in einer sicheren Umgebung benutzt wird. Sichere Umgebungen sind z. B. von Unternehmensnetzwerk getrennte Netzwerke oder eine direkte Verbindung mit dem zur Konfiguration benutzten PC. Nach erfolgter Sicherheitskonfiguration kann der Bus Controller auch in einer nicht sicheren Umgebung sicher betrieben werden.

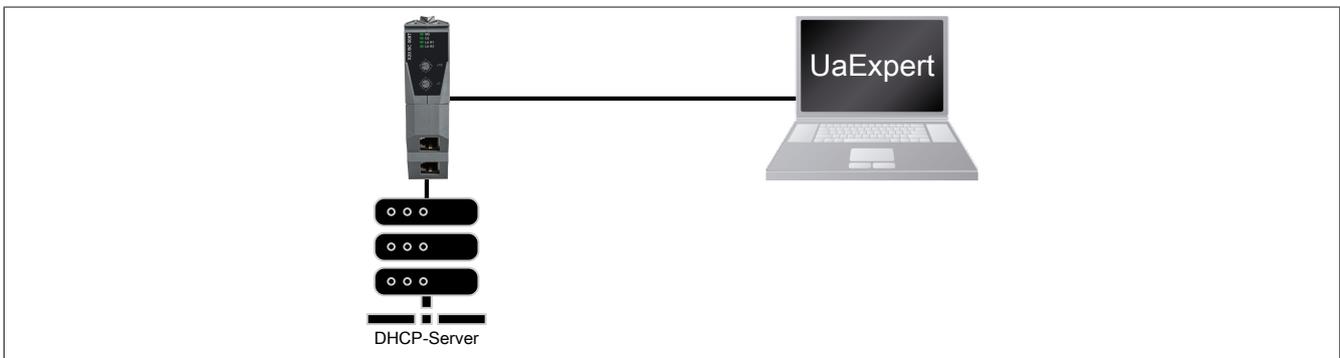
### 3.1 Vorbereitung

In den folgenden Beispielen wird die OPC UA Client-Software "UaExpert" für die Konfiguration verwendet. Sie kann aber auch mit anderen, vergleichbaren Tools durchgeführt werden.

Dabei sollte folgende Mindestversion verwendet werden:

- UaExpert ab Version 1.6  
Download: <https://www.unified-automation.com>

Zu Beginn kann der folgende Aufbau für eine Erstkonfiguration verwendet werden. Dieser besteht aus einem PC mit UaExpert-Software, einem direkt angeschlossenen Bus Controller und einem DHCP-Server. Der DHCP-Server kann dabei auch Teil des PCs sein.



### 3.2 Verbindungsaufbau

#### Information:

Um Problem beim Verbindungsaufbau zu vermeiden, siehe auch Abschnitt 5.6 "Integration im IT-Netzwerk".

In der Werkseinstellung wird am Bus Controller ein DHCP-Client gestartet und ein Hostname abhängig von Produktkennung und MAC-Adresse generiert. Ein im Netzwerk vorhandener DHCP-Server kann dadurch dem Bus Controller eine IP-Adresse zuweisen. Zusätzlich ist am Bus Controller Multicast-DNS (mDNS) aktiviert.

In der Werkseinstellung werden folgende Netzwerkeinstellungen vom DHCP-Server übernommen:

- IP-Adresse
- Subnetzmaske
- Gateway
- Hostname
- Domäne
- DNS-Server
- NTP-Server

Um die Werkseinstellungen zu ändern (siehe Abschnitt 3.5 "Allgemeine Netzwerkeinstellungen über OPC UA"), muss zuerst einer der folgenden, werkseitig vorhandenen Mechanismen für die erste Verbindung verwendet werden.

### 3.2.1 Verbindungsaufbau per Hostname

#### 3.2.1.1 Hostnamen ermitteln

Für den Verbindungsaufbau muss zuerst der Hostname des Bus Controllers bekannt sein. In den Werkseinstellungen wird dieser aus der Produktkennung und der Bus Controller-MAC-Adresse generiert und hat folgendes Format:

x20bc008t-[MAC-Adresse]

#### Information:

**Nach einer Änderung des Hostnamens ist der Default-Hostname aus Produktkennung und Bus Controller-MAC-Adresse nicht mehr gültig.**

#### Beispiel

Für einen Bus Controller mit MAC-Adresse 00:60:65:00:22:01 ergibt sich folgender Hostname:

x20bc008t-006065002201

Um den Hostnamen zu ermitteln gibt es folgende Möglichkeiten:

##### 3.2.1.1.1 Hostname mit Gehäuseaufdruck ermitteln

Die Bus Controller-MAC-Adresse ist, zusammen mit den MAC-Adressen der Ports, am Gehäuse aufgedruckt.

##### 3.2.1.1.2 Hostname mit LLDP und Direktverbindung ermitteln

Alternativ kann der Hostname über eine Netzwerkverbindung mit LLDP ermittelt werden. Der Bus Controller veröffentlicht die MAC-Adresse des Endpoints im Netzwerk über das "Link Layer Discovery Protokoll (LLDP)" mit der Bezeichnung "ChassisID" an direkte Nachbargeräte. Diese lässt sich z. B. von einem PC mit Linux und direktem Geräteanschluss mithilfe von LLDP ermitteln:

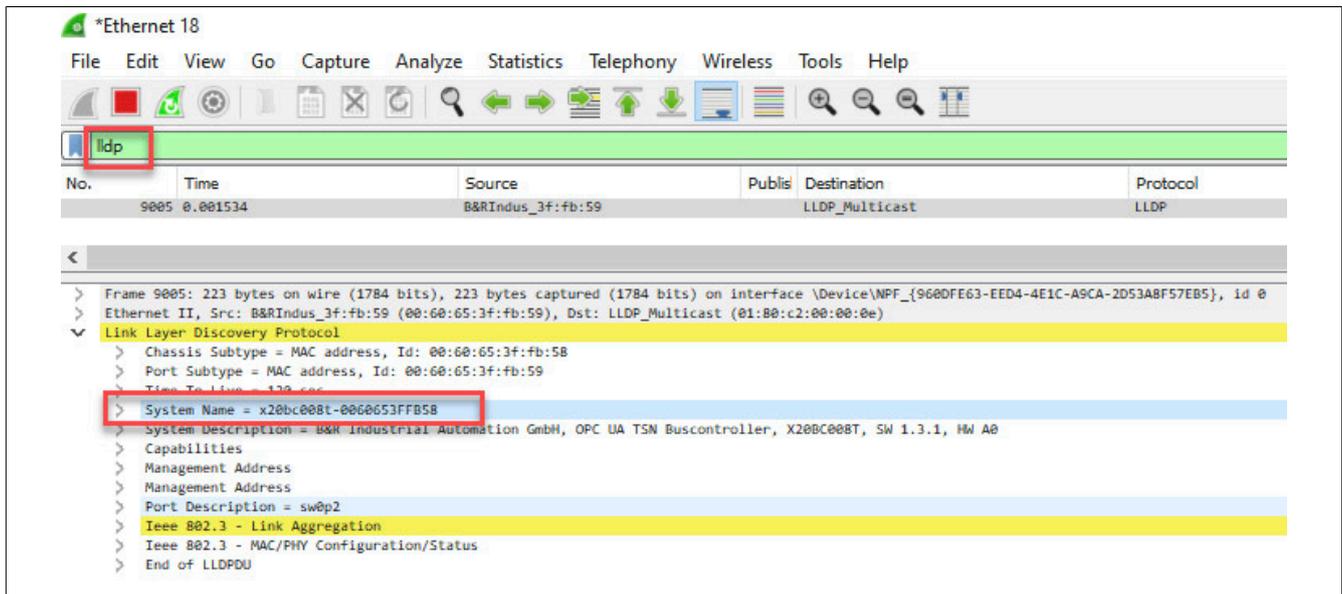


#### Beispiel

```
$ lldpctl
-----
Interface:   enx9cebe8ae5553, via: LLDP, RID: 42, Time: 0 day, 01:03:00
Chassis:
  ChassisID:   mac 00:60:65:00:22:01
  SysName:     x20bc008t-006065002201.home
  SysDescr:    B&R Industrial Automation GmbH, 802.1Q OPC UA FX Buscontroller, X20BC008T,
               SW 1.0.0, HW C0
  MgmtIP:      192.168.0.128
  MgmtIP:      2a02:810d:6e3f:e9a0:260:65ff:fe00:2201
  Capability:   Bridge, on
  Capability:   Router, off
  Capability:   Wlan, off
  Capability:   Station, off
Port:
  PortID:      mac 00:60:65:00:22:03
  PortDescr:   sw0p3
  PMD autoneg: supported: yes, enabled: yes
  Adv:         100Base-TX, HD: no, FD: yes
  Adv:         1000Base-T, HD: no, FD: yes
  MAU oper type: 100BaseTXFD - 2 pair category 5 UTP, full duplex mode
-----
```

### 3.2.1.1.3 Hostname mit LLDP und Direktverbindung mittels Wireshark ermitteln

Steht am PC kein LLDP zur Verfügung, kann auch eine Netzwerkaufzeichnung z. B. mittels Wireshark durchgeführt werden. Diese enthalten den Hostnamen. Dazu in Wireshark den Filter "lldp" anwenden und in den Details des LLDP-Datagramms den "System Name" auslesen → dieser entspricht dem Hostnamen.

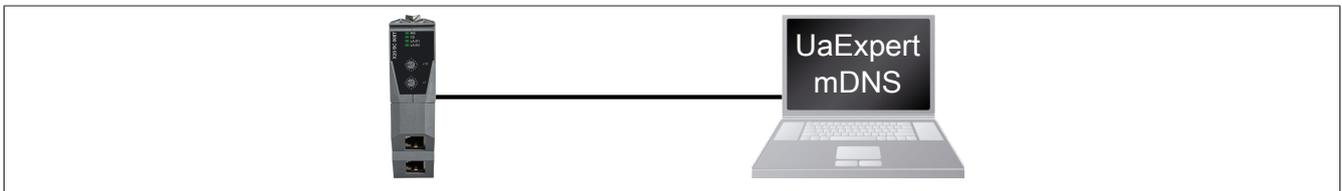


### 3.2.1.2 Hostnamen auflösen

Nachdem der Hostname ermittelt wurde, muss er durch die Netzwerk-Infrastruktur in eine IP-Adresse aufgelöst werden. Dafür gibt es folgende Möglichkeiten:

#### 3.2.1.2.1 Hostname-Auflösung per mDNS

Nachdem der Hostname bekannt ist, kann der Bus Controller vom PC aus über diesen Namen angesprochen werden. Die Verbindung erfolgt in diesem Fall über den Hostnamen und der ".local"-mDNS-Domäne. Die IP-Adresse muss bei dieser Möglichkeit nicht bekannt sein.



Folgende "Endpoint-URL" kann im UaExpert für den Verbindungsaufbau verwendet werden (siehe [3.4 "Anlegen des initialen Benutzers"](#)):

```
opc.tcp://<Produktkennung>-<MAC-Adresse>.local:4840
```

Bzw. für dieses Beispiel:

```
opc.tcp://x20bc008t-006065002201.local:4840
```

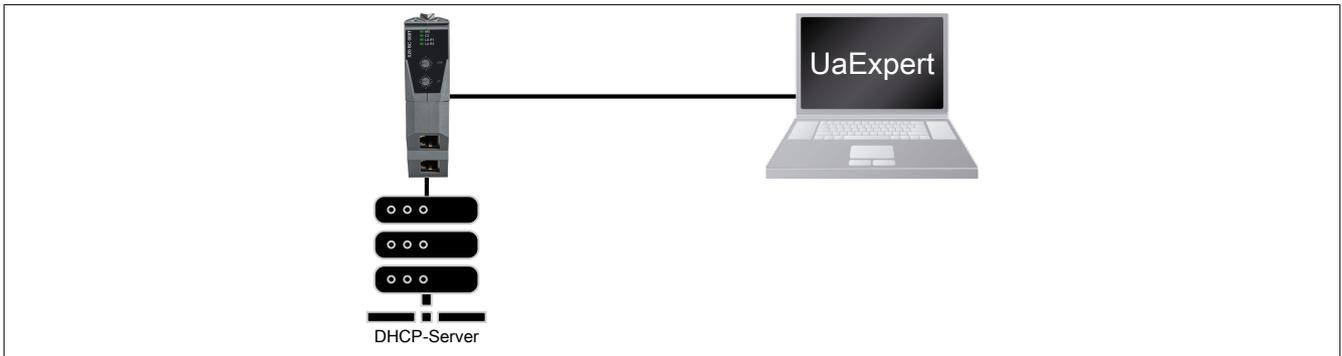
### Information:

Der OPC UA Server am Bus Controller erwartet eingehende Verbindungen auf Port 4840.

### 3.2.1.2 Hostname-Auflösung per DNS

In großen Netzwerken mit vielen Teilnehmern oder wenn eine DHCP/DNS-Infrastruktur vorhanden ist und genutzt wird, besteht die Möglichkeit mDNS über das OPC UA Informationsmodell zu deaktivieren.

Die Verbindung erfolgt über den Hostnamen, da eine DHCP/DNS-Infrastruktur existiert. Die IP-Adresse muss bei dieser Möglichkeit nicht bekannt sein.



Folgende "Endpoint-URL" kann im UaExpert für den Verbindungsaufbau verwendet werden (siehe [3.4 "Anlegen des initialen Benutzers"](#)):

```
opc.tcp://<Produktkennung>-<MAC-Adresse>:4840
```

Bzw. für dieses Beispiel:

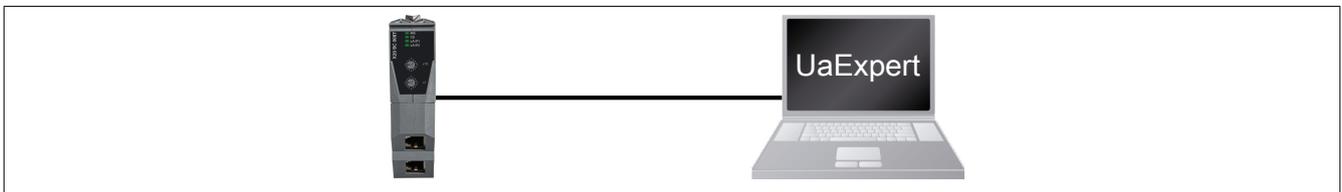
```
opc.tcp://x20bc008t-006065002201:4840
```

### 3.2.2 Verbindungsaufbau per IP-Adresse

Je nach vorhandener Infrastruktur kann die Verbindung durch eine statische oder dynamischen IP-Adresse erfolgen.

#### 3.2.2.1 Statische IP-Adresse

Für diese Methode wird kein DHCP-Server benötigt. Mit Hilfe des [Nummerschalters](#) wird die IPv4-Adresse für den aktuellen Bootvorgang auf den Wert "192.168.1.1" gesetzt.



Folgende "Endpoint-URL" kann im UaExpert für den Verbindungsaufbau verwendet werden (siehe [3.4 "Anlegen des initialen Benutzers"](#)):

```
opc.tcp://192.168.1.1:4840
```

Falls die IPv4-Adresse bereits konfiguriert und bekannt ist, ist der Resetvorgang nicht notwendig. In diesem Fall lautet die "Endpoint-URL" im UaExpert:

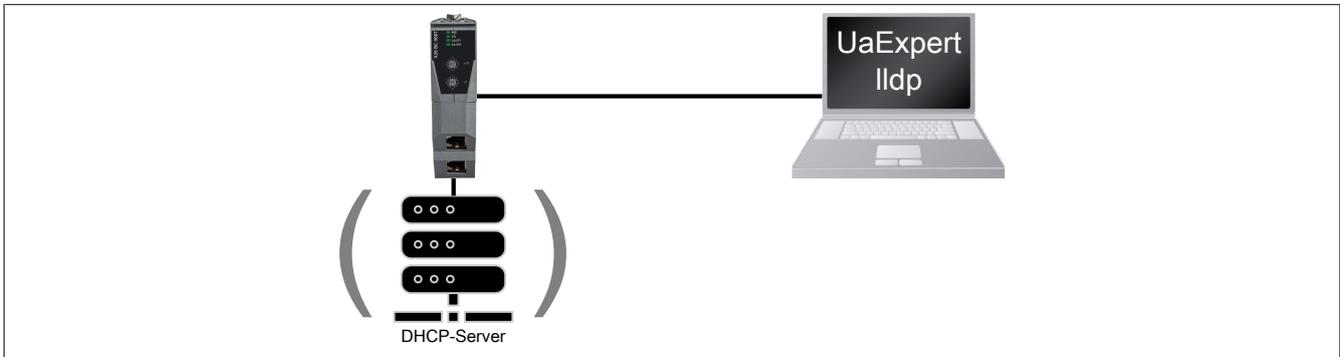
```
opc.tcp://<Bekannte IP-Adresse>:4840
```

### 3.2.2.2 Dynamische oder unbekannte IP-Adresse

Die Zuweisung einer IP-Adresse an den Bus Controller kann auf mehrere Arten erfolgen:

- Durch den DHCP-Server
- Bekommt der Bus Controller keine IP-Adresse per DHCP zugewiesen, wird vom Bus Controller automatisch eine zufällige IPv4 Link-Local (IPv4LL) Adresse generiert

Diese zugewiesene IPv4-Adresse lässt sich per LLDP (siehe Abschnitt [Hostname mit LLDP und Direktverbindung ermitteln](#)) mit der Bezeichnung "MgmtIP" ermitteln.

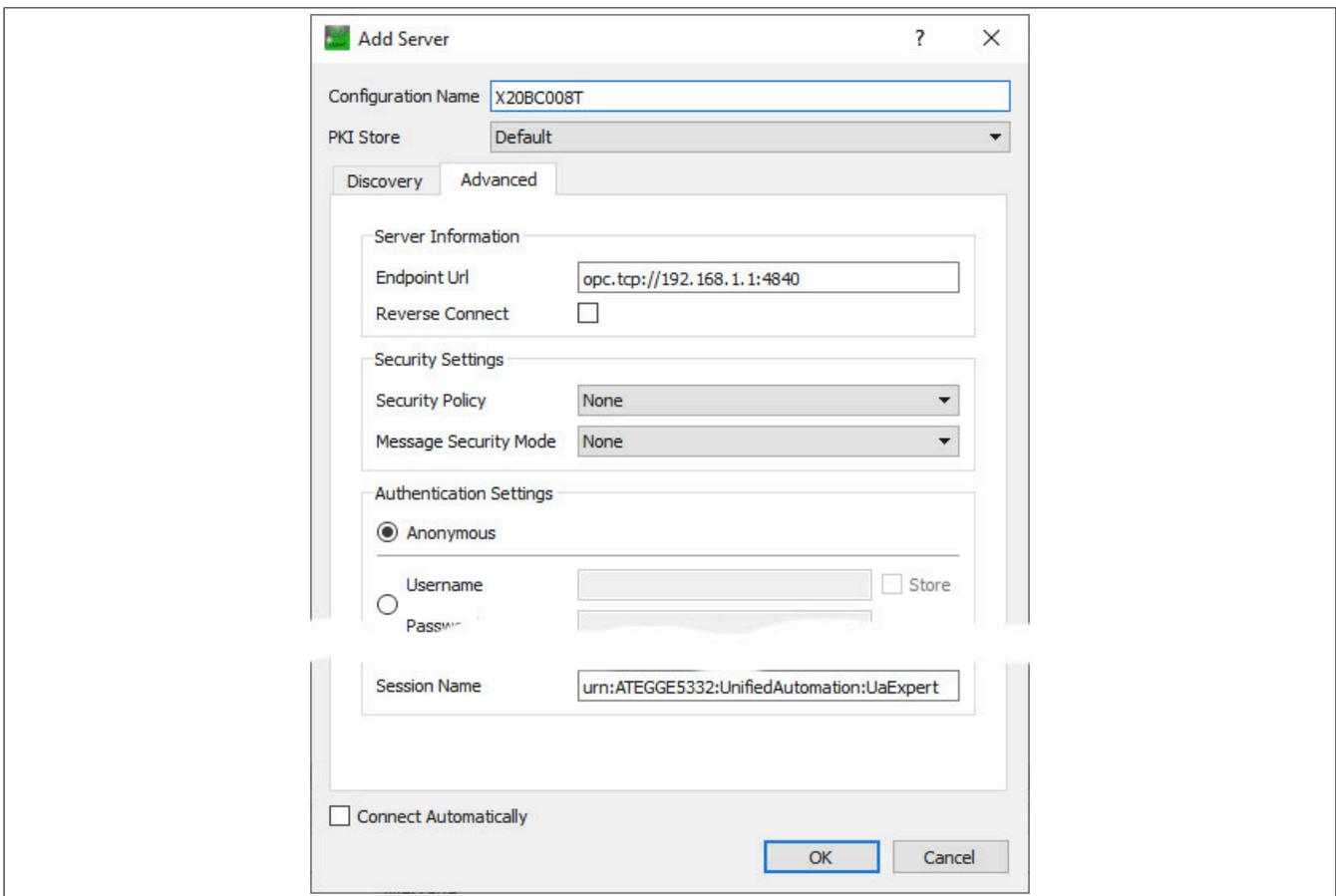


Folgende "Endpoint-URL" kann im UaExpert für den Verbindungsaufbau verwendet werden (siehe [3.4 "Anlegen des initialen Benutzers"](#)):

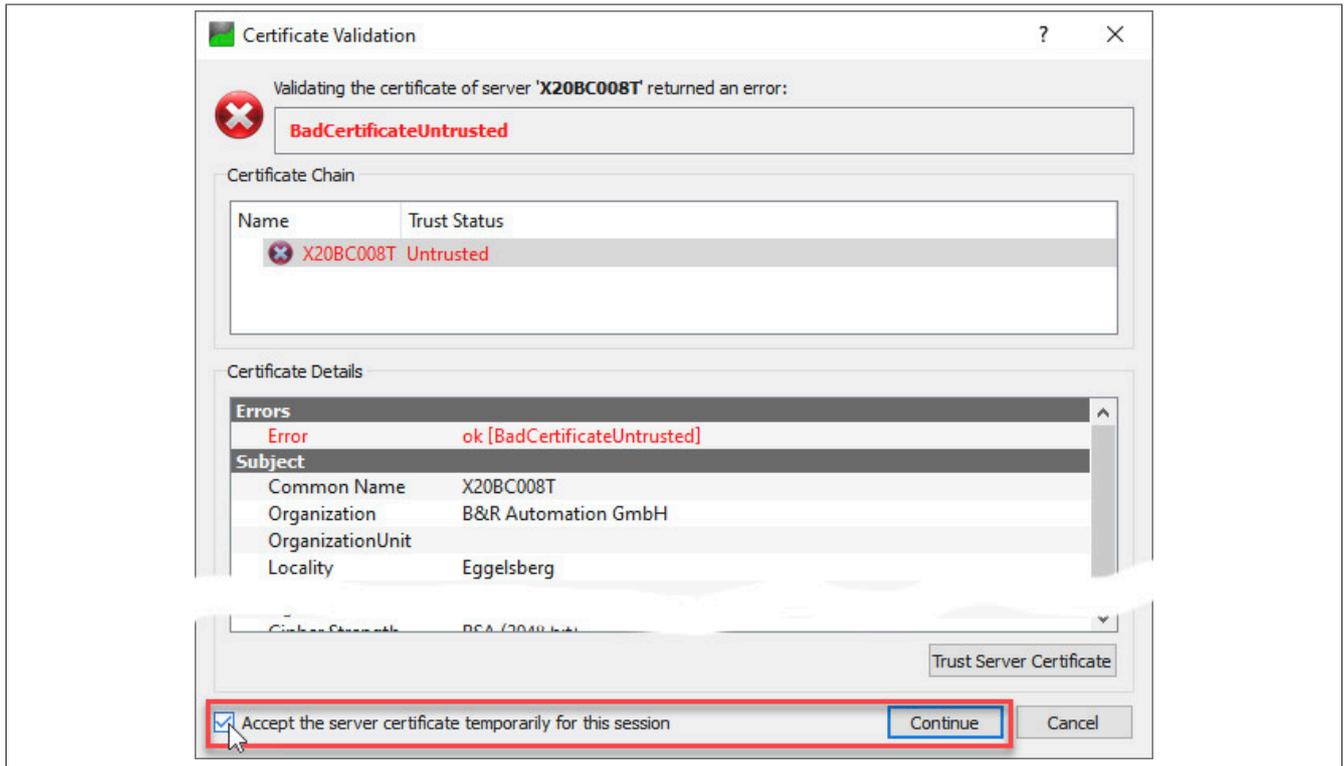
`opc.tcp://<Ermittelte IP-Adresse>:4840`

### 3.3 Mit OPC UA Client verbinden

- Für die erste OPC UA Verbindung ist die Einstellung *Anonymous* zu verwenden. Zusätzlich sollte eine angemessene Security-Policy wie *Basic256SHA256* ausgewählt werden, da sensible Daten übertragen werden.



- Der Bus Controller ist initial noch nicht in eine Public-Key-Infrastruktur (PKI) eingebunden und hat daher lediglich ein selbst erzeugtes Zertifikat. Dieses Zertifikat ist korrekt, der Client kann dessen Herkunft aber nicht verifizieren und warnt daher. In einer vertrauenswürdigen Umgebung ist es aber sicher, dieses Zertifikat zu akzeptieren.



- Durch Auswahl von "Accept the server certificate temporarily for this session" und einen Klick auf *Continue* wird das Zertifikat akzeptiert.

### Information:

In einer nicht vertrauenswürdigen Umgebung kann durch das Akzeptieren eines solchen selbst erstellten Zertifikats ein gewisses Risiko entstehen. Ein Angreifer könnte sich als "Man-in-the-Middle" in die Kommunikation einklinken und den Datenverkehr trotz Verschlüsselung mitlesen und verfälschen.

## 3.4 Anlegen des initialen Benutzers

### Information:

Es muss zwingend ein Benutzer angelegt werden, ansonsten kann keine weitere Konfiguration durchgeführt werden.

### Benutzer anlegen

Der Bus Controller ist aktuell noch im Kommissionierungsmodus und erlaubt dem anonymen Client nur den Aufruf weniger Methoden. Diese enthalten das Anlegen des ersten Benutzers, das Setzen des Passworts und die Zuordnung zur Rolle *SecurityAdmin*.

- Als erster Schritt wird der Benutzer angelegt, der für die weitere Konfiguration zuständig ist. Dies geschieht durch Aufruf der Methode *Root/Objects/Server/ServerCapabilities/UserSet/AddUser*. Durch einen Klick auf *Call...* wird der Benutzerdialog angezeigt.



Input Arguments			
Name	Value	DataType	Description
UserName	admin	String	

Output Arguments			
Name	Value	DataType	Description
UserNodeId	0	Numeric	NodeId

Result	

Ein erfolgreicher Aufruf wird unter "Result" angezeigt und die Knoten-ID des angelegten Benutzers zurückgegeben.

Input Arguments			
Name	Value	DataType	Description
UserName	admin	String	

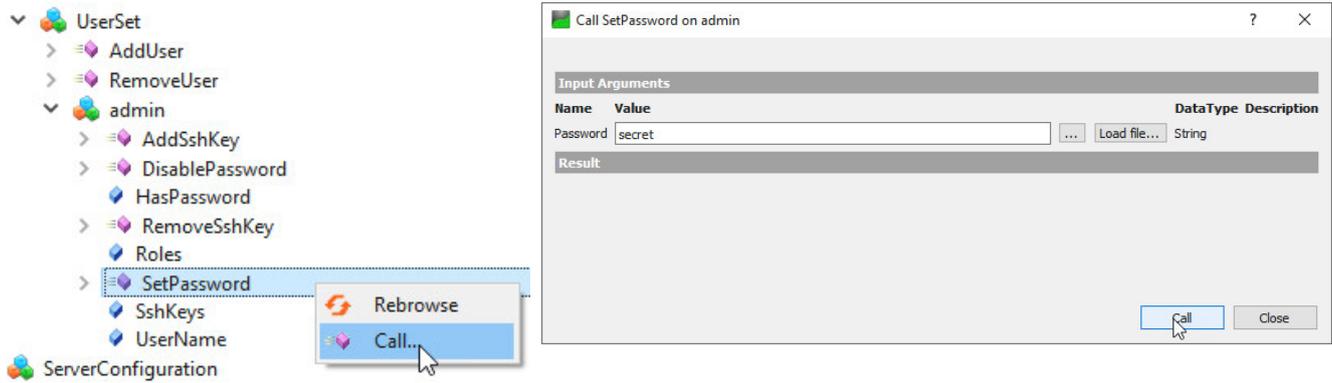
Output Arguments			
Name	Value	DataType	Description
UserNodeId	1	String	NodeId
	User.admin@UserObject		

Result	
Succeeded	

## Passwort zuordnen

- Der Name des neu angelegten Benutzers wird im Informationsmodell angezeigt. Um das Passwort zu konfigurieren, wird die Methode *Root/Objects/Server/ServerCapabilities/UserSet/<NAME>/SetPassword* aufgerufen. Durch einen Klick auf *Call...* wird der Passwortdialog angezeigt.

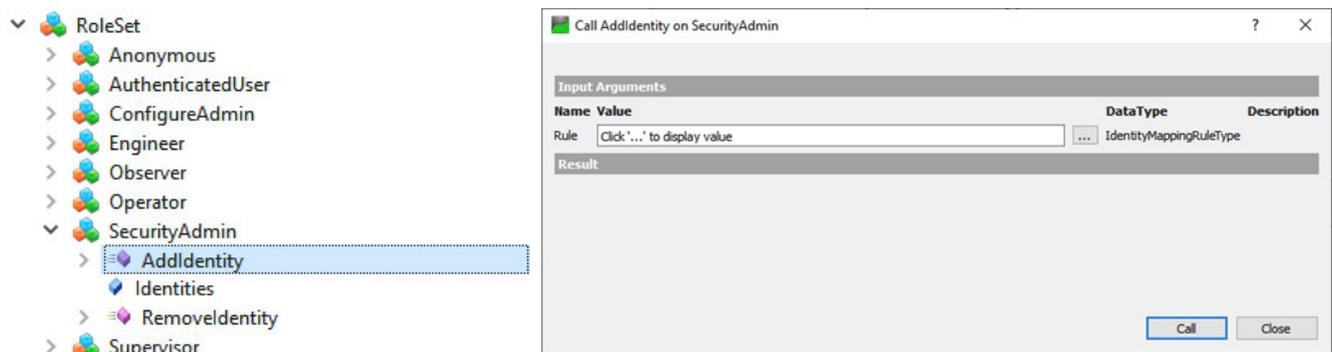


### Information:

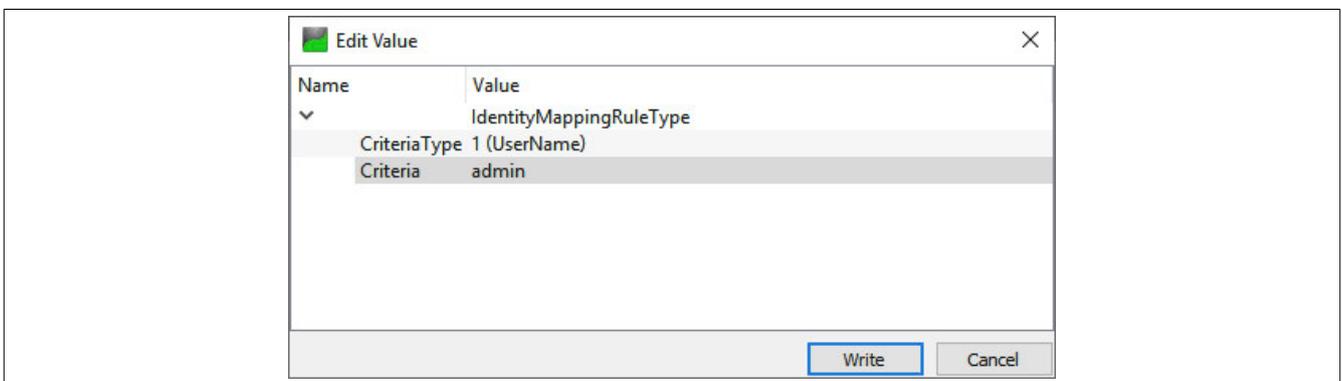
Das Passwort wird verschlüsselt vom Client zum Bus Controller übertragen. Um ungewollte Zugriffe auf den Bus Controller zu vermeiden, ist sicherzustellen, dass das Passwort während der Eingabe nicht von unbefugten Personen gesehen werden kann.

## SecurityAdmin-Rolle zuweisen

- Als nächstes ist dem Benutzer die für die weitere Konfiguration nötigen Berechtigungen als "Security Admin" zuzuweisen. Dazu wird die Methode *Root/Objects/Server/ServerCapabilities/RoleSet/SecurityAdmin/AddIdentity* aufgerufen.



Nach einem Klick auf "..." kann als *CriteriaType* der Eintrag "1 (UserName)" ausgewählt werden. Als "Criteria" wird der Benutzername angegeben.



Mit einem Klick auf *Write* werden die Daten übernommen und der Dialog geschlossen. Anschließend wird der SecurityAdmin-Dialog mit Klick auf *Call* geschlossen und der Benutzer damit als Security Admin angemeldet.

## Rollenzuordnung anzeigen

• Einem Benutzer können mehrere Rollen zugeordnet werden, bzw. mehrere Benutzer können dieselbe Rolle ausüben. Mit Hilfe der beiden Properties *Root/Objects/Server/ServerCapabilities/RoleSet* und *.../ServerCapabilities/UserSet* können diese eingesehen werden.

### Beispiel

Angabe aller Benutzer, welche als SecurityAdmin angemeldet sind.

- ▼ RoleSet
  - > Anonymous
  - > AuthenticatedUser
  - > ConfigureAdmin
  - > Engineer
  - > Observer
  - > Operator
  - ▼ SecurityAdmin
    - > AddIdentity
    - Identities
    - > RemoveIdentity
  - > Supervisor
  - ServerProfileArray
  - SoftwareCertificates

Value	
SourceTimestamp	03-Mar-21 16:01:29.198
SourcePicoseconds	0
ServerTimestamp	03-Mar-21 16:01:29.198
ServerPicoseconds	0
StatusCode	Good (0x00000000)
▼ Value	IdentityMappingRuleType Array[1]
▼ [0]	IdentityMappingRuleType
CriteriaType	1 (UserName)
Criteria	admin

### Beispiel

Angabe aller Rollen, welche dem Benutzer mit Namen "admin" zugeordnet sind.

- ▼ UserSet
  - > AddUser
  - > RemoveUser
  - ▼ admin
    - > AddSshKey
    - > DisablePassword
    - Password
    - > RemoveSshKey
    - Roles
    - > SetPassword
    - SshKeys

Value	
SourceTimestamp	03-Mar-21 16:06:27.937
SourcePicoseconds	0
ServerTimestamp	03-Mar-21 16:06:27.937
ServerPicoseconds	0
StatusCode	Good (0x00000000)
▼ Value	String Array[1]
[0]	SecurityAdmin

## Weitere Zuordnungen

Mit dem Anlegen des ersten Benutzers, dem Setzen des Passworts und der Zuordnung zur Rolle *SecurityAdmin* sind die Möglichkeiten des anonym angemeldeten Clients erschöpft.

• Um weitere Zuordnungen und Einstellungen vornehmen zu können, muss die Verbindung zum Bus Controller getrennt und eine neue, mit Benutzernamen und Passwort authentifizierte Sitzung begonnen werden.

Security Settings

Security Policy: Basic256Sha256

Message Security Mode: Sign & Encrypt

Authentication Settings

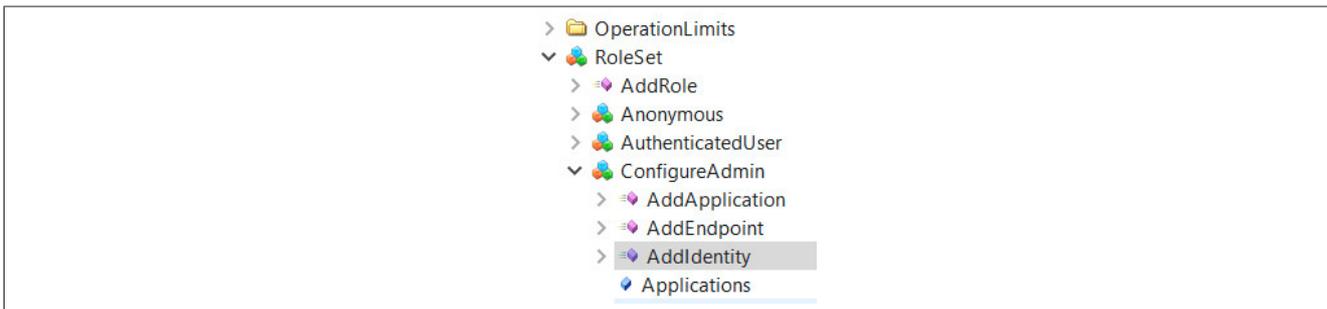
Anonymous

---

Username: admin  Store

    Password: .....

- Damit zusätzliche Einstellungen vorgenommen werden können, muss dem Benutzer zusätzlich die Rolle *ConfigureAdmin* zugewiesen werden. Dazu wird die Methode *Root/Objects/Server/ServerCapabilities/RoleSet/ConfigureAdmin/AddIdentity* aufgerufen und der Name, wie unter [SecurityAdmin-Rolle zuweisen](#) beschrieben, zugeordnet.



### 3.5 Allgemeine Netzwerkeinstellungen über OPC UA

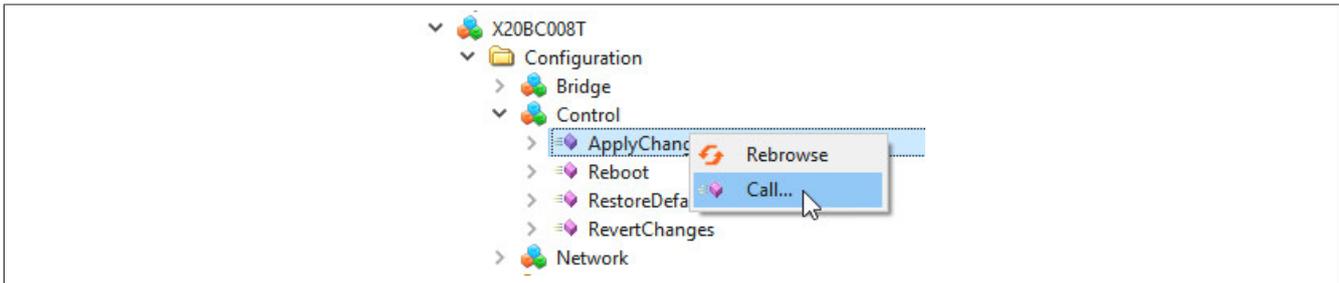
Eine gültige Netzwerkkonfiguration kann über OPC UA durchgeführt werden.

- Dafür werden die verschiedenen Parameter für die Netzwerk-Konfiguration unter *Root/Objects/DeviceSet/X20BC008T/Configuration/Network* aufgerufen und entsprechend beschrieben.

UserRolePermissions	RolePermissionType Array[0]
AccessRestrictions	BadAttributeIdInvalid (0x80350000)
Value	
SourceTimestamp	01.01.1970 02:23:28.219
SourcePicoseconds	0
ServerTimestamp	01.01.1970 02:23:28.219
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	192.168.1.1
DataType	
NamespaceIndex	String
IdentifierType	0
Identifier	Numeric
	12 [String]

Knotenname	Beschreibung
EnableDHCP	Aktiviert beziehungsweise deaktiviert die DHCP-Client-Funktionalität - Bei fehlender IP-Zuweisung durch einen DHCP-Server wird dem Bus Controller eine zufällige Link Local Adresse aus dem Bereich 169.254.0.0/16 zugewiesen. IPv4LL (RFC3927). - Wenn der DHCP-Client aktiviert ist, werden die Parameter <i>Gateway</i> , <i>IP Address</i> , <i>Netmask</i> , sowie <i>Primary DNS</i> und <i>Secondary DNS</i> vom DHCP-Server bezogen.
Gateway	Konfiguration der Default-Gateway IP-Adresse - Wenn der Parameter <i>EnableDHCP</i> gesetzt ist, kann zusätzlich eine Gateway-Adresse vom DHCP-Server übermittelt werden. - Wenn der Parameter <i>Gateway</i> gesetzt ist, wird die manuelle Konfiguration verwendet und die vom DHCP-Server übermittelte Adresse ignoriert.
Hostname	Konfiguration des Hostnamens
IP-Address	Konfiguration einer statischen IP-Adresse - Wenn der Parameter <i>EnableDHCP</i> gesetzt ist, dann wird der Parameter ignoriert und die vom DHCP-Server übermittelte IP-Adresse verwendet.
Primary DNS Secondary DNS	Konfiguration eines primären bzw. sekundären DNS-Servers - Wenn der Parameter <i>EnableDHCP</i> gesetzt ist, können zusätzlich Adressen für DNS-Server vom DHCP-Server übermittelt werden. - Wenn mindestens 1 DNS-Server manuell gesetzt ist, wird die manuelle Konfiguration verwendet und die vom DHCP-Server übermittelten Adressen werden ignoriert.
Netmask	Einstellung der Subnetzmaske - Wenn der Parameter <i>EnableDHCP</i> gesetzt ist, wird dieser Parameter ignoriert und die vom DHCP-Server übermittelte Subnetzmaske verwendet.
EnableMulticastDNS	Aktiviert beziehungsweise deaktiviert Multicast DNS (mDNS) - In der Werkseinstellung ist mDNS aktiviert, um auch bei fehlender Netzwerkinfrastruktur den Bus Controller über den Hostnamen ansprechen zu können.

- Damit die neue Konfigurationsdaten gespeichert werden, muss die Methode *Root/Objects/DeviceSet/X20BC008T/Configuration/Control/ApplyChanges* aufgerufen werden.



## Information:

Die neue Netzwerkkonfiguration wird erst beim Neustart des Bus Controllers übernommen.

### 3.6 Zeitsynchronisation

Für den Betrieb benötigt der Bus Controller Informationen zur aktuellen Uhrzeit. Diese wird vor allem benötigt, damit digitale Zertifikate korrekt verarbeitet werden können und um die Zeitstempel von OPC UA Werten richtig zu setzen.

Im Folgenden wird beschrieben wie die sogenannte "WallClock" konfiguriert werden muss, damit eine Synchronisation über das Network Time Protocol (NTP) erfolgt. Die notwendigen Parameter befinden sich unter *Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WallClock*.

- Damit NTP für die Zeitsynchronisation verwendet wird, muss über Parameter *.../WallClock/TimeSyncProtocol* das Protokoll für die Synchronisation auf NTP eingestellt werden.

UserRolePermissions	RolePermission type Array[UI]
AccessRestrictions	BadAttributeIdInvalid (0x80350000)
Value	
SourceTimestamp	01.01.1970 02:30:43.695
SourcePicoseconds	0
ServerTimestamp	01.01.1970 02:30:43.695
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	0 (NONE)
DataType	0 (NONE)
NamespaceIndex	1 (PTP)
IdentifierType	2 (NTP)
Identifier	3012
ValueRank	-1 (Scalar)
AccessRestrictions	BadAttributeIdInvalid (0x80350000)

- Der nächste Konfigurationsschritt ist von der Art des Netzwerks abhängig, in dem sich der Bus Controller befindet.
  - Wenn im Netzwerk Zeitserver mittels DHCP übermittelt werden, muss kein Zeitserver eingestellt werden, sondern es werden die vom DHCP-Server übermittelten Zeitserver verwendet.
  - Wenn im Netzwerk kein Zeitserver mittels DHCP übermittelt wird, muss im Unterobjekt *NTP* mindestens 1 Zeitserver konfiguriert werden. Hierzu ist im Attribut *Value* des Knotens *TimeServer0x* der Hostnamen oder die IP-Adresse einzutragen.
- Damit die neuen Konfigurationsdaten gespeichert werden, muss die Methode *Root/Objects/DeviceSet/X20BC008T/Configuration/Control/ApplyChanges* aufgerufen werden.

## Information:

Die neue Konfiguration wird erst beim Neustart des Bus Controller übernommen.

### 3.7 Neustart und Reset

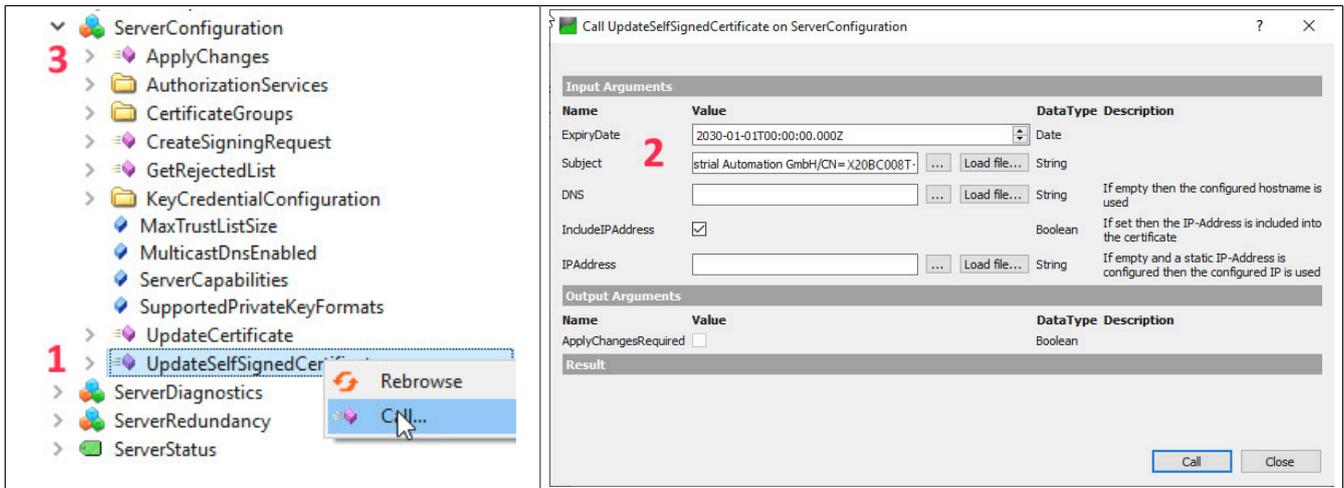
Ein Neustart kann über die Methode *Root/Objects/DeviceSet/X20BC008T/Configuration/Control/Reboot* ausgelöst werden. Vorher mittels der Methode *ApplyChanges* gespeicherte Konfigurationen werden beim Hochfahren des Bus Controllers übernommen und angewendet.

Wurde die Netzwerkkonfiguration geändert, ist nach dem Neustart der Bus Controller nur unter den neuen Einstellungen erreichbar. Bei Verbindungsproblemen sind daher im UaExpert die Verbindungseinstellungen entsprechend der neuen Konfiguration anzupassen.

### 3.8 Aktualisierung des Self-Signed Zertifikats

Der Bus Controller verfügt im Informationsmodell über eine Methode, die verwendet werden kann, um auf einfache Weise ein neues selbstsigniertes Zertifikat zu erzeugen, das notwendige applikationsspezifische Informationen enthält.

- Für die Aktualisierung muss die Methode *UpdateSelfSignedCertificate* durch einen Klick auf *Call* unter *Root/Objects/Server/ServerConfiguration* (1) aufgerufen werden.



Im Methodendialog (2) werden die gewünschten Werte eingegeben. Die Methode verfügt über folgende Argumente:

Argument	Beschreibung																								
<b>Eingangsargumente</b>																									
ExpiryDate	Ablaufdatum, bis zu dem das Zertifikat gültig ist.  <b>Information:</b> Die Eingabe wird nur auf den Tag genau ausgewertet																								
Subject	Sequenz aus X.509 Name-Wert-Paaren die durch ein "/"-Zeichen getrennt werden. Die folgenden Namen sind vorgesehen: <table border="1"> <thead> <tr> <th>Name</th> <th>Vollständiger Name</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>CN</td> <td>CommonName</td> <td>Name des Produkts oder vergleichbare Information</td> </tr> <tr> <td>O</td> <td>Organization</td> <td>Information Name der Organisation die den Bus Controller betreibt</td> </tr> <tr> <td>OU</td> <td>Organization Unit</td> <td>Organisationseinheit</td> </tr> <tr> <td>DC</td> <td>Domain Component</td> <td>Domain der Organisation</td> </tr> <tr> <td>L</td> <td>Locality</td> <td>Ort oder Stadt</td> </tr> <tr> <td>S</td> <td>State</td> <td>Bundesstaat</td> </tr> <tr> <td>C</td> <td>Country</td> <td>2-Zeichen Ländercode</td> </tr> </tbody> </table> <b>Information:</b> Die Angabe der Werte /CN und /O ist verpflichtend. <b>Beispiel</b> "/O=B&R Industrial Automation GmbH/CN=X20BC008T-OPCUA/DC=X20BC008T/DC=maschine/DC=customer/DC=com"	Name	Vollständiger Name	Beschreibung	CN	CommonName	Name des Produkts oder vergleichbare Information	O	Organization	Information Name der Organisation die den Bus Controller betreibt	OU	Organization Unit	Organisationseinheit	DC	Domain Component	Domain der Organisation	L	Locality	Ort oder Stadt	S	State	Bundesstaat	C	Country	2-Zeichen Ländercode
Name	Vollständiger Name	Beschreibung																							
CN	CommonName	Name des Produkts oder vergleichbare Information																							
O	Organization	Information Name der Organisation die den Bus Controller betreibt																							
OU	Organization Unit	Organisationseinheit																							
DC	Domain Component	Domain der Organisation																							
L	Locality	Ort oder Stadt																							
S	State	Bundesstaat																							
C	Country	2-Zeichen Ländercode																							
DNS (optional)	Hostname oder Fully Qualified Domain Name (FQDN) des Bus Controllers. Wenn bei diesem Parameter ein leerer String angegeben wird, wird der konfigurierte Hostname des Bus Controller in das Zertifikat eingetragen.																								
IncludeIPAddress	Gibt an, ob eine IP-Adresse in das Zertifikat eingetragen werden soll. Das Eintragen der IP-Adresse ist notwendig, wenn die IP-Adresse statisch vergeben ist und Clients mit Hilfe der IP-Adresse auf den Bus Controller zugreifen (Zum Beispiel über die URL opc.tcp://192.168.1.1:4840). Wird die IP-Adresse über einen DHCP-Server bezogen, ist es nicht sinnvoll eine IP-Adresse in das Zertifikat einzutragen, da sie dynamisch zugeteilt wird und nicht immer gleich ist.																								
IP Address (optional)	IP-Adresse, die in das Zertifikat eingetragen werden soll. Wenn hier ein leerer String übergeben wird und <i>IncludeIPAddress</i> gesetzt ist, dann wird die konfigurierte IP-Adresse in das Zertifikat eingetragen.																								
<b>Ausgangsargumente</b>																									
ApplyChangesRequired	Zeigt an, ob die Methode <i>Root/Objects/Server/ServerConfiguration/ApplyChanges</i> ausgeführt werden kann, um die Änderungen zu übernehmen.																								

- Wenn das Zertifikat erfolgreich erstellt werden konnte, muss im Anschluss die Methode *Root/Objects/Server/ServerConfiguration/ApplyChanges* (3) aufgerufen werden, um die Änderungen zu übernehmen.

### Information:

Beim Aufruf der Methode *ApplyChanges* werden alle verbundenen Clients getrennt. Eine neue Verbindung ist erst wieder möglich, wenn dem neuen Zertifikat vertraut wird.

### Information:

Da für das Zertifikatsmanagement möglicherweise private Schlüssel übertragen werden, ist der Aufruf nur möglich, wenn eine verschlüsselte Verbindung zwischen Bus Controller und OPC UA Client besteht.

## 4 Firmwareupdate über OPC UA

Mit der Firmwareupdate-Funktionalität lässt sich über OPC UA die Firmware des Bus Controller auf einen beliebigen Versionsstand bringen. Dabei bleibt sichergestellt, dass auch bei einem Spannungsausfall oder einer Unterbrechung der Übertragung stets eine kommunikationsfähige Firmware geladen wird.

Der Updatemechanismus richtet sich nach der Spezifikation "OPC 10000-100 - UA Specification Part 100 - Devices 1.03.0" und verwendet die "Cached-Loading" Option, bei der die Firmwaredatei zuerst auf den Server geladen und in einem zweiten Schritt installiert wird. Zuletzt muss die installierte Firmware noch aktiviert werden.

Um ein Firmwareupdate durchzuführen wird ein UaExpert Client benötigt, der folgende OPC UA Typen unterstützt:

- FileType
- TemporaryFileTransferType
- OptionSet

Für Details zur Durchführung des Firmwareupdates mit UaExpert siehe Abschnitt 4.1 "Update durchführen".

Für eine detaillierte Beschreibung der Struktur des Firmwareupdate Objekts siehe 11.1.2 " Firmwareupdate".

### 4.1 Update durchführen

Alle benötigten Methoden und Statusinformationen für den Firmwareupdate befinden sich unter *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate*. Zusätzlich wird noch die Methode "Reboot" unter *Root/Objects/DeviceSet/X20BC008T/Configuration/Control/Reboot* benötigt.

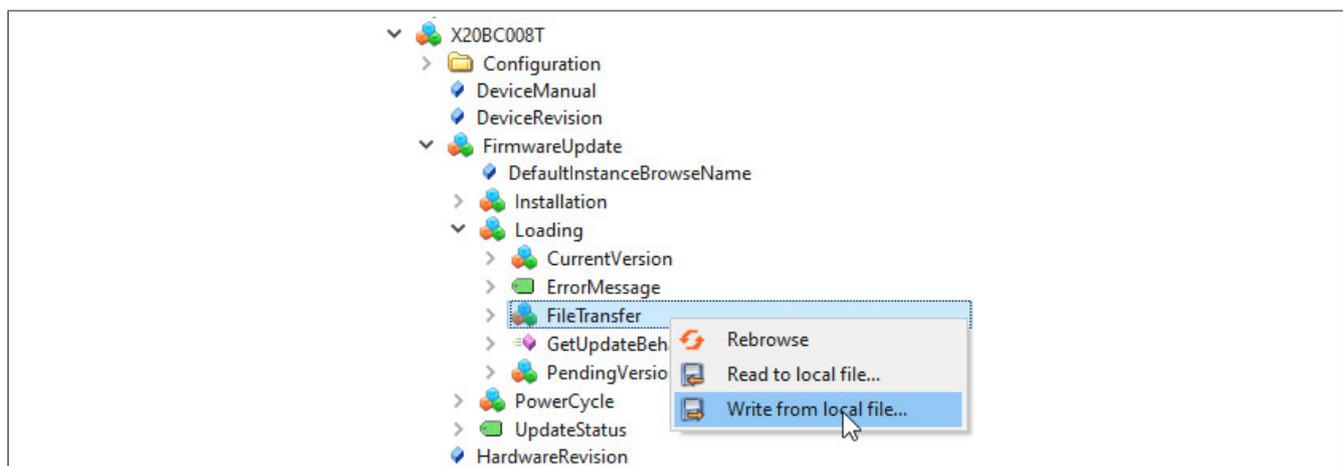
Ein Firmwareupdate kann mit UaExpert auf einfache Art durchgeführt werden. Dazu sind folgende Schritte nötig:

#### • Vorbereitung

Gewünschte Firmwareupdate-Datei von der [B&R Homepage \(https://www.br-automation.com\)](https://www.br-automation.com) herunterladen und entpacken.

#### • Übertragung

Nachdem eine Verbindung mit dem Bus Controller hergestellt wurde, im Objekt *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate/Loading/FileTransfer* einen Rechtsklick auf *Write from local file ...* durchführen und die entpackte Firmwareupdate Datei (\*.fw) auswählen.



• Die ausgewählte Datei wird von UaExpert auf den Bus Controller übertragen. Zur Kontrolle kann die zu installierende Datei durch Aufruf der Methode *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate/Loading/Pending-Version/SoftwareRevision* überprüft werden. Diese muss mit der gerade übertragenen Datei übereinstimmen und lässt sich anhand des letzten Teils des Dateinamens ermitteln:

- <Bestellnummer>\*V<SoftwareRevision>.zip

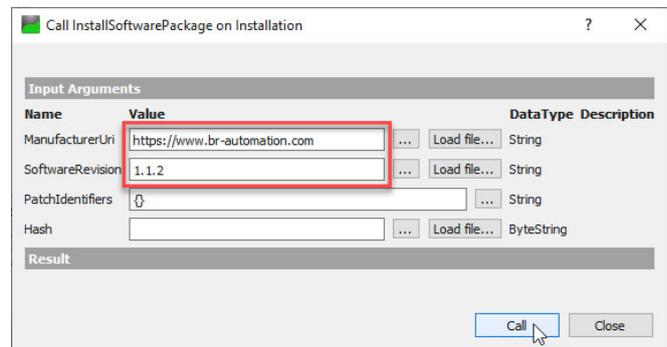
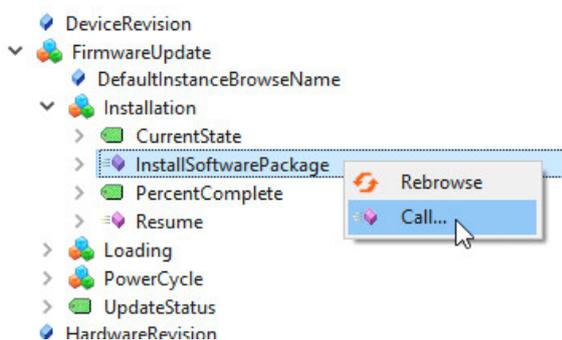
#### Beispiel

X20BC008T\_FIRMWARE\_V1.0.0.zip → entspricht SoftwareRevision = 1.0.0

#### • Installation

Im Objekt *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate/Installation/InstallSoftwarePackage* einen Rechtsklick auf *Call* durchführen und die erforderlichen Parameter eintragen. Diese sind:

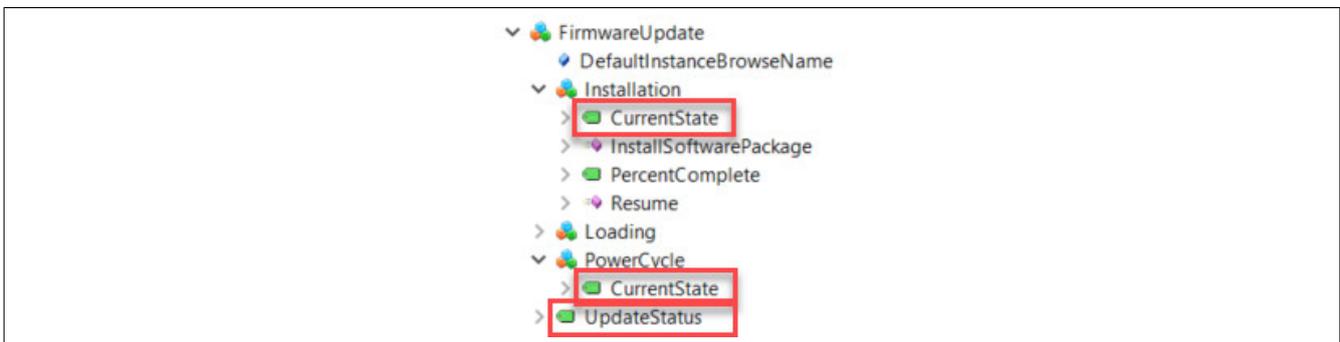
- ManufacturerURI: "https://www.br-automation.com"
- SoftwareRevision: entsprechend Beispiel oben



• Installation mit Klick auf *Call* abschließen und warten, bis die Installation abgeschlossen wurde. Der Status einer erfolgreichen Installation kann mit folgenden Parametern überprüft werden:

- Parameter *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate/Installation/CurrentState* zeigt "Installing"
- Parameter *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate/PowerCycle/CurrentState* zeigt "WaitingForPowerCycle"

Beide Parameter müssen den beschriebenen Wert anzeigen. Alternativ kann auch der Parameter *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate/UpdateStatus* ausgewertet werden. Dieser sollte den Wert "[INFO] Installation successful, reboot required" enthalten.



#### Information:

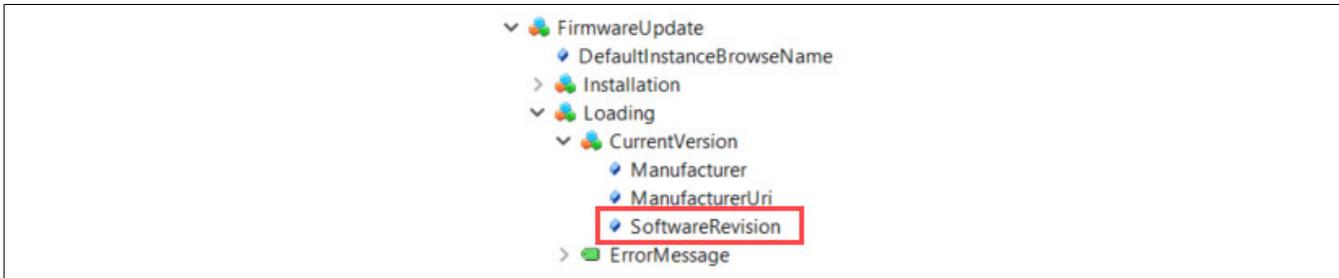
Die Firmware-Installation kann bis zu einer Minute dauern.

Der anschließende Neustart darf erst durchgeführt werden, wenn der Parameter *Root/Objects/DeviceSet/X20BC008T/PowerCycle/CurrentState* den Status "WaitingForPowerCycle" anzeigt. Ansonsten wird das Firmware-Update abgebrochen und das Gerät bootet wieder mit der alten Version.

## • Neustart und Überprüfung

Einen Neustart durchführen. Dieser kann durch Aufruf der Methode *Root/Objects/DeviceSet/X20BC008T/Configuration/Control/Reboot* (siehe 6.1.1 "Methoden für Bus Controller Konfiguration") oder durch ein Aus- und Einschalten der Spannungsversorgung erfolgen.

Nach dem Neustart kann die erfolgreiche Aktivierung des Firmwareupdates überprüft werden. Das geschieht durch Auslesen des Knotens *Root/Objects/DeviceSet/X20BC008T/Loading/CurrentVersion/SoftwareRevision*. Die angezeigte Firmwareversion muss mit der SoftwareRevision der Firmwareupdate "\*.zip"-Datei identisch sein.



## • Fehlerbehandlung

Falls während des Firmwareupdates ein gravierender Fehler auftritt, muss dieser zurückgesetzt werden, da im Fehlerzustand kein weiterer Firmwareupdate möglich ist. Dies kann durch folgende Möglichkeiten geschehen:

- Quittierung des Fehlers mittels der Methode *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate/Installation/Resume*
- Rücksetzen des Fehlerzustands durch einen Neustart. Dadurch wird wieder die ursprüngliche Firmware geladen.

Das fehlgeschlagene Firmwareupdate wird durch jede der beiden Methoden korrekt abgebrochen und beendet.

## 5 Features / Funktionalität

### 5.1 Unterstützte Module

Die folgende Tabelle zeigt alle in der Datenbank des Bus Controllers gespeicherten I/O-Module.

Bestellnummer	Beschreibung
<b>X20 Module</b>	
X20AI1744	X20 Analoges Eingangsmodul, 1 DMS-Vollbrücken-Eingang, 24 Bit Wandlerauflösung, 5 kHz Eingangsfilter
X20(c)AI1744-3	X20 Analoges Eingangsmodul, 1 DMS-Vollbrücken-Eingang, 24 Bit Wandlerauflösung, 5 Hz Eingangsfilter
X20AI1744-10	X20 Analoges Eingangsmodul, 1 DMS-Vollbrücken-Eingang 10 V, 24 Bit Wandlerauflösung, 5 kHz Eingangsfilter
X20AI2222	X20 Analoges Eingangsmodul, 2 Eingänge, ±10 V, 13 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X20AI2237	X20 Analoges Eingangsmodul, 2 Eingänge, ±10 V, 16 Bit Wandlerauflösung, Einzelkanal galvanisch getrennt und mit eigener Sensorversorgung, NetTime-Funktion
X20AI2322	X20 Analoges Eingangsmodul, 2 Eingänge, 0 bis 20 mA / 4 bis 20 mA, 12 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X20AI2622	X20 Analoges Eingangsmodul, 2 Eingänge, ±10 V oder 0 bis 20 mA / 4 bis 20 mA, 13 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X20AI4222	X20 Analoges Eingangsmodul, 4 Eingänge, ±10 V, 13 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X20AI4322	X20 Analoges Eingangsmodul, 4 Eingänge, 0 bis 20 mA / 4 bis 20 mA, 12 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X20(c)AI4622	X20 Analoges Eingangsmodul, 4 Eingänge, ±10 V oder 0 bis 20 mA / 4 bis 20 mA, 13 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X20AI8221	X20 Analoges Eingangsmodul, 8 Eingänge, ±10 V, 13 Bit Wandlerauflösung
X20AI8321	X20 Analoges Eingangsmodul, 8 Eingänge, 0 bis 20 mA, 12 Bit Wandlerauflösung
X20AIA744	X20 Analoges Eingangsmodul, 2 DMS Vollbrücken Eingänge, 24 Bit Wandlerauflösung, 2,5 kHz Eingangsfilter
X20AIB744	X20 Analoges Eingangsmodul, 4 DMS-Vollbrücken-Eingänge, 24 Bit Wandlerauflösung, 2,5 kHz Eingangsfilter
X20(c)AO2437	X20 Analoges Ausgangsmodul, 2 Ausgänge, 4 bis 20 mA / 0 bis 20 mA oder 0 bis 24 mA, 16 Bit Wandlerauflösung, Einzelkanal galvanisch getrennt
X20AO2622	X20 Analoges Ausgangsmodul, 2 Ausgänge, ±10 V oder 0 bis 20 mA / 4 bis 20 mA, 13 Bit Wandlerauflösung
X20(c)AO4622	X20 Analoges Ausgangsmodul, 4 Ausgänge, ±10 V oder 0 bis 20 mA / 4 bis 20 mA, 13 Bit Wandlerauflösung
X20AO4635	X20 Analoges Ausgangsmodul, 4 Ausgänge, ±10 V oder 0 bis 20 mA, 16 Bit Wandlerauflösung, geringe Temperaturdrift
X20AT2222	X20 Temperatur-Eingangsmodul, 2 Eingänge Widerstandsmessung, PT100, PT1000, Auflösung 0,1°C, 3-Leitertechnik
X20AT2311	X20 Temperatur-Eingangsmodul, 2 Eingänge Widerstandsmessung, PT100, Auflösung 0,001°C, 4-Leitertechnik
X20AT2402	X20 Temperatur-Eingangsmodul, 2 Eingänge Thermoelement, Typ J, K, N, S, B, R, Auflösung 0,1°C
X20(c)AT4222	X20 Temperatur-Eingangsmodul, 4 Eingänge Widerstandsmessung, PT100, PT1000, Auflösung 0,1°C, 3-Leitertechnik
X20AT4232	X20 Temperatur-Eingangsmodul, 4 Eingänge Widerstandsmessung, NTC 10 kOhm, Auflösung 0,1°C, 2-Leitertechnik
X20(c)AT6402	X20 Temperatur-Eingangsmodul, 6 Eingänge Thermoelement, Typ J, K, N, S, B, R, Auflösung 0,1°C
X20(c)BR9300	X20 Busempfänger, X2X Link, Einspeisung für X2X Link und interne I/O-Versorgung
X20(c)BT9100	X20 Bussender, X2X Link, Einspeisung für interne I/O-Versorgung
X20BT9400	X20 Bussender, X2X Link, Einspeisung für interne I/O-Versorgung, X2X Link Versorgung für X67 Module, Verpolungsschutz, kurzschlussfest, überlastfest, Parallelschaltung möglich, Redundanzbetrieb möglich
X20CM1941	X20 Resolvermodul, 14 Bit Resolvereingang, Konverter bis zu 12 Bit ABR-Ausgang

Bestellnummer	Beschreibung
X20CM8281	X20 Universelles Mischmodul, 4 digitale Eingänge, 24 VDC, Sink, 1-Leitertechnik, 2 digitale Ausgänge, 0,5 A, Source, 1-Leitertechnik, 1 analoger Eingang, $\pm 10$ V oder 0 bis 20 mA / 4 bis 20 mA, 12 Bit Wandlerrauflösung, 1 analoger Ausgang, $\pm 10$ V / 0 bis 20 mA, 12 Bit Wandlerrauflösung, 2 Zähler als Ereigniszähler oder zur Torzeitmessung
X20CM8323	X20 PWM-Modul, 8 digitale Ausgänge zum Schalten von elektromechanischen Lasten, 0,6 A Dauerstrom, 2 A Spitzenstrom, Strommonitoring, Schaltzeitpunkterkennung
X20(c)DC1196	X20 Digitales Zählermodul, 1 ABR-Inkrementalgeber, 5 V, 600 kHz Eingangsfrequenz, 4-fach Auswertung
X20(c)DC1198	X20 Digitales Zählermodul, 1 SSI-Absolutwertgeber, 5 V, 1 MBit/s, 32 Bit
X20(c)DC1396	X20 Digitales Zählermodul, 1 ABR-Inkrementalgeber, 24 V, 100 kHz Eingangsfrequenz, 4-fach Auswertung
X20(c)DC2190	X20 Digitales Zählermodul, Ultraschall Wegmessmodul, Schnittstellen: EP-Start/Stopp, DPI/IP, 2 Wegmessstäbe, 4 Wegeerfassung
X20DC2395	X20 Digitales Zählermodul, 1 SSI-Absolutwertgeber, 24 V, 1 ABR-Inkrementalgeber, 24 V, 2 AB-Inkrementalgeber, 24 V, 4 Ereigniszähler oder 2 PWM, lokale Zeitmessfunktionen
X20DC2396	X20 Digitales Zählermodul, 2 ABR-Inkrementalgeber, 24 V, 100 kHz Eingangsfrequenz, 4-fach Auswertung
X20DC2398	X20 Digitales Zählermodul, 2 SSI-Absolutwertgeber, 24 V, 125 kBit/s, 32 Bit
X20DC4395	X20 Digitales Zählermodul, 2 SSI-Absolutwertgeber, 24 V, 2 ABR-Inkrementalgeber, 24 V, 4 AB-Inkrementalgeber, 24 V, 8 Ereigniszähler oder 4 PWM, lokale Zeitmessfunktionen
X20DI0471	X20 Digitales Eingangsmodul, 10 Eingänge, 5-48 VDC, Sink, Eingangsfilter parametrierbar, 1-Leitertechnik
X20DI2371	X20 Digitales Eingangsmodul, 2 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 3-Leitertechnik
X20DI2372	X20 Digitales Eingangsmodul, 2 Eingänge, 24 VDC, Source, Eingangsfilter parametrierbar, 3-Leitertechnik
X20DI2377	X20 Digitales Eingangsmodul, 2 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 2 Ereigniszähler 50 kHz, 3-Leitertechnik
X20(c)DI4371	X20 Digitales Eingangsmodul, 4 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 3-Leitertechnik
X20DI4372	X20 Digitales Eingangsmodul, 4 Eingänge, 24 VDC, Source, Eingangsfilter parametrierbar, 3-Leitertechnik
X20(c)DI4375	X20 Digitales Eingangsmodul, 4 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, Drahtbruch- und Kurzschlusserkennung, 3-Leitertechnik
X20DI4653	X20 Digitales Eingangsmodul, 4 Eingänge, 100 bis 240 VAC, 240 V codiert, 2-Leitertechnik
X20(c)DI4760	X20 Digitales Eingangsmodul, 4 NAMUR-Eingänge, 8,05 V
X20(c)DI6371	X20 Digitales Eingangsmodul, 6 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 2-Leitertechnik
X20(c)DI6372	X20 Digitales Eingangsmodul, 6 Eingänge, 24 VDC, Source, Eingangsfilter parametrierbar, 2-Leitertechnik
X20DI6373	X20 Digitales Eingangsmodul, 6 Eingänge, 24 VDC, Sink/Source, alle Eingänge potenzialfrei, Eingangsfilter parametrierbar, 2-Leitertechnik
X20DI6553	X20 Digitales Eingangsmodul, 6 Eingänge, 100 bis 120 VAC, 240 V codiert, 1-Leitertechnik
X20DI8371	X20 Digitales Eingangsmodul, 8 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 1-Leitertechnik
X20(c)DI9371	X20 Digitales Eingangsmodul, 12 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 1-Leitertechnik
X20(c)DI9372	X20 Digitales Eingangsmodul, 12 Eingänge, 24 VDC, Source, Eingangsfilter parametrierbar, 1-Leitertechnik
X20DID371	X20 Digitales Eingangsmodul, 8 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 2-Leitertechnik
X20(c)DIF371	X20 Digitales Eingangsmodul, 16 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 1-Leitertechnik
X20(c)DM9324	X20 Digitales Mischmodul, 8 Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, 4 Ausgänge, 24 VDC, 0,5 A, Source, 1-Leitertechnik
X20DO2321	X20 Digitales Ausgangsmodul, 2 Ausgänge, 24 VDC, 0,5 A, Sink, 3-Leitertechnik
X20DO2322	X20 Digitales Ausgangsmodul, 2 Ausgänge, 24 VDC, 0,5 A, Source, 3-Leitertechnik
X20DO2623	X20 Digitales Ausgangsmodul, 2 Ausgänge, 100 bis 240 VAC, 1 A, Source, 240 V codiert, 3-Leitertechnik
X20(c)DO2633	X20 Digitales Ausgangsmodul, 2 Triac-Ausgänge, 48 bis 240 VAC, 2 A, L-schaltend, Phasenanschnittsteuerung, 240 V codiert
X20DO2649	X20 Digitales Ausgangsmodul, 2 Relais, Wechslerkontakte, 240 VAC / 5 A, 24 VDC / 5 A
X20DO4321	X20 Digitales Ausgangsmodul, 4 Ausgänge, 24 VDC, 0,5 A, Sink, 3-Leitertechnik
X20(c)DO4322	X20 Digitales Ausgangsmodul, 4 Ausgänge, 24 VDC, 0,5 A, Source, 3-Leitertechnik
X20DO4332	X20 Digitales Ausgangsmodul, 4 Ausgänge, 24 VDC, 2 A, Source, 3-Leitertechnik
X20DO4332-1	X20 Digitales Ausgangsmodul, 4 Ausgänge, 24 VDC, 2 A, Source, 3-Leitertechnik, PWM-Ausgang
X20DO4529	X20 Digitales Ausgangsmodul, 4 Relais, Wechslerkontakte, 115 VAC / 0,5 A, 24 VDC / 1 A
X20DO4613	X20 Digitales Ausgangsmodul, 4 Triac-Koppler-Ausgänge, 48 bis 240 VAC, 50 mA, Nulldurchgangserkennung, 240 V codiert
X20DO4623	X20 Digitales Ausgangsmodul, 4 Ausgänge, 100 bis 240 VAC, 0,5 A, Source, 240 V codiert, 2-Leitertechnik
X20(c)DO4633	X20 Digitales Ausgangsmodul, 4 Triac-Ausgänge, 48 bis 240 VAC, 1 A, L-schaltend, Phasenanschnittsteuerung, 240 V codiert
X20(c)DO4649	X20 Digitales Ausgangsmodul, 4 Relais, Schließerkontakte, 240 VAC / 5 A
X20(c)DO6321	X20 Digitales Ausgangsmodul, 6 Ausgänge, 24 VDC, 0,5 A, Sink, 2-Leitertechnik
X20(c)DO6322	X20 Digitales Ausgangsmodul, 6 Ausgänge, 24 VDC, 0,5 A, Source, 2-Leitertechnik
X20DO6325	X20 Digitales Ausgangsmodul, 6 Ausgänge, 24 VDC, 0,5 A, Source, Drahtbruch- und Überlasterkennung, 2-Leitertechnik
X20(c)DO6529	X20 Digitales Ausgangsmodul, 6 Relais, Schließerkontakte, 115 VAC / 0,5 A, 30 VDC / 1 A
X20(c)DO6639	X20 Digitales Ausgangsmodul, 6 Relais, Schließerkontakte, 240 VAC / 2 A, 30 VDC / 2 A
X20DO8232	X20 Digitales Ausgangsmodul, 8 Ausgänge, 12 VDC, 2 A, Source, Einspeisung direkt am Modul, 1-Leitertechnik
X20DO8322	X20 Digitales Ausgangsmodul, 8 Ausgänge, 24 VDC, 0,5 A, Source, 1-Leitertechnik
X20DO8323	X20 Digitales Ausgangsmodul, 8 Ausgänge, 12 bis 24 V, 0,5 A, Sink/Source, 1-Leitertechnik, Vollbrücke, Halbbrücke, thermischer Überlastschutz
X20(c)DO8331	X20 Digitales Ausgangsmodul, 8 Ausgänge, 24 VDC, 2 A, Sink, Einspeisung direkt am Modul, 1-Leitertechnik
X20(c)DO8332	X20 Digitales Ausgangsmodul, 8 Ausgänge, 24 VDC, 2 A, Source, Einspeisung direkt am Modul, 1-Leitertechnik
X20DO8332-1	X20 Digitales Ausgangsmodul, 8 Ausgänge, 24 VDC, 2 A, Source, optimiert für induktive Lasten, Einspeisung direkt am Modul, 1-Leitertechnik
X20(c)DO9321	X20 Digitales Ausgangsmodul, 12 Ausgänge, 24 VDC, 0,5 A, Sink, 1-Leitertechnik
X20(c)DO9322	X20 Digitales Ausgangsmodul, 12 Ausgänge, 24 VDC, 0,5 A, Source, 1-Leitertechnik
X20DOD322	X20 Digitales Ausgangsmodul, 8 Ausgänge, 24 VDC, 0,5 A, Source, 2-Leitertechnik
X20(c)DOF322	X20 Digitales Ausgangsmodul, 16 Ausgänge, 24 VDC, 0,5 A, Source, 1-Leitertechnik
X20MM2436	X20 PWM-Motormodul, 24 bis 39 VDC $\pm 25\%$ , 2 PWM-Motorbrücken, 3 A Dauerstrom, 3,5 A Spitzenstrom, 4 digitale Eingänge 24 VDC, Sink, als Inkrementalgeber parametrierbar
X20MM3332	X20 Digitales Motormodul, 24 VDC, 3 digitale Ausgänge, Vollbrücke (H-Brücke), 3 A Dauerstrom, 5 A Spitzenstrom
X20MM4331	X20 Digitales Motormodul, 24 VDC, 4 digitale Ausgänge, Halbbrücke, 3 A Dauerstrom, 5 A Spitzenstrom
X20MM4456	X20 PWM-Motormodul, 24 bis 48 VDC $\pm 25\%$ , 4 PWM-Motorbrücken, 6 A Dauerstrom, 10 A Spitzenstrom, 4x 4 digitale Eingänge 24 VDC, Sink, als Inkrementalgeber parametrierbar
X20PD0011	X20 Potenzialverteilermodul, 12x GND, integrierte Feinsicherung

Bestellnummer	Beschreibung
X20PD0012	X20 Potenzialverteilermodul, 12x 24 VDC, integrierte Feinsicherung
X20PD0016	X20 Potenzialverteilermodul, 5x GND, 5x 24 VDC, potenzialfreie Einspeisung, integrierte Feinsicherung
X20(c)PD2113	X20 Potenzialverteilermodul, 6x GND, 6x 24 VDC, mit Einspeisemöglichkeit, integrierte Feinsicherung
X20(c)PS2100	X20 Einspeisemodul, für interne I/O-Versorgung
X20(c)PS2110	X20 Einspeisemodul, für interne I/O-Versorgung, integrierte Feinsicherung
X20(c)PS3300	X20 Einspeisemodul, für X2X Link und interne I/O-Versorgung
X20(c)PS3310	X20 Einspeisemodul, für X2X Link und interne I/O-Versorgung, integrierte Feinsicherung
X20PS4951	X20 Einspeisemodul, für Potentiometer, 4x $\pm 10$ V für Potentiometerversorgung
X20(c)PS9400	X20 Einspeisemodul, für Bus Controller und interne I/O-Versorgung, X2X Link Versorgung
X20PS9402	X20 Einspeisemodul, für Bus Controller und interne I/O-Versorgung, X2X Link Versorgung, Einspeisung galvanisch nicht getrennt
X20PS9600	X20 Einspeisemodul, für Compact-S CPU und interne I/O-Versorgung, X2X Link Versorgung
X20PS9602	X20 Einspeisemodul, für Compact-S CPU und interne I/O-Versorgung, X2X Link Versorgung, Einspeisung galvanisch nicht getrennt
<b>X67 Module</b>	
X67AI1223	X67 Analoges Eingangsmodul, 4 Eingänge, $\pm 10$ V, 12 Bit Wandlerauflösung, Eingangsfilter parametrierbar, Drahtbrucherkennung
X67AI1233	X67 Analoges Eingangsmodul, 4 Eingänge, $\pm 10$ V, 16 Bit Wandlerauflösung, Eingangsfilter parametrierbar, Drahtbrucherkennung
X67AI1323	X67 Analoges Eingangsmodul, 4 Eingänge, 0 bis 20 mA oder 4 bis 20 mA, 12 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X67AI1333	X67 Analoges Eingangsmodul, 4 Eingänge, 0 bis 20 mA oder 4 bis 20 mA, 16 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X67AI2744	X67 Analoges Eingangsmodul, 2 DMS-Vollbrücken Eingänge, 10 V, 24 Bit Wandlerauflösung
X67AI4850	X67 Analoges Eingangsmodul, 4 Eingänge, Potentiometer Wegaufnehmer 15 Bit
X67AM1223	X67 Analoges Mischmodul, 2 Eingänge, 2 Ausgänge, $\pm 10$ V, 12 Bit Wandlerauflösung, Eingangsfilter parametrierbar, Drahtbrucherkennung bei den Eingängen
X67AM1323	X67 Analoges Mischmodul, 2 Eingänge, 2 Ausgänge, 0 bis 20 mA, 12 Bit Wandlerauflösung, Eingangsfilter parametrierbar
X67AO1223	X67 Analoges Ausgangsmodul, 4 Ausgänge, $\pm 10$ V, 12 Bit Wandlerauflösung
X67AO1323	X67 Analoges Ausgangsmodul, 4 Ausgänge, 0 bis 20 mA, 12 Bit Wandlerauflösung
X67AT1311	X67 Temperatur Eingangsmodul, 4 Eingänge Widerstandsmessung, 2- oder 4-Leitermessung, PT100, Auflösung 0,01 K
X67AT1322	X67 Temperatur Eingangsmodul, 4 Eingänge Widerstandsmessung, 2- oder 4-Leitermessung, PT100, PT1000, KTY10, KTY84, Auflösung 0,1 K
X67AT1402	X67 Temperatur Eingangsmodul, 4 Eingänge Thermoelemente, Typ J, K, N, R, S, Auflösung 0,1 K
X67DI1371	X67 Digitales Eingangsmodul, 8 Eingänge, 24 VDC, Sink, Eingangsfilter 1 ms
X67DI1371.Lxx	X67 Digitales Eingangsmodul, 16 Eingänge, 24 VDC, Sink, Eingangsfilter 1 ms, High-Density-Modul
X67DI1372	X67 Digitales Eingangsmodul, 8 Eingänge, 24 VDC, Source, Eingangsfilter 1 ms
X67DM1321	X67 Digitales Mischmodul, 8 Kanäle wahlweise als Ein- oder Ausgang parametrierbar, 24 VDC, 0,5 A, Eingangsfilter parametrierbar, 2 Ereigniszähler 50 kHz
X67DM1321.Lxx	X67 Digitales Mischmodul, 16 Kanäle wahlweise als Ein- oder Ausgang parametrierbar, 24 VDC, 0,5 A, Eingangsfilter parametrierbar, 2 Ereigniszähler 50 kHz, M8-Anschlussstechnik, High-Density-Modul
X67DM1321.L12-1	X67 Digitales Mischmodul, 16 Kanäle wahlweise als Ein- oder Ausgang parametrierbar, 24 VDC, 0,5 A, Pinning-Variante, Eingangsfilter parametrierbar, 2 Ereigniszähler 50 kHz, M12-Anschlussstechnik, High-Density-Modul
X67DM9321	X67 Digitales Mischmodul, 8 Kanäle wahlweise als Ein- oder Ausgang parametrierbar, 24 VDC, 0,5 A, Eingangsfilter parametrierbar, 2 Ereigniszähler 50 kHz, X2X Link Adressschalter
X67DM9321.L12	X67 Digitales Mischmodul, 16 Kanäle wahlweise als Ein- oder Ausgang parametrierbar, 24 VDC, 0,5 A, Eingangsfilter parametrierbar, 2 Ereigniszähler 50 kHz, M12-Anschlussstechnik, X2X Link Adressschalter, High-Density-Modul
X67DM9331.L12	X67 Digitales Mischmodul, 8 Kanäle wahlweise als Ein- oder Ausgang parametrierbar, 24 VDC, 2 A, Eingangsfilter parametrierbar, Sensor-/Aktorversorgung einzelkanalüberwacht, M12-Anschlussstechnik, X2X Link Adressschalter, High-Density-Modul
X67DO1332	X67 Digitales Ausgangsmodul, 8 Ausgänge, 24 VDC, 2 A, Ausgangsstatus rücklesbar
X67DO9332.L12	X67 Digitales Ausgangsmodul, 8 Ausgänge, 24 VDC, 2 A, Aktorversorgung einzelkanalüberwacht, M12-Anschlussstechnik, X2X Link Adressschalter, High-Density-Modul
X67DV1311.L08	X67 Digitales Ventilsteuerungsmodul, 16 digitale Ausgänge, 24 VDC, 0,1 A, 1 M16-Anschluss, 16 digitale Eingänge, 24 VDC, Sink, Eingangsfilter parametrierbar, M8-Anschlussstechnik, High-Density-Modul
X67SM2436	X67 Schrittmotormodul, I/O-Versorgung 24-38,5 VDC $\pm 25\%$ , 8 A max., 2 Motoranschlüsse, 3 A Dauerstrom, 5 A Spitzenstrom, 2x 3 digitale Eingänge 24 VDC, Sink, als Inkrementalgeber parametrierbar, NetTime-Funktion
<b>Motorstartermodul</b>	
SFM1-A11.1	Smart Function Modul mit X2X

## 5.2 Verwendete Namespaces

Im Bus Controller werden folgende Namespaces verwendet:

Index	Namespace URL	Beschreibung
0	http://opcfoundation.org/UA/	Adressraum für Typen und Objekte, welche in der OPC UA Spezifikation definiert sind
1	http://br-automation.com/OpcUa/X20BC008T/<Seriennummer>/	Dieser Namespace-URL stellt den Adressraum des Bus Controllers dar, auf dem der OPC UA Server läuft. Die <Seriennummer> entspricht der Seriennummer des Bus Controllers
2	http://opcfoundation.org/UA/DI/	Adressraum für Typen und Objekte, welche in der OPC UA Companion Spezifikation für Geräteintegration (DI = Device Integration) definiert sind.
3	http://br-automation.com/OpcUa/BrDevice	Basis-Informationsmodell für B&R Feldgeräte
4	http://br-automation.com/OpcUa/io-system	Informationsmodell des Bus Controllers

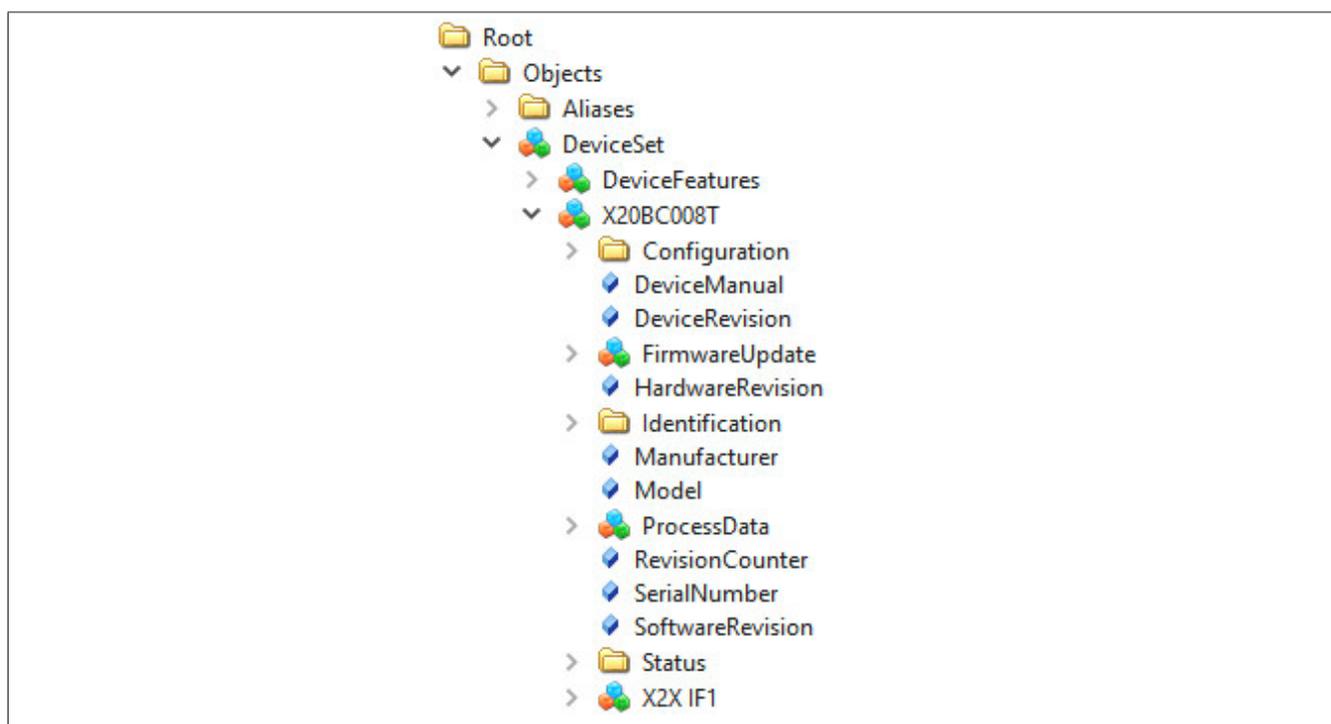
Die verwendeten Namespaces können auch im OPC UA Informationsmodell ausgelesen werden:

The screenshot shows the OPC UA Information Model Explorer. On the left, the tree view is expanded to 'Server' > 'NamespaceArray'. On the right, the properties of 'NamespaceArray' are displayed:

- UserRolePermissions**: RolePermissionType Array[2]
- AccessRestrictions**: BadAttributeldInvalid (0x80350000)
- Value**:
  - SourceTimestamp: 12.07.2021 15:19:05.387
  - SourcePicoSeconds: 0
  - ServerTimestamp: 12.07.2021 15:19:05.387
  - ServerPicoSeconds: 0
  - StatusCode: Good (0x00000000)
  - Value: String Array[5]
    - [0]: http://opcfoundation.org/UA/
    - [1]: http://br-automation.com/OpcUa/X20BC008T/F6290168454/
    - [2]: http://opcfoundation.org/UA/DI/
    - [3]: http://br-automation.com/OpcUa/BrDevice
    - [4]: http://br-automation.com/OpcUa/BC/io-system/
- DataType**: String
- NamespaceIndex**: 0
- IdentifierType**: Numeric
- Identifier**: 12 [String]
- ValueRank**: 1 (OneDimension)

### 5.3 Geräteinformation

Unter dem Knoten *Root/Objects/DeviceSet/X20BC008T* befinden sich weitere Knoten, durch die Basisinformationen des Bus Controllers ausgelesen werden können:



Knotenname	Beschreibung
DeviceManual	URL, unter der weitere Informationen zum Modul zur Verfügung stehen
DeviceRevision bzw. Processdata/HardwareVariant	B&R Hardwarevariante
HardwareRevision	Hardwarerevision des Bus Controllers
Manufacturer	Hersteller des Bus Controllers
Model	Modulbezeichnung
RevisionCounter	Reserviert (immer -1)
SerialNumber	Vollständige Seriennummer als String
SoftwareRevision	Aktuelle Softwarerevision
Identification/ModuleID bzw. Processdata/ModuleID	Numerische Identifikationsnummer des Moduls
Processdata/SerialNumber	Seriennummer als 32 Bit Integer

## 5.4 Zeitsynchronisation und Zeitdomänen

Der Bus Controller verfügt über 2 voneinander unabhängige Uhren, die mit unterschiedlichen Zeitdomänen synchronisiert werden können. Damit bekommen alle Netzwerkgeräte, welche zur gleichen Zeitdomäne synchronisiert sind, ein einheitliches Zeitverhalten. Das heißt, dass sowohl die Zeitwerte als auch die Frequenzen der Uhren miteinander abgestimmt werden. Dadurch lassen sich Aktivitäten auf diversen Geräten zeitlich exakt koordinieren und deren Zeitstempel in eine genaue Zeitabfolge bringen.

Für die Zeitsynchronisation am Bus Controller kann entweder das Network Time Protocol (NTP) oder das IEEE 802.1AS-2020 Profil des Precision Time Protocol (PTP) verwendet werden.

Die beiden Uhren können wie folgt eingesetzt werden:

### WallClock

Die Wallclock entspricht der klassischen Systemuhr. Sie kann über NTP oder PTP mit der aktuelle UTC-Zeit synchronisiert und beispielsweise für Logging-Zeitstempel oder Zertifikatsvalidierung verwendet werden.

### WorkingClock

Die WorkingClock ist unabhängig von der UTC-Zeit und wird vor allem für das zeitgenaue Versenden der TSN-Ethernet-Frames verwendet. Im Vergleich zur WallClock wird bei der WorkingClock sichergestellt, dass nach der erstmaligen Synchronisation keine weiteren Sprünge im Zeitverlauf mehr erfolgen (z. B. durch Schaltsekunden, wie sie bei UTC-Zeit vorkommen). Um sicherzustellen, dass TSN-Ethernet-Frames ausreichend genau versendet werden, ist bei der WorkingClock die Anforderung an die Synchronisationsgenauigkeit viel höher. Aus diesem Grund steht für die WorkingClock nur PTP als Synchronisationsmethode zu Verfügung.

### Information:

**Es ist zu beachten, dass am Bus Controller die PTP-Zeitsynchronisation für jede PTP-Zeitdomäne, die von verbundenen Netzwerkgeräten verwendet wird, aktiviert werden muss. Dies ist auch notwendig, wenn der Bus Controller die entsprechende Domäne selbst nicht benutzt.**

## 5.5 Netzwerkmanagementprotokolle

Für das erweiterte Netzwerkmanagement unterstützt der Bus Controller weitere Protokolle, die automatisch beim Hochlauf aktiviert werden.

### 5.5.1 Multiple Spanning Tree Protocol (MSTP)

Dieses Protokoll dient der logischen Auftrennung redundanter Verbindungen im Netzwerk, sodass Broadcast-Frames nicht mehrfach im Kreis gesendet werden können, was eine unnötige Belastung des Netzwerks darstellen würde. Durch Austausch von Konfigurationsnachrichten wird eine logische Baumtopologie erstellt, welche die Weiterleitung von Ethernet-Frames auf redundanten Pfaden verhindert. Im Falle von MSTP wird für alle vorhandenen Virtuellen Local Area Networks (VLANs) ein eigener logischer Baum aufgebaut.

Da sich die logischen Baumtopologien im Netzwerk dynamisch ändern können, sind diese für zeitgesteuerte Kommunikation ungeeignet. Daher ist es notwendig, Netzwerkpfade für zeitgesteuerte Kommunikation explizit mittels TSN-Konfiguration festzulegen. Das geschieht durch Konfiguration von Weiterleitungsregeln. Diese festgelegten Pfade werden nicht durch MSTP beeinflusst.

Detailinformationen zu MSTP können dem Standard IEEE 802.1Q entnommen werden. Die Konfiguration des MSTP Stacks oder Statusabfragen können via NETCONF durchgeführt werden. Die Konfigurationsparameter und Statuswerte sind dem entsprechenden YANG-Modell zu entnehmen.

### 5.5.2 Link Layer Discovery Protocol (LLDP)

LLDP wird dazu benutzt, um zwischen benachbarten Netzwerkgeräten Informationen zur Identität und zu unterstützten Funktionalitäten auszutauschen. Diese Informationen werden individuell für jeden Port gesammelt, an dem ein LLDP-fähiges Gerät angeschlossen ist.

Detailinformationen zu LLDP können dem Standard IEEE 802.1AB entnommen werden. Eine Statusabfrage ist via NETCONF, mit dem dazugehörigen YANG-Modell, möglich. Zu Diagnosezwecken sind einige der Statuswerte auch im OPC UA Informationsmodell aufgelegt (siehe dazu 7.1 "Port-Status").

## 5.6 Integration im IT-Netzwerk

Beim Hochlauf des Bus Controllers wird automatisch der Multiple Spanning Tree Protocol (MSTP) Stack gestartet (siehe 5.5.1 "Multiple Spanning Tree Protocol (MSTP)"). Bei der Ausführung dieses Stacks werden Konfigurationsnachrichten zwischen Netzwerkgeräten ausgetauscht, welche mittels Bridge Protocol Data Unit (BPDU) Frames übertragen werden.

Diese Konfigurationsdaten können die logische Topologie eines Netzwerks beeinflussen, was insbesondere im Fehlerfall oder im Falle bewusster Manipulation zu unerwünschtem Verhalten im Netzwerk führen kann. Um derartigen Problemen vorzubeugen, unterstützen viele handelsübliche Switches sogenannte BPDU-Filter, welche auf einzelne Ports angewandt werden können. Einerseits können diese Filter dazu genutzt werden, um BPDU-Pakete zu verwerfen. Andererseits ist es möglich, den Port, mit dem das verursachende Gerät verbunden ist, zu sperren.

Bei der Integration des Bus Controllers in ein IT-Netzwerk sind die Vorgaben der IT-Administration zu beachten, damit es zu keinen Störungen kommt, z. B. durch automatischen Ausschluss des Ports, an dem der Bus Controller angeschlossen wurde. Sind im IT-Netzwerk keine BPDU-Pakete erlaubt, so sollte an dem IT-Switch, welcher für den Bus Controller als Zugang zum IT-Netzwerk dient, ein BPDU-Filter eingestellt werden, der entsprechende Pakete an der Weiterleitung hindert.

## 5.7 Integration im TSN-Netzwerk

Bis Softwareversion 1.4.x wird die Integration des Bus Controllers in ein TSN-Netzwerk nur als Endknoten (Endpoint) unterstützt, das heißt, es darf nur 1 Ethernet-Port verwendet werden. Insbesondere die Weiterleitung von Datagrammen mit VLAN-Tag über beide Ethernet-Ports wird nicht unterstützt. Der Bus Controller sollte daher direkt an einen TSN-fähigen Switch angeschlossen werden.

## 5.8 Geräteeigenschaften

Die Ports des Bus Controllers, deren Bezeichnungen und TSN-Fähigkeit sind in der folgenden Tabelle aufgelistet.

Bezeichnung im OPC UA Informationsmodell	Interne Bezeichnung (LLDP)	TSN-fähig
ETH1	sw0p2	Ja
ETH2	sw0p3	Ja

### Information:

Die internen Bezeichnungen **sw0p1** und **sw0ep** werden jeweils für den Internen und den Management Port des Bus Controllers verwendet. Diese haben keinen Bezug zu den externen Ports des Bus Controllers.

Die folgende Tabelle zeigt die Dimensionierung verschiedener Eigenschaften des Bus Controllers. Wenn nicht explizit angegeben, gelten die angegebenen Werte global für den gesamten Bus Controller.

Eigenschaft	Wert
Anzahl Filtering Database (FDB) Einträge	512
Maximale Anzahl VIDs	64
Anzahl Queues pro Port	8
Anzahl Gate Control List (GCL) Einträge pro Port	255

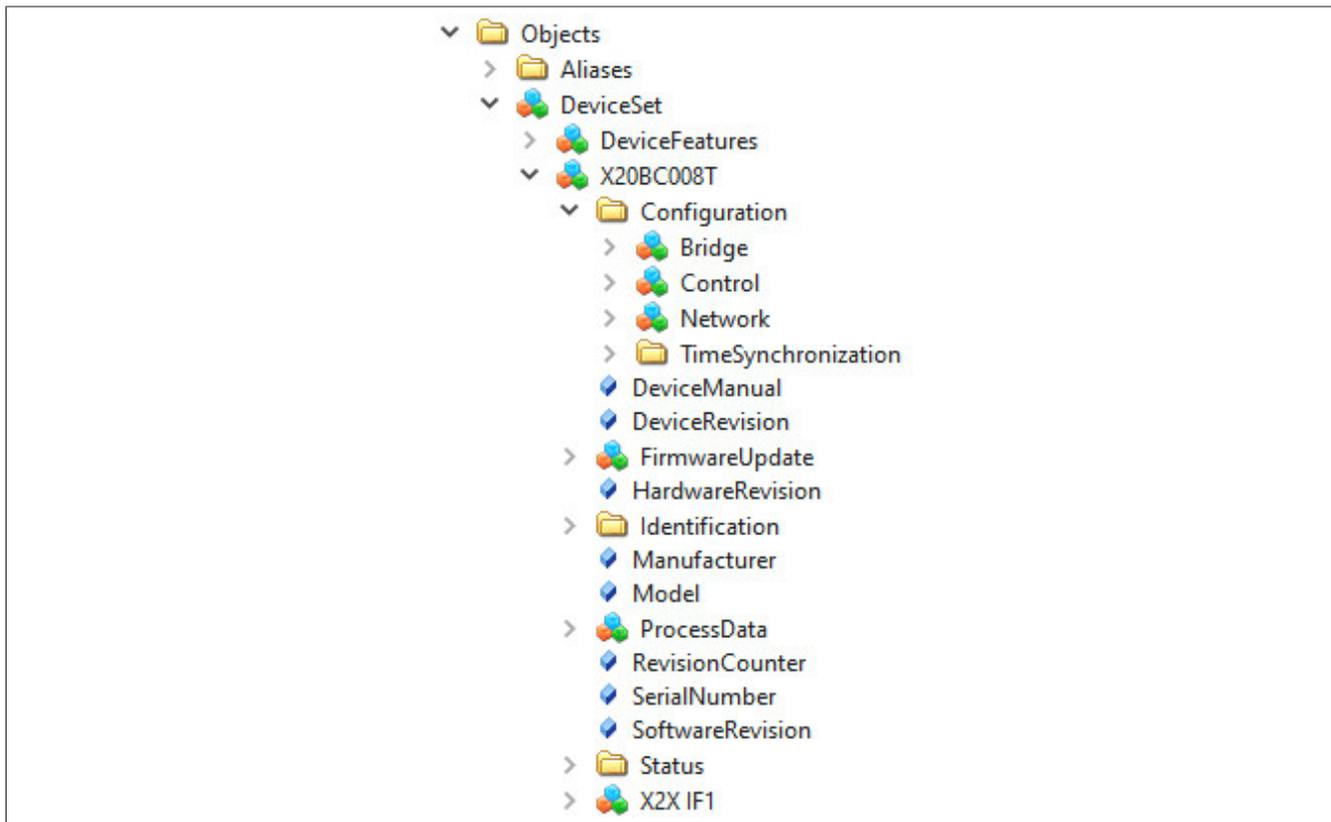
## 6 Konfiguration

Die Konfiguration erfolgt, indem Werte auf entsprechende OPC UA Variablenknoten geschrieben werden. Diese Konfigurationsknoten sind als hierarchische OPC UA Objekte organisiert, die jeweils eine bestimmte Funktionalität gruppieren.

Geschriebene Konfigurationswerte werden nicht unmittelbar übernommen. Erst durch den Aufruf einer Methode [6.1.1.1 "ApplyChanges"](#) werden Änderungen innerhalb des jeweiligen Objekts gespeichert und angewandt.

### 6.1 Bus Controller Objekt

Dieses Objekt enthält alle Daten und Methoden der Bus Controller Applikation.

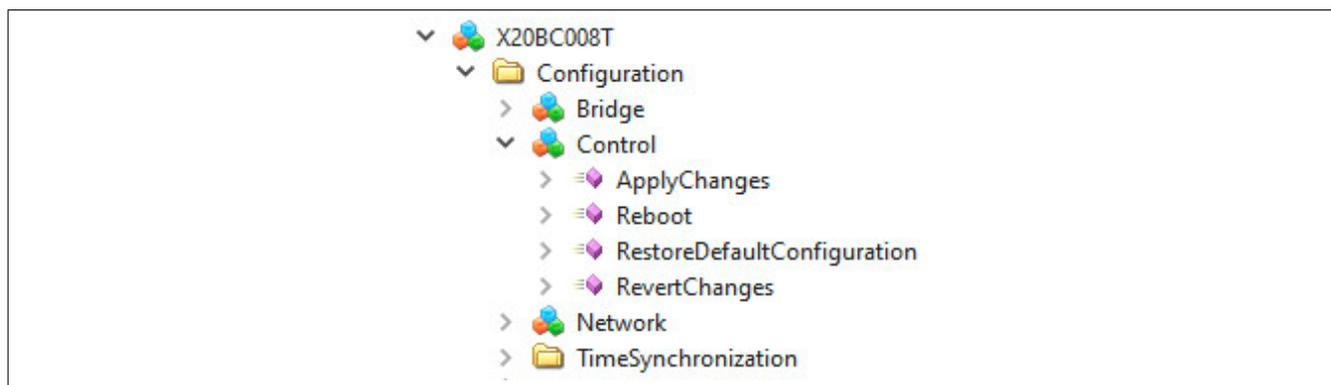


#### Information:

Die Methoden [6.1.1.1 "ApplyChanges"](#), [6.1.1.4 "RevertChanges"](#) und [6.1.1.3 "RestoreDefaultConfiguration"](#) beziehen sich nur auf die Konfigurationswerte innerhalb des übergeordneten "Konfiguration"-Ordners. Andere Konfigurationen, wie die X2X-Konfiguration oder die Benutzerkonfiguration, werden davon NICHT beeinflusst.

## 6.1.1 Methoden für Bus Controller Konfiguration

Position der Methoden im Informationsmodell: *Root/Objects/DeviceSet/X20BC008T/Configuration/Control*



### 6.1.1.1 ApplyChanges

Geänderte Werte werden erst durch einen Aufruf dieser Methode gespeichert und übernommen.

### 6.1.1.2 Reboot

Löst einen Neustart des Bus Controllers aus.

### 6.1.1.3 RestoreDefaultConfiguration

Die Default-Konfigurationswerte werden wiederhergestellt.

#### **Information:**

Durch den Aufruf der Methode werden die Default-Konfigurationswerte nur temporär in die Knoten geladen. Um die Default-Konfiguration zu Speichern bzw. zu Übernehmen ist zusätzlich ein Aufruf der Methode *ApplyChanges* notwendig.

### 6.1.1.4 RevertChanges

Die zuletzt mit *ApplyChanges* gespeicherten Werte werden wiederhergestellt.

## 6.1.2 Allgemeine Netzwerkkonfiguration

Die Konfiguration kann über das OPC UA Informationsmodell vorgenommen werden. Die entsprechenden Parameter befinden sich im Modell unter dem Knoten *Root/Objects/DeviceSet/X20BC008T/Configuration/Network*.

Knotenname	Beschreibung
EnableDHCP	Aktiviert beziehungsweise deaktiviert die DHCP-Client-Funktionalität - Bei fehlender IP-Zuweisung durch einen DHCP-Server wird dem Bus Controller eine zufällige Link Local Adresse aus dem Bereich 169.254.0.0/16 zugewiesen. IPv4LL (RFC3927). - Wenn der DHCP-Client aktiviert ist, werden die Parameter <i>Gateway</i> , <i>IP Address</i> , <i>Netmask</i> , sowie <i>Primary DNS</i> und <i>Secondary DNS</i> vom DHCP-Server bezogen.
Gateway	Konfiguration der Default-Gateway IP-Adresse - Wenn der Parameter <i>EnableDHCP</i> gesetzt ist, kann zusätzlich eine Gateway-Adresse vom DHCP-Server übermittelt werden. - Wenn der Parameter <i>Gateway</i> gesetzt ist, wird die manuelle Konfiguration verwendet und die vom DHCP-Server übermittelte Adresse ignoriert.
Hostname	Konfiguration des Hostnamens
IP-Address	Konfiguration einer statischen IP-Adresse - Wenn der Parameter <i>EnableDHCP</i> gesetzt ist, dann wird der Parameter ignoriert und die vom DHCP-Server übermittelte IP-Adresse verwendet.
Primary DNS Secondary DNS	Konfiguration eines primären bzw. sekundären DNS-Servers - Wenn der Parameter <i>EnableDHCP</i> gesetzt ist, können zusätzlich Adressen für DNS-Server vom DHCP-Server übermittelt werden. - Wenn mindestens 1 DNS-Server manuell gesetzt ist, wird die manuelle Konfiguration verwendet und die vom DHCP-Server übermittelten Adressen werden ignoriert.
Netmask	Einstellung der Subnetzmaske - Wenn der Parameter <i>EnableDHCP</i> gesetzt ist, wird dieser Parameter ignoriert und die vom DHCP-Server übermittelte Subnetzmaske verwendet.
EnableMulticastDNS	Aktiviert beziehungsweise deaktiviert Multicast DNS (mDNS) - In der Werkseinstellung ist mDNS aktiviert, um auch bei fehlender Netzwerkinfrastruktur den Bus Controller über den Hostnamen ansprechen zu können.

Damit neue Konfigurationsdaten gespeichert werden, muss die Methode *Root/Objects/DeviceSet/X20BC008T/Configuration/Control/ApplyChanges* aufgerufen werden.

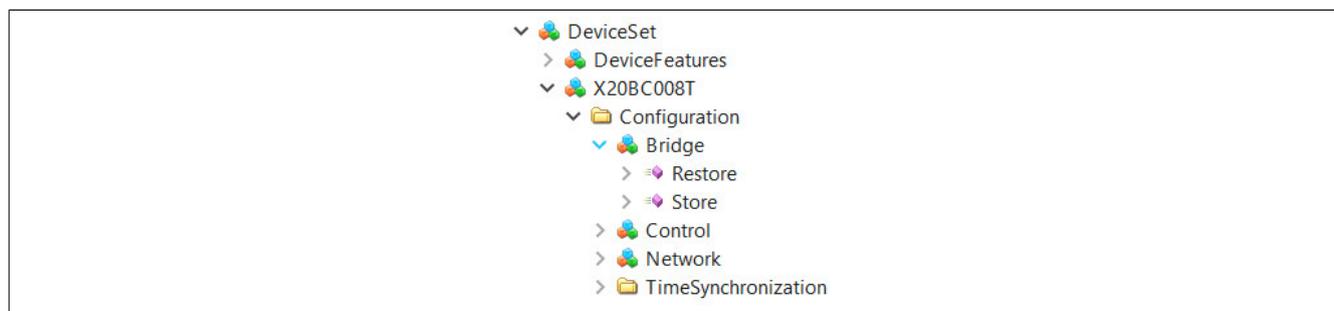
### Information:

Die Netzwerkkonfiguration wird erst beim Neustart des Bus Controllers übernommen.

## 6.1.3 Bridge-Konfiguration

### 6.1.3.1 Methoden für Bridge-Konfiguration

Im OPC UA Informationsmodell werden 2 Methoden bereitgestellt, um die aktuelle Bridge-Konfiguration zu persistieren bzw. diese auf Werkseinstellungen zurückzusetzen.



#### 6.1.3.2 Restore

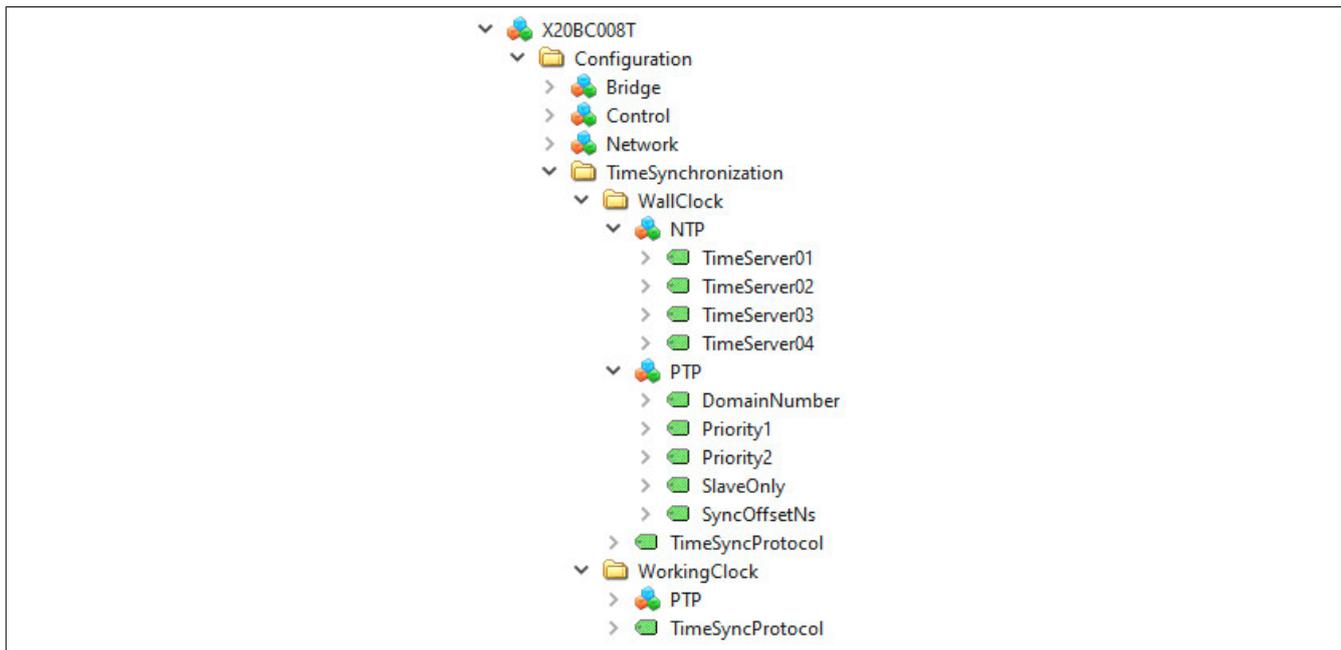
Die aktuelle Bridge-Konfiguration wird gelöscht und auf Werkseinstellungen zurückgesetzt. Nach dem Ausführen dieser Methode ist ein Neustart des Geräts erforderlich.

#### 6.1.3.3 Store

Die aktuelle Bridge-Konfiguration wird am Gerät persistiert.

## 6.1.4 Zeitsynchronisation

Die verwendeten Protokolle zur Zeitsynchronisation müssen für die beiden Zeitdomänen *WallClock* und *WorkingClock* getrennt konfiguriert werden.



Position der Daten im Informationsmodell: *Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization*

### Information:

Die Parameter für die Zeitsynchronisation werden erst durch den Aufruf der Methode *ApplyChanges* übernommen.

#### 6.1.4.1 NTP

Der NTP-Client kann für die *WallClock* aktiviert werden, indem der Konfigurationsparameter *TimeSyncProtocol* auf den Wert "2 (NTP)" eingestellt wird. Es können bis zu 4 Zeitserver angegeben werden. Optional ist es möglich, dass NTP-Server die Adressen durch den DHCP-Server zugewiesen bekommen. Wenn mehrere Zeitserver zur Verfügung stehen (entweder durch Konfiguration oder vom DHCP-Server bezogen), wird davon einer ausgewählt, der für die Zeitsynchronisation verwendet wird. Die anderen stehen als Redundanz zu Verfügung und werden verwendet, falls der aktuell aktive Zeitserver ausfällt.

Knotenname	Beschreibung
TimeServer0x	URL oder IP-Adresse von bis zu 4 Zeitservern, die manuell konfiguriert werden können. Bei Konfiguration von mehreren Zeitservern ist die Auswahl-Reihenfolge der Zeitserver nicht festgelegt.

#### 6.1.4.2 PTP

### Information:

Die PTP-Konfiguration über OPC UA wird in der Version 1.3.1 noch nicht unterstützt.

#### 6.1.4.3 TimeSyncProtocol

Die Konfigurationsparameter für NTP bzw. PTP sind nur dann gültig, wenn das entsprechende Synchronisationsprotokoll ausgewählt wurde.

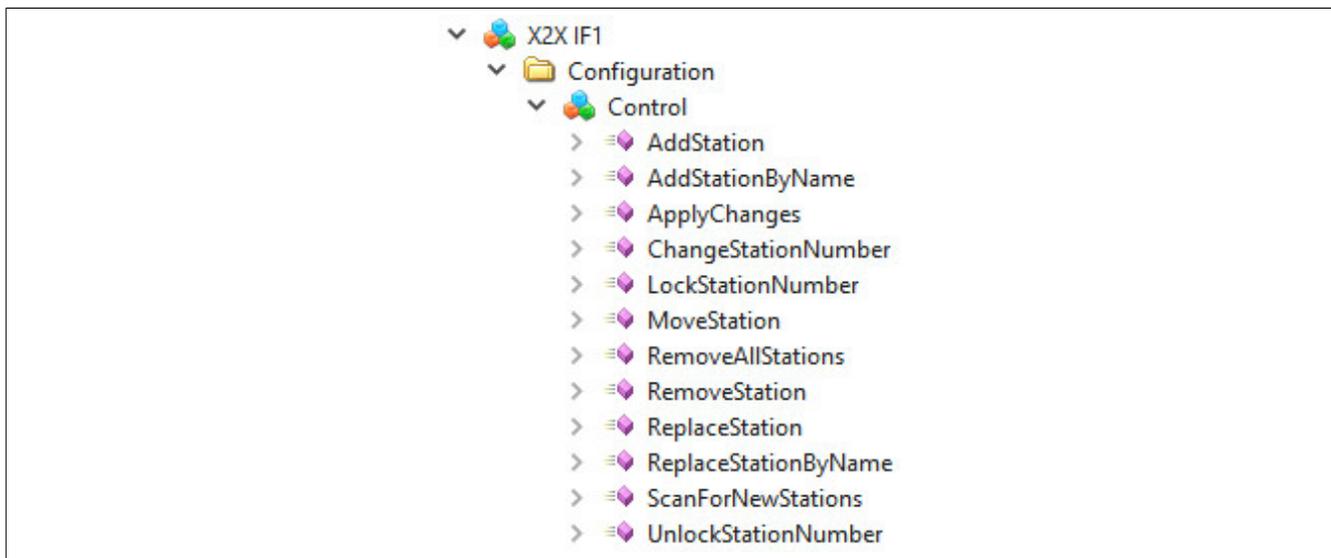
Knotenname	Beschreibung
TimeSyncProtocol	Über diesen Parameter kann die Synchronisation der jeweiligen Uhr aktiviert bzw. das Synchronisationsprotokoll ausgewählt werden. Mögliche Werte: 0 Kein Synchronisationsprotokoll ausgewählt 1 PTP-Protokoll ausgewählt 2 NTP-Protokoll ausgewählt

## 6.2 X2X Link Objekt

Das X2X Link Objekt ist dem Bus Controller Objekt (X20BC008T) untergeordnet und enthält alle Daten, die mit dem X2X Link zusammenhängen. Die verfügbaren Konfigurationsmethoden und Variablen können verwendet werden, um Stationen einzufügen, zu verschieben oder zu Löschen. Des Weiteren können die Stationen über entsprechende Parameter konfiguriert werden.

### 6.2.1 X2X Link Inbetriebnahme

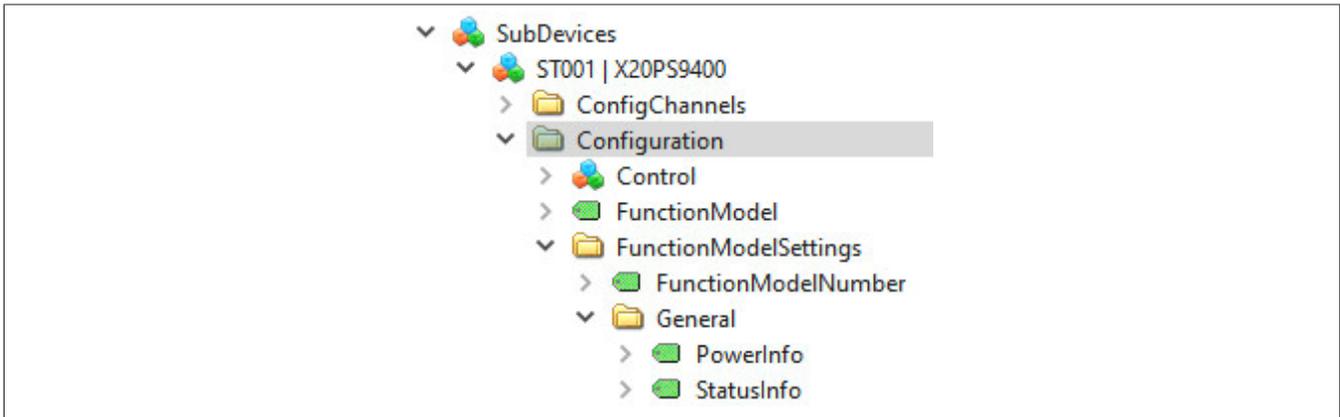
Für die Inbetriebnahme eines noch nicht konfigurierten Bus Controllers müssen zunächst die benötigten Module im OPC UA Modell hinzugefügt werden. Dies wird mit Hilfe der im Abschnitt 6.2.2 "Methoden für X2X Link Konfiguration" beschriebenen Methoden durchgeführt:



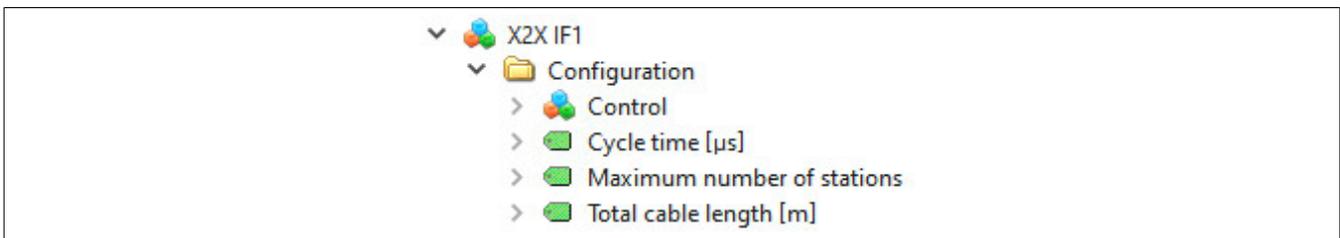
• Mit der Methode "ScanForNewStations" können alle am Bus Controller physikalisch gesteckten Module automatisch eingefügt werden. Anschließend muss das Informationsmodell im Browser aktualisiert werden, um die Stationen unter "SubDevices" anzuzeigen.



- Je nach Bedarf erfolgt nun die Konfiguration der Module im OPC UA Modell. Wird diese nicht verändert, kommt die voreingestellte Konfiguration zur Anwendung.



Die globalen X2X Link Parameter sind im Ordner "Configuration" unterhalb des Schnittstellen-Knotens zu finden:



- Mit der Methode [6.2.2.10 "ApplyChanges"](#) wird die Konfiguration gespeichert und am X2X Link übernommen. Diese gespeicherte Konfiguration wird bei jedem Neustart geladen und automatisch gestartet.

Bei einer nachträglichen Änderung der I/O-Konfiguration der Module oder der X2X Link Parameter, muss die Änderung mit "ApplyChanges" erneut übernommen werden.

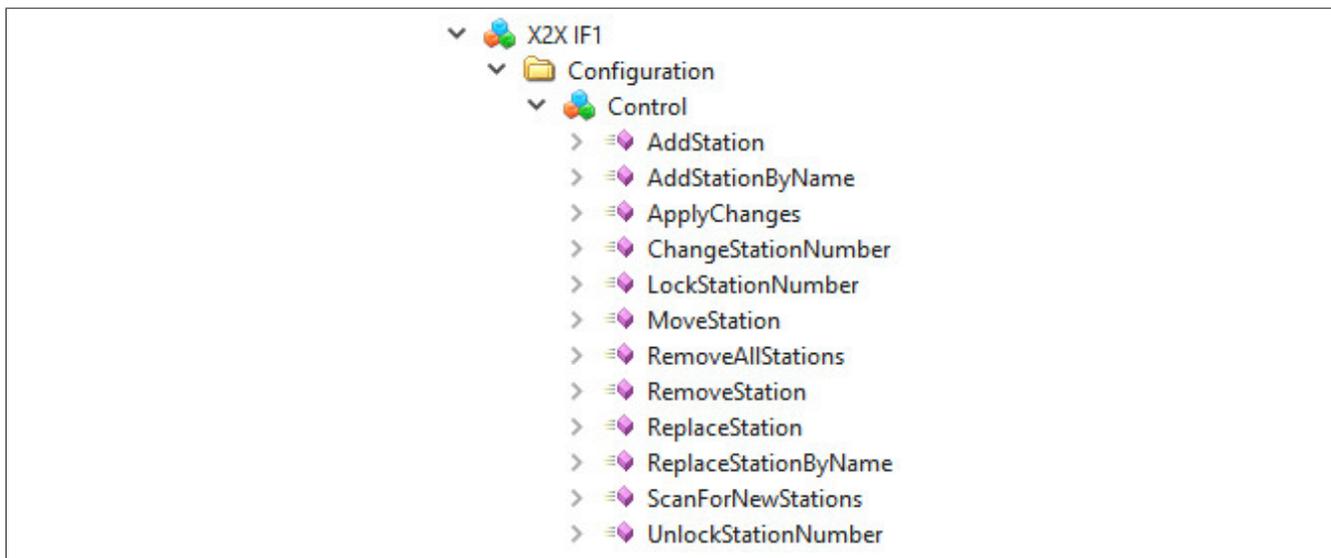
### Information:

Die Methode "ApplyChanges" in diesem Objekt bezieht sich nur auf die X2X-Konfiguration und die Konfiguration der darunter liegenden I/O-Module. Andere Konfigurationswerte, wie zum Beispiel die Netzwerkkonfiguration werden damit nicht übernommen.

## 6.2.2 Methoden für X2X Link Konfiguration

Der physikalischen X2X Link Schnittstelle wird im OPC UA Informationsmodell eine Reihe von Methoden zum Steuern und Konfigurieren des X2X Links angeboten. Der Name (bzw. NodeId) der Schnittstelle entspricht der Schnittstellenadresse wie auch im Automation Studio üblich ("IF1").

Die Methoden von IF1 sind im Knoten "IF1@Control" zu finden und haben die NodeId "IF1.Control@<MethodName>":



### Information:

Modellmanipulationen und Konfigurationsparameter der Stationen werden erst übernommen und permanent gespeichert, wenn [6.2.2.10 "ApplyChanges"](#) aufgerufen wird.

### 6.2.2.1 AddStation / AddStationByName

Hinzufügen neuer Stationen ins OPC UA Informationsmodell.

Nachfolgende Module werden nicht verschoben. Ist die Stationsnummer nicht frei, wird ein Fehler gemeldet.

#### AddStation

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer (0 = nach letztem Modul anhängen)
ModuleId	X2XAvailableModules	ModuleID des zu erzeugenden Moduls

#### AddStationByName

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer (0 = nach letztem Modul anhängen)
ModuleName	String	Modulname des zu erzeugenden Moduls

### 6.2.2.2 ChangeStationNumber

Ändern der Stationsnummer einer Stationsgruppe.

Verschiebt alle Stationen einer **Stationsgruppe** an eine neue Position. Damit kann z. B. die Konfiguration an einen geänderten Knotennummerschalter angepasst werden.

Dabei gelten folgende Regeln:

- Die Zielposition kann an beliebiger Stelle vor oder nach der aktuellen Position sein. Andere Stationsgruppen dürfen übersprungen werden.
- Die unter *Destination* angegebene Zielposition muss noch frei sein, es sei denn sie befindet sich innerhalb der zu verschiebenden Stationsgruppe.
- Es muss genügend Platz für alle nachfolgenden Stationen der Stationsgruppe vorhanden sein; das heißt, es dürfen sich keine Module innerhalb des Zielbereichs befinden.

Im Fehlerfall wird die Methode nicht durchgeführt.

Nach dem Verschieben wird die Stationsnummer des ersten Moduls automatisch als gesperrt markiert. Dadurch wird das Modul beim Verschieben anderer Gruppen nicht mehr mitverschoben. Siehe dazu 6.2.2.3 "LockStationNumber".

#### Beispiel

Ausgangszustand	Verschieben: Station 16 nach Station 17	Verschieben: Station 16 nach Station 64
<pre> SubDevices ├── ST001   X20PS9400 ├── ST002   X20AI8321 ├── ST004   X20DI8371 ├── ST016   X20DO8322 (locked station number) ├── ST017   X20DO8322 ├── ST018   X20DO8322 ├── ST032   X20AI8221 (locked station number) ├── ST033   X20AO2622 └── SupportedTypes           </pre>	<pre> SubDevices ├── ST001   X20PS9400 ├── ST002   X20AI8321 ├── ST004   X20DI8371 ├── ST017   X20DO8322 (locked station number) ├── ST018   X20DO8322 ├── ST019   X20DO8322 ├── ST032   X20AI8221 (locked station number) ├── ST033   X20AO2622 └── SupportedTypes           </pre>	<pre> SubDevices ├── ST001   X20PS9400 ├── ST002   X20AI8321 ├── ST004   X20DI8371 ├── ST032   X20AI8221 (locked station number) ├── ST033   X20AO2622 ├── ST064   X20DO8322 (locked station number) ├── ST065   X20DO8322 ├── ST066   X20DO8322 └── SupportedTypes           </pre>

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer
Destination	UInt32	Stationsnummer der Zielposition

### 6.2.2.3 LockStationNumber

Sperren einer Stationsnummer.

Damit können zusammenhängende Stationsgruppen definiert werden. Eine Stationsgruppe enthält alle Stationen ab der gesperrten Stationsnummer bis exclusive der nächsten gesperrten Stationsnummer (Siehe 1 im Bild unten). Definierte Stationsgruppen können durch Aufruf der Methode 6.2.2.2 "ChangeStationNumber" unabhängig voneinander verschoben werden.

#### Beispiel mehrerer Stationsgruppen

```

SubDevices
├── ST001 | X20PS9400
├── ST002 | X20AI8321
├── ST004 | X20DI8371
├── ST016 | X20DO8322 (locked station number)
├── ST017 | X20DO8322
├── ST018 | X20DO8322
├── ST032 | X20AI8221 (locked station number)
├── ST033 | X20AO2622
└── SupportedTypes
          
```

Der Sperrstatus eine Stationsnummer kann mittels Eigenschaft *LockedStationNumber* der jeweiligen Station ausgelesen werden.

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer

#### 6.2.2.4 MoveStation

Verschieben einer Station im OPC UA Informationsmodell auf eine andere Position.

Ist die Stationsnummer nicht frei, wird ein Fehler gemeldet.

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer
Destination	UInt32	Stationsnummer der Zielposition

#### Information:

Stationen mit gesperrter Stationsnummer können nicht verschoben werden. In diesem Fall ist die Methode 6.2.2.2 "ChangeStationNumber" zu verwenden.

#### 6.2.2.5 RemoveAllStations

Entfernen aller Stationen vom OPC UA Informationsmodell.

#### 6.2.2.6 RemoveStation

Entfernen einer Station vom OPC UA Informationsmodell.

Nachfolgende Module werden nicht verschoben.

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer

#### 6.2.2.7 ReplaceStation / ReplaceStationByName

Ersetzen einer Station im OPC UA Informationsmodell durch ein anderes Modul.

Existiert auf der Stationsnummer bereits ein Modul mit dem gleichen Namen, erfolgt keine Änderung. Ändert sich der Name, wird das Modul mit dem neuen ersetzt. Ansonsten wird das Modul neu angelegt.

Das geänderte Modul ist sofort im OPC UA Informationsmodell enthalten und kann mittels OPC UA Client weiter konfiguriert werden.

#### ReplaceStation

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer (0 = nach letztem Modul anhängen)
ModuleID	X2XAvailableModules	ModuleID des zu erzeugenden Moduls

#### ReplaceStationByName

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer (0 = nach letztem Modul anhängen)
ModuleName	String	Modulname des zu erzeugenden Moduls

#### 6.2.2.8 ScanForNewStations

Scannen des X2X Links auf neue Stationen.

Zum Scannen des X2X Links muss dieser zuvor vollständig hochgefahren und bereit sein. Ansonsten wird ein Fehler gemeldet.

Diese Methode überprüft, ob neue Stationen am X2X Link gefunden wurden, die noch nicht in der Konfiguration vorhanden sind. Neu gefundene Stationen werden dabei automatisch dem OPC UA Informationsmodell hinzugefügt.

Module, welche auf Busmodulen mit Knotennummernschaltern stecken, werden automatisch als gesperrt markiert und mit der entsprechenden Nummer angemeldet.

#### Information:

Wurde der Knotennummernschalter eines Busmoduls geändert, muss diese Änderung zuvor manuell mit der Methode 6.2.2.2 "ChangeStationNumber" durchgeführt werden. Ansonsten werden die Stationen an der neuen Position erneut hinzugefügt.

### 6.2.2.9 UnlockStationNumber

Entsperren einer Stationsnummer und Auflösen einer Stationsgruppe.

Für Details siehe [6.2.2.3 "LockStationNumber"](#).

Eingangsargumente	Datentyp	Beschreibung
Station	UInt32	Stationsnummer

### 6.2.2.10 ApplyChanges

Anwenden einer neuen oder geänderten Konfiguration.

Durchgeführte Konfigurationsänderungen werden permanent gespeichert und bei einem Neustart wieder geladen. Nach diesem Schritt sind die, je nach Konfiguration der Module, verfügbaren Kanäle im Ordner "ProcessData" mittels OPC UA Client sichtbar.

Wurde die Konfiguration fehlerfrei übernommen, wird sie anschließend am X2X Link gestartet. Tritt beim Prüfen und Speichern der Konfiguration ein Fehler auf, wird dieser gemeldet und die alte Konfiguration bleibt erhalten.

## 6.3 PubSub-Konfiguration

Die OPC UA Publisher-Subscriber-Konfiguration kann durch Übertragen einer "\*.uabinary"-Datei durchgeführt werden. Dieses Datei enthält alle Informationen die für den zyklischen Datenaustausch erforderlich sind. UaExpert verfügt über einen integrierten Editor um eine solche Datei zu Erstellen und zu Bearbeiten (Für ein detailliertes Beispiel siehe Abschnitt [6.3.2 "PubSub-Konfigurationsbeispiel"](#)).

Die Datei kann über das Objekt *Root/Objects/Server/PublishSubscribe/PubSubBinary* gelesen bzw. geschrieben werden. Die Konfiguration wird aktiviert, sobald eine neue Datei geschrieben wurde.

Die Datei enthält die folgenden Konfigurationsdaten:

- Publisher-Konfiguration: Gibt an, welche Prozessdaten periodisch versendet werden und mit welcher Periode dies geschieht.
- Subscriber-Konfiguration: Gibt an, welche Prozessdaten eines Publishers empfangen werden und wohin diese Daten weitergegeben werden.
- Allgemeine Verbindungskonfiguration

Aktuell können nur zyklische Prozessdaten von X2X-Modulen in Publisher- und Subscriber-Konfigurationen verwendet werden. Dies sind alle Knoten im Prozessdata-Ordner eines X2X-Moduls außer den Knoten ModuleOk, ModuleId, SerialNumber und FirmwareVersion.

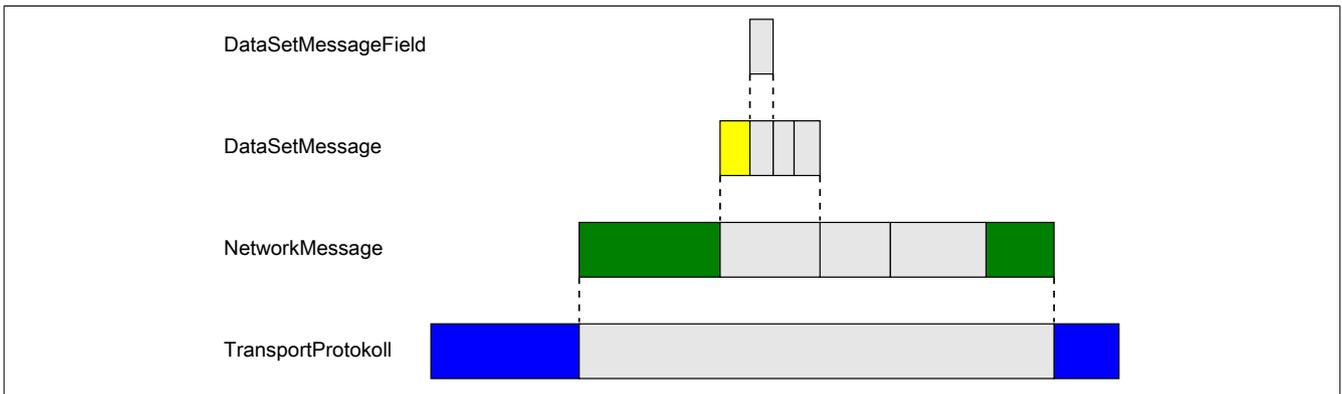
### 6.3.1 Allgemeines

Beim PubSub-Kommunikationsmodell erfolgt die Kommunikation zwischen sogenannten Publishern und Subscribern. Publisher versenden die Nachrichten. Sogenannte Subscriber empfangen die Nachrichten und verarbeiten sie weiter.

Der X20BC008T unterstützt aktuell nur das "UADP-Periodic-Fixed" Header Layout und das Transportprotokoll "opc.udp":

- **UADP-Periodic-Fixed:** Bei diesem Modus werden die Nachrichten ähnlich wie bei POWERLINK periodisch in einem festen Intervall gesendet, unabhängig von Änderungen in den Daten. Die Variablen (Datenpunkte), die innerhalb einer Nachricht übertragen werden, werden zur Konfigurationszeit festgelegt.
- **Opc.udp:** Das Senden der Daten erfolgt typischerweise im Multicast-Netzwerk. Der Empfang der gesendeten Nachrichten wird nicht bestätigt, d. h. es erfolgt keine Überprüfung, ob bzw. bei wie vielen Subscribern eine Nachricht empfangen wird.

### 6.3.1.1 Aufbau einer PubSub Nachricht



- 1) Das **DataSetMessageField** enthält einen einzelnen Variablenwert/Datenpunkt.
- 2) Jede **DataSetMessage** besteht aus einem oder mehreren DataSetMessageFields. Unter Anderem enthält der Header (gelb markiert) der DataSetMessage eine DataSetWriterId und Statusinformationen:
  - *DataSetWriterId:* Identifiziert ein Dataset innerhalb der NetworkMessage. Die DataSetWriterId muss daher innerhalb der NetworkMessage eindeutig sein.
  - *Statusinformation:* Geben Auskunft über die Gültigkeit der enthaltenen DataSetMessageFields.

#### Information:

Beim X20BC008T wird in der Defaultkonfiguration eine DataSetMessage pro I/O-Modul angelegt.

- 3) Innerhalb einer **NetworkMessage** befinden sich eine oder mehrere DataSetMessages. Unter Anderem enthält der Header (grün markiert) der NetworkMessage verschiedene IDs und GroupVersion:
  - *WriterGroupId:* Identifiziert verschiedene Nachrichten desselben Publishers und muss daher innerhalb des Publishers eindeutig sein.
  - *PublisherId:* Identifiziert den Publisher und muss daher innerhalb des Netzwerks eindeutig sein.
  - *GroupVersion:* Identifiziert die Version einer Nachricht. Die Version wird automatisch geändert, wenn eine Nachricht umkonfiguriert wird.
- 4) Über das **Transportprotokoll** wird die NetworkMessage übertragen.

#### Information:

Damit ein Subscriber Nachrichten akzeptiert und verarbeitet, müssen in der jeweiligen Reader(Subscriber)-Konfiguration alle 4 IDs (DataSetWriterId, WriterGroupId, PublisherId und GroupVersion) übereinstimmen.

## 6.3.2 PubSub-Konfigurationsbeispiel

### Voraussetzungen

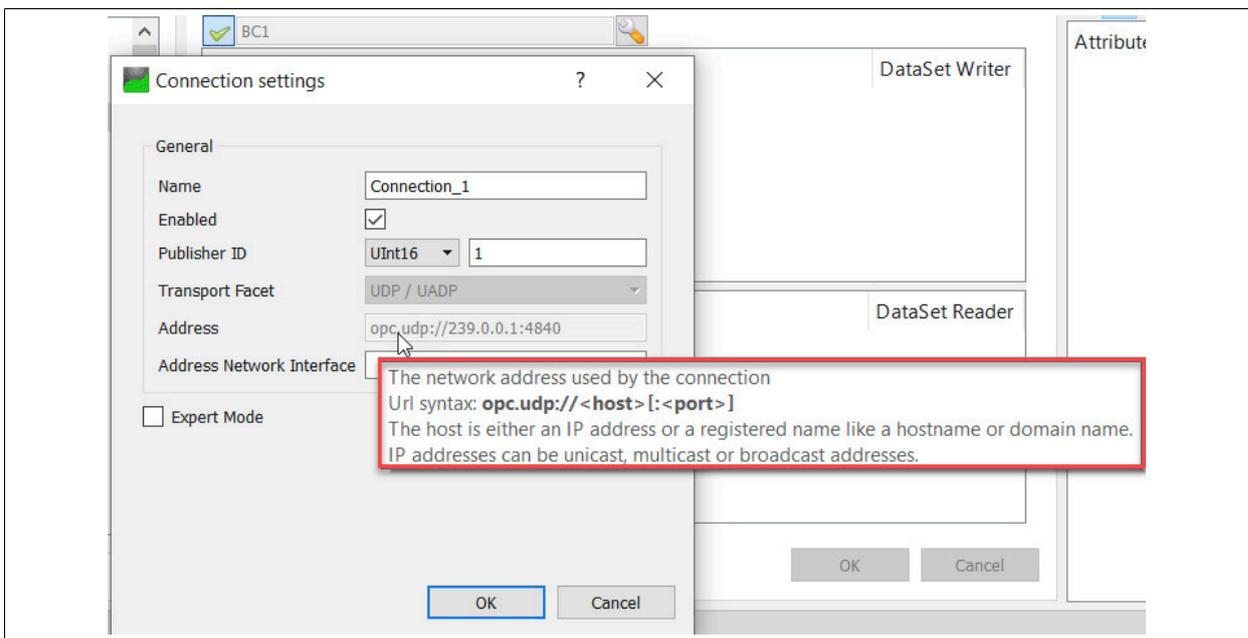
Um eine PubSub-Konfiguration mit UaExpert durchführen zu können, werden folgende Bedingungen vorausgesetzt:

- Eine Verbindung zum Bus Controller mit UaExpert wurde aufgebaut.
- Die am X2X Link vorhandenen Module wurden bereits konfiguriert (siehe 6.2.1 "X2X Link Inbetriebnahme").
- Alle Module zeigen eine konstant grün leuchtende Status-LED an.

### Information:

Die Dokumentation zur PubSub Konfiguration im UaExpert ist unter "Help - UaExpert Manual - Index - PubSub Config View" zu finden.

Die Bedeutung eines Konfigurationsfelds im UaExpert kann mit der Maus angezeigt werden, indem der Mauszeiger darüber platziert wird.

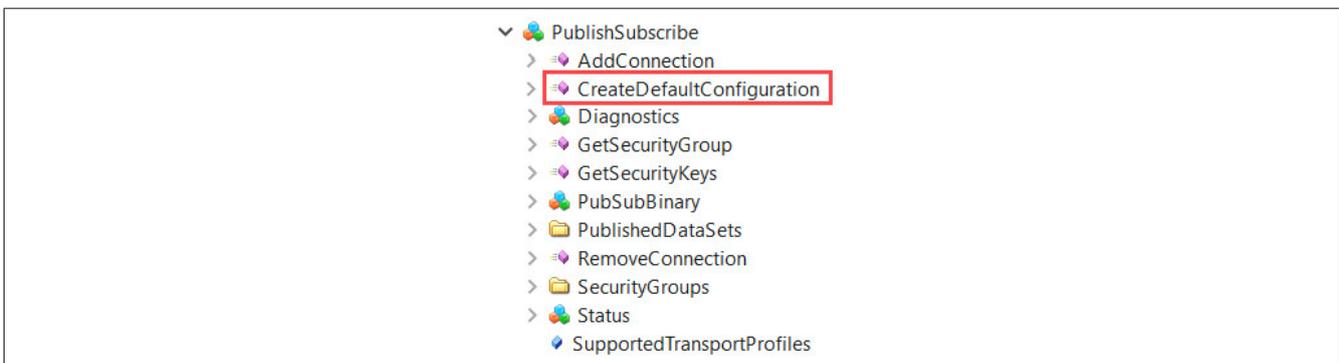


### 6.3.2.1 Bus Controller als Publisher mit UaExpert konfigurieren

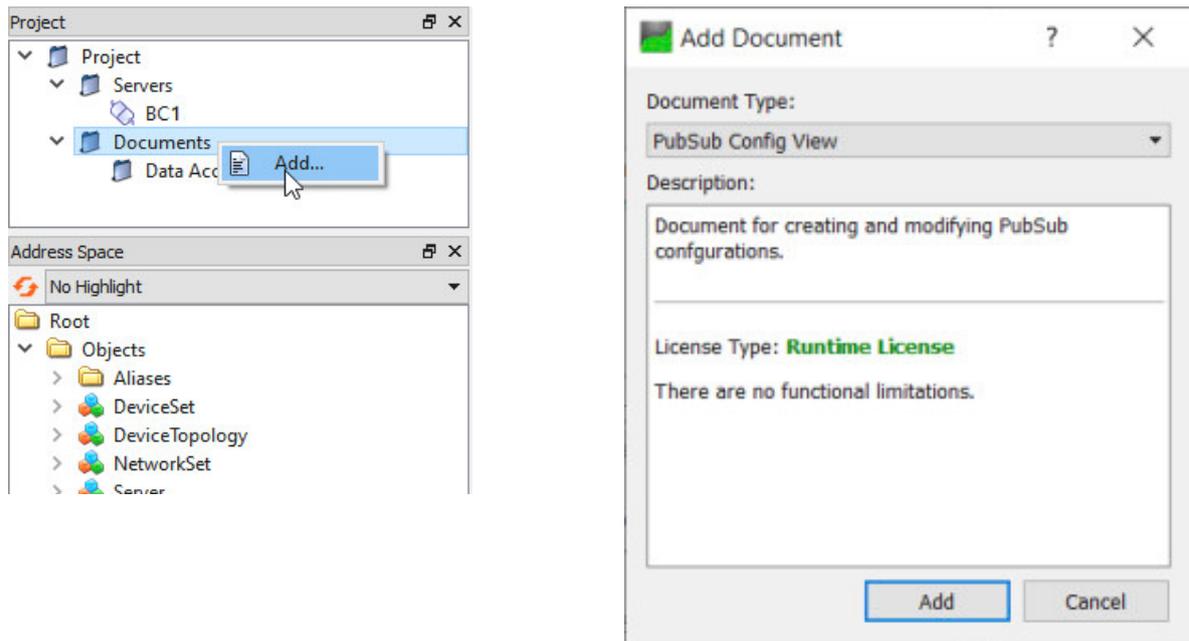
#### Default Publisher erzeugen und abrufen

- Beim Bus Controller unter *Root/Objects/Server/PublishSubscribe* die Methode *CreateDefaultConfiguration* aufrufen.

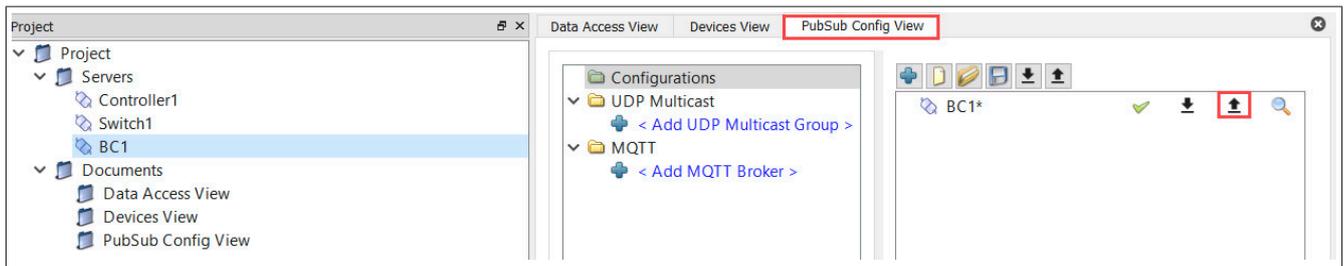
Damit wird eine Default-Konfiguration für PubSub erzeugt, welche den Konfigurationsaufwand wesentlich reduziert und vereinfacht. Sie enthält gültige Werte für alle wesentlichen Parameter und alle Eingangsprozessdaten der konfigurierten IO-Module werden bereits zur Publisher-Konfiguration hinzugefügt.



- Im Projektfenster des UaExpert im Kontextmenü von *Documents* auf *Add...* klicken. Ein Dialog öffnet sich. In diesem Dialog den Dokument-Typ *PubSub Config View* auswählen und durch Klick auf *Add* bestätigen.



- Generierte Default-Publisher-Konfigurationsdatei hochladen. Dies erfolgt durch Auswahl des Ordners *Configurations* unter *PubSub Config View* und Klick auf den Upload-Pfeil.



## Parameter anpassen

Es sollten bei Bedarf nur die hier beschriebenen Parameter angepasst werden.

- Destination IP-Adresse

Die Destination-IP Adresse wird bei der Netzwerkplanung festgelegt und ist Teil einer UDP Multicast Gruppe im UaExpert. Publisher und deren Subscriber müssen sich in der gleichen Multicast Gruppe befinden.

`opc.udp://<Destination-IP>:<Port>`

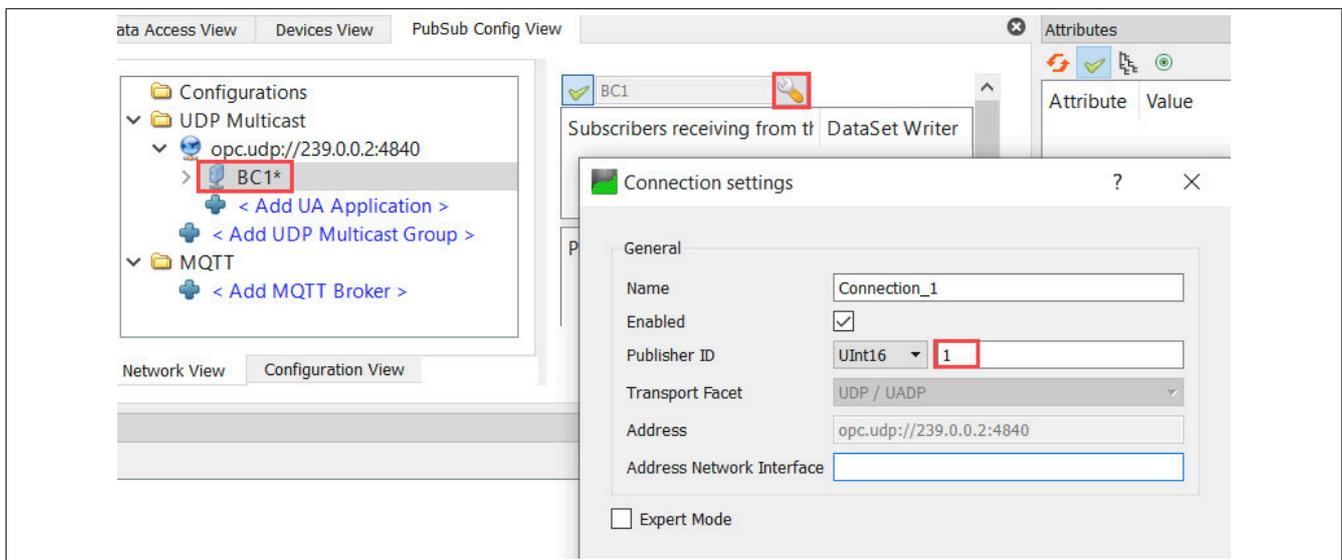


### Information:

Bei Anbindung an einen TSN-Stream ist darauf zu achten, dass eine Multicast-IP Adresse verwendet wird, die zur Multicast-DMAC (Destination MAC-Adresse) passt.

- PublisherId

Die PublisherId muss im Netzwerk eindeutig sein, darf also für einen Publisher nur einmal existieren.



### Information:

Die PublisherId des Default-Publishers wird von den letzten beiden Stellen der Bus Controller IP-Adresse abgeleitet und ist somit im Netzwerk für kleine Netzwerke eindeutig. Sie muss also nicht zwingend verändert werden.

- Publishing Interval

Das Publishing Interval (in ms) legt die Periode fest, mit der das DataSet aktualisiert, das heißt, versendet wird und ist innerhalb einer *WriterGroup* definiert. Damit wirkt sich eine Änderung im *DataSetWriter* auch auf andere *DataSetWriter* einer *WriterGroup* aus. Diese können nicht einzeln angepasst werden.

The screenshot displays the 'PubSub Config View' of the software. On the left, a tree view shows the configuration hierarchy: Configurations > UDP Multicast > opc.udp://239.0.0.2:4840 > BC1\* > IF1.ST1, IF1.ST2, IF1.ST3. The 'IF1.ST2' node is highlighted with a red box. In the center, the 'DataSet Writer Settings' dialog box is open, showing the 'DataSet Writer' tab. The 'Group selection' dropdown is set to 'WriterGroup\_1 | 2ms | UADP-Periodic-Fixed'. The 'General' section contains the following settings: Name: WriterGroup\_1, Enabled: checked, Security Mode: None, Max Network Message Size: 1400, Writer Group ID: 1, Publishing Interval: 2,00 (highlighted with a red box), Keep Alive Time: 6,00, Header Layout: UADP-Periodic-Fixed. The 'Expert Mode' checkbox is unchecked. In the background, a table titled 'Published DataSet' is visible, listing fields like IF1.ST1 through IF1.ST6 and their corresponding data types and published variable node IDs.

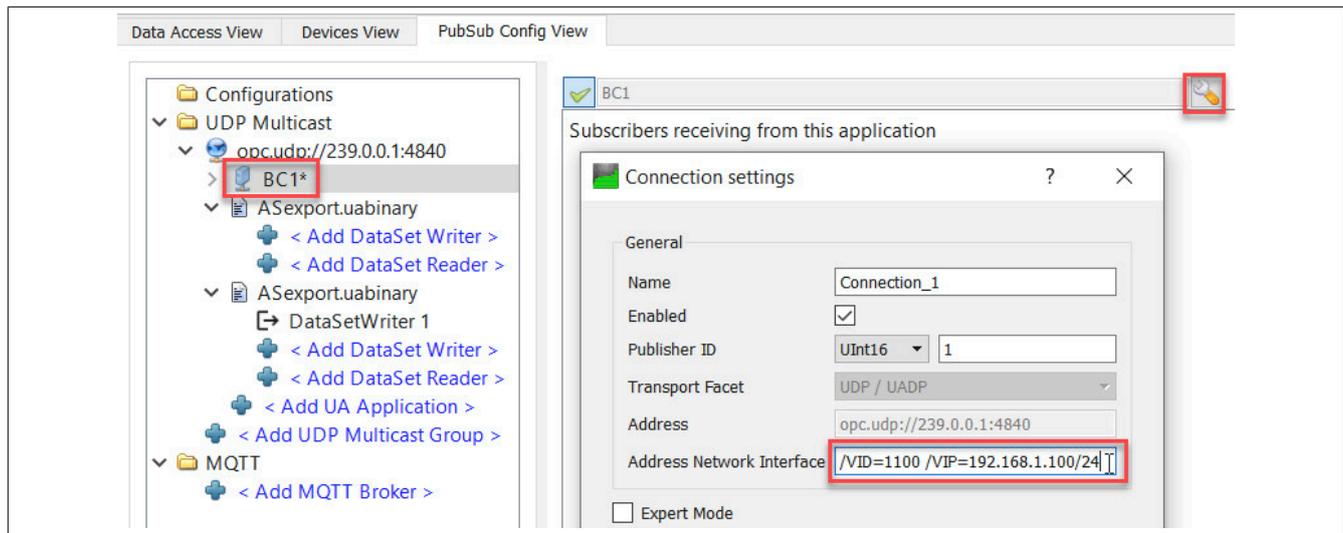
- VLAN-Parameter (optional)

Diese Parameter werden im *Address Network Interface* Feld in den *Connection Settings* festgelegt. Sie werden im Publisher-Kontext verwendet, um einen Publisher als Talker an einen TSN-Stream anzubinden, oder das Publisher-Datagramm mit einem VLAN-Tag zu versehen.

Folgende VLAN-Parameter können verwendet werden:

- VLAN ID: "/VID=<VLAN ID>"
- PCP (Priority Code Point): "/VP=<PCP>"
- Source IP<sup>1)</sup>: "/VIP=<Source IP>"

- 1) Optionaler Parameter; Falls angegeben, wird dieser Parameter als Source-IP-Adresse für das Publisher-Datagramm verwendet. Falls nicht angegeben, wird die Bus Controller Host-IP-Adresse verwendet. Die Subnetzmaske ist ebenfalls optional und entspricht dem Wert "32", sofern nicht angegeben. 2 Varianten dieses Parameters sind möglich: "/VIP=<Source IP>" oder "/VIP=<Source IP>/<subnet mask>".

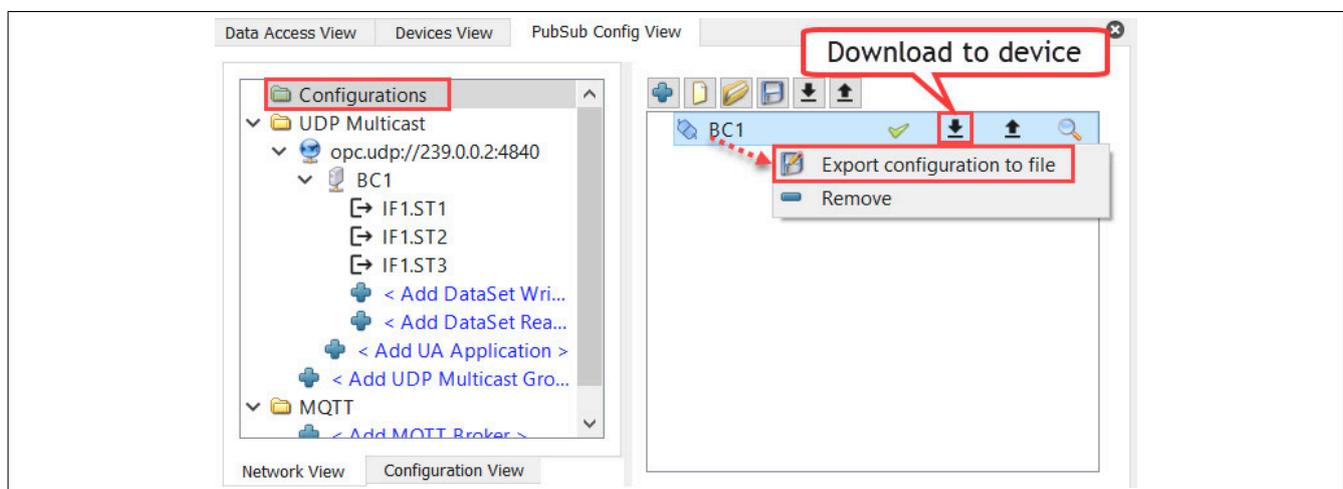


### Konfiguration auf den Bus Controller übertragen

- In der *PubSub Config View* den Ordner *Configurations* auswählen und auf den Download-Pfeil klicken. Zusätzlich soll mittels rechten Mausklick auf den Bus Controller - *Export configuration to file* eine lokale Sicherungskopie gespeichert werden.

### Information:

Die Sicherungskopie dient nicht nur der Sicherung, sondern auch als Referenzdatei für die Subscriber-Konfiguration eines anderen Servers und kann z. B. in Automation Studio importiert werden.

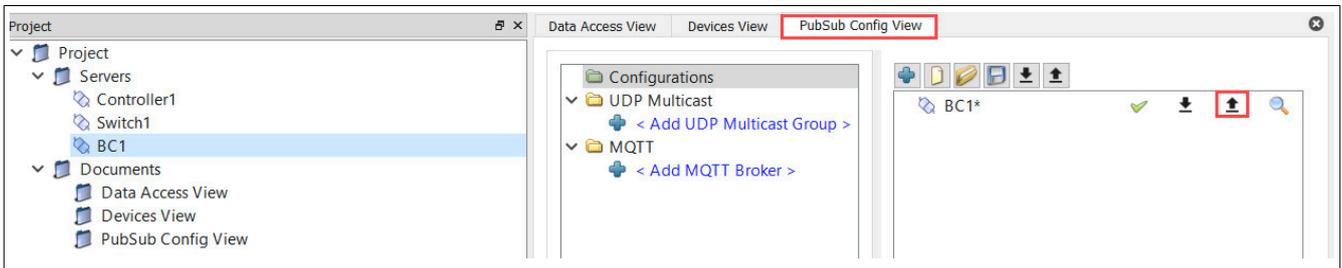


Der Bus Controller versendet nach erfolgter Übertragung der Konfiguration periodisch Publisher-Datagramme.

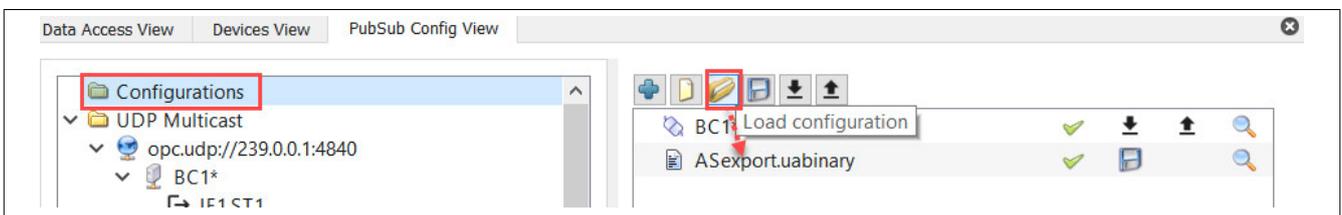
### 6.3.2.2 Bus Controller als Subscriber mit UaExpert konfigurieren

#### Referenzieren eines Publishers für die Subscriber-Konfiguration

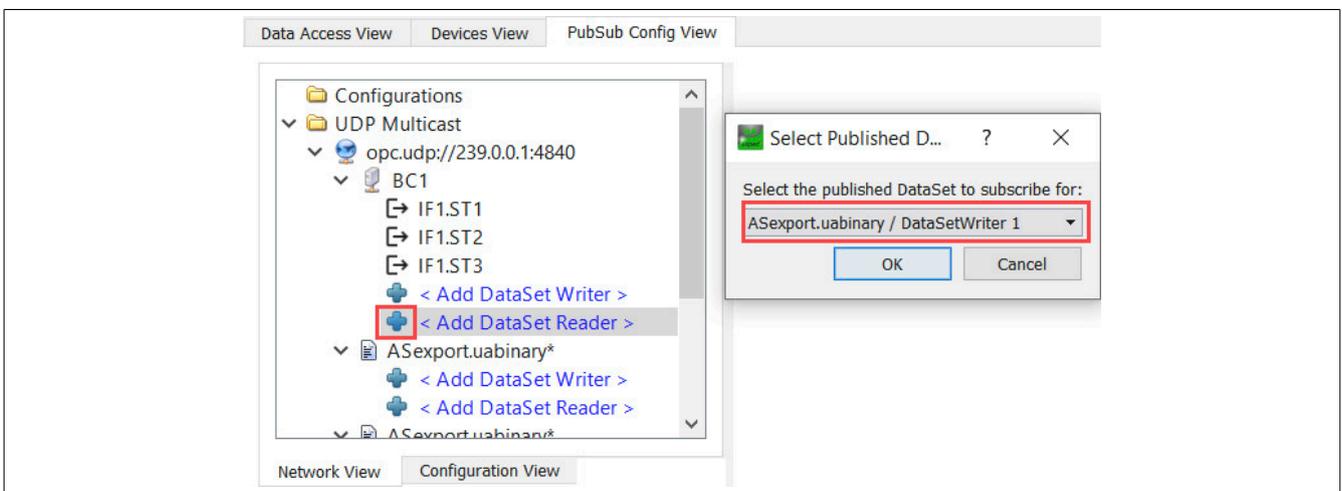
- Falls auf dem Bus Controller bereits eine Konfiguration existiert, muss diese zuvor geladen werden, da diese ansonsten beim nächsten Konfigurationstransfer leer überschrieben wird. Dazu in der *PubSub Config View* den Ordner *Configurations* auswählen und auf den Upload-Pfeil klicken.



- Um die Konfiguration zu laden, Ordner *Configurations* auswählen. Lokale "\*.uabinary"-Datei auswählen bzw. durch Klick auf den Upload-Pfeil die Konfiguration von einem anderen Server laden.

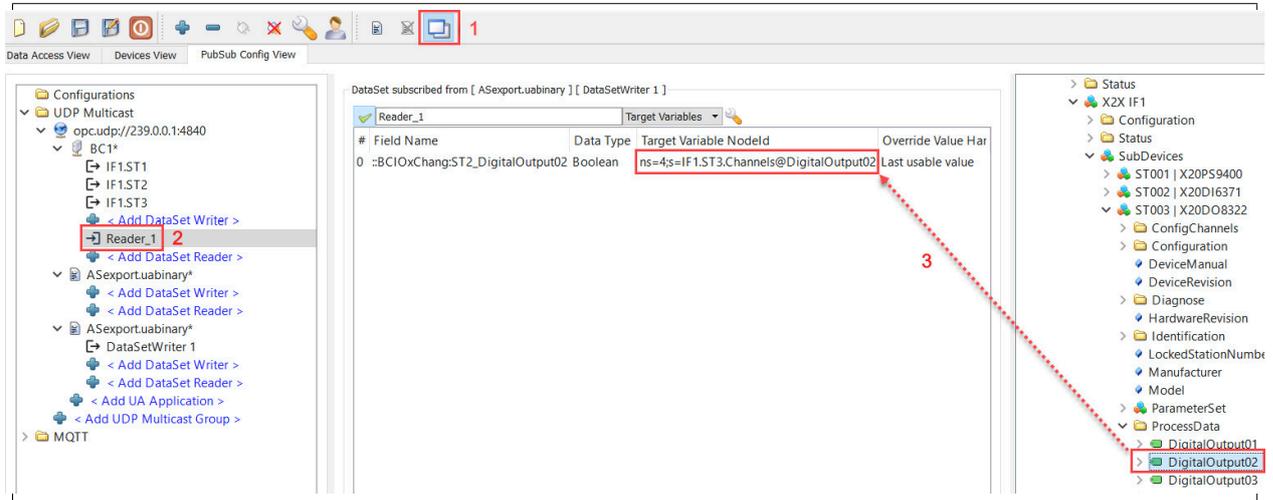


- Durch Klick auf "+" einen neuen *DataSetReader* für Bus Controller Verbindung anlegen und im Dialog "Select Published DataSet" einen "DataSetWriter" als Referenz auswählen.



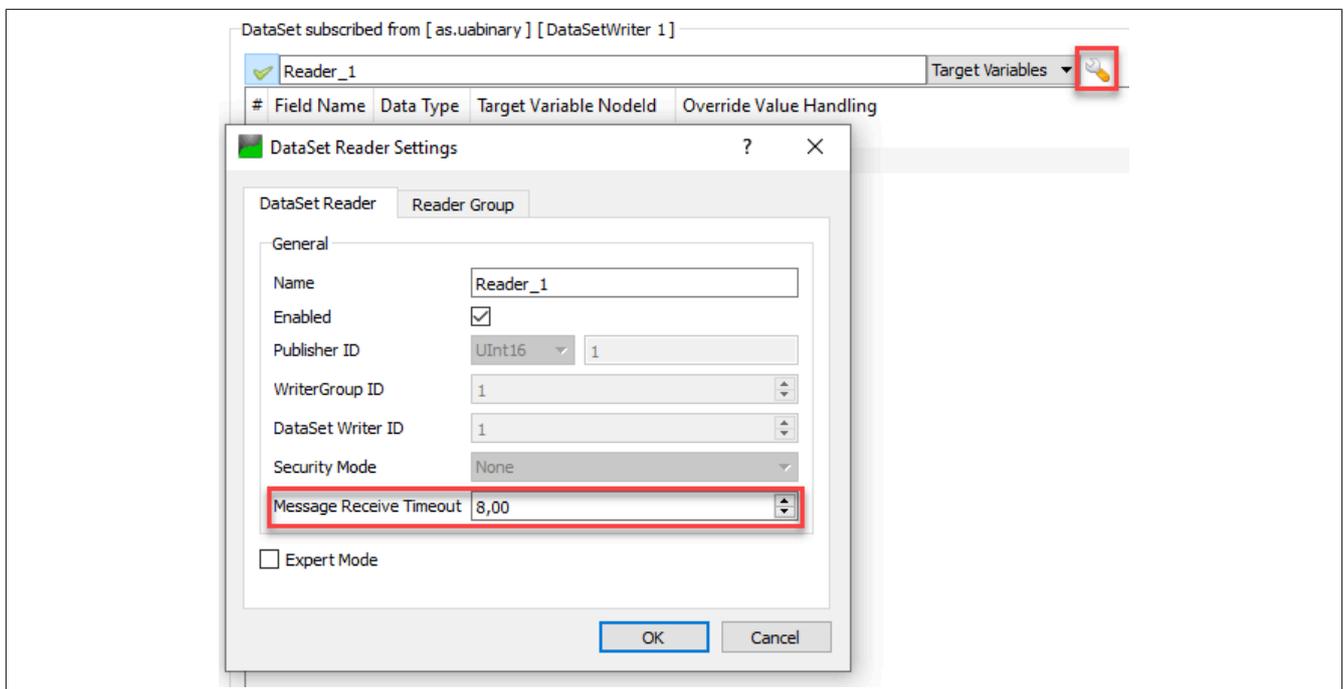
- Um die Modul-Datenpunkte mit Publisher-Daten zu verknüpfen, folgende Schritte durchführen:

- 1) Ansicht durch Klick auf "Hide all dock widgets" wechseln.
- 2) DataSetReader auswählen. Dies fügt die "Address Space View" des zugehörigen Servers zur Ansicht hinzu.
- 3) Zum Knoten navigieren, der mit den Publisher-Daten verknüpft werden soll, z. B. *Root/Objects/Device-Set/X20BC008T/X2X IF1/SubDevices/ST003/X20DO8322/ProcessData/DigitalOutput02* und Knoten auf die gewünschte Target Variable Nodeld ziehen. In diesem Beispiel lässt sich nur ein Datenpunkt zuweisen. Ansonsten evtl. weitere Knoten nacheinander zuweisen.



- Message Receive Timeout

Für jeden DataSet Reader kann in der jeweiligen Konfiguration der *Message Receive Timeout* eingestellt werden. Dies ist wichtig für den Fall, dass die Netzwerkverbindung oder der jeweilige Publisher wegfällt. Nach Ablauf des *Message Receive Timeouts* werden alle Ausgänge zurückgesetzt.



- VLAN-Parameter (optional)

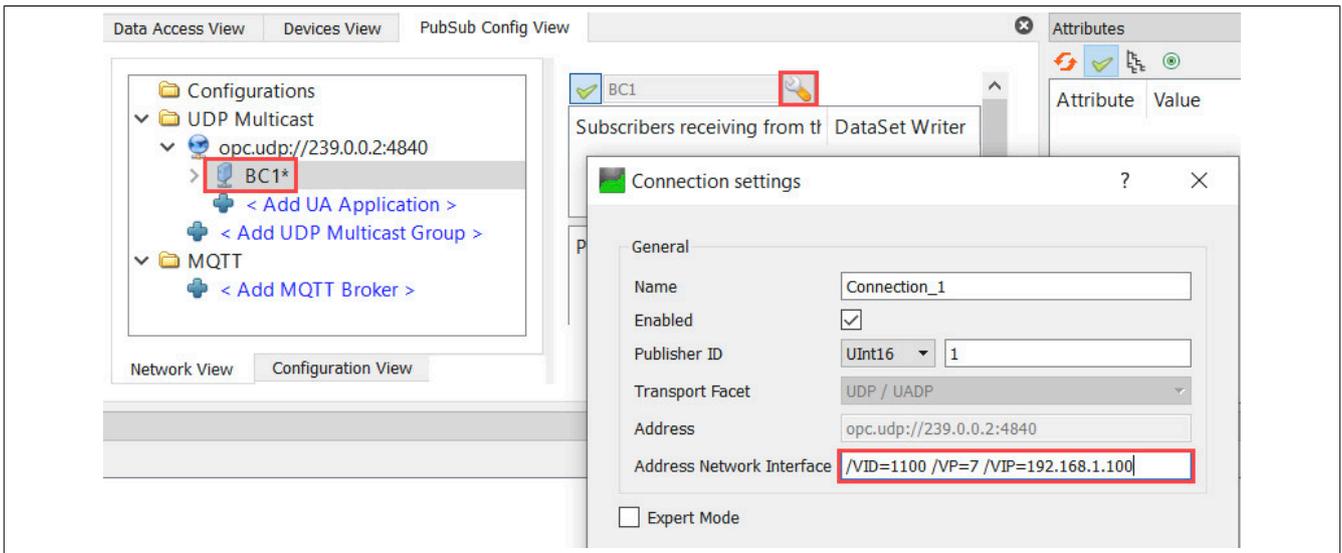
Diese Parameter werden im *Address Network Interface* Feld im "Connection Settings"-Dialog festgelegt. Sie werden im Subscriber-Kontext verwendet, um einen Subscriber als Listener an einen TSN-Stream anzubinden, oder ein Subscriber-Datagramm mit VLAN-Tag zu empfangen.

Folgende VLAN-Parameter können verwendet werden:

- VLAN ID: "/VID=<VLAN ID>"
- PCP (Priority Code Point): "/VP=<PCP>"
- Source IP : "/VIP=<Source IP>"

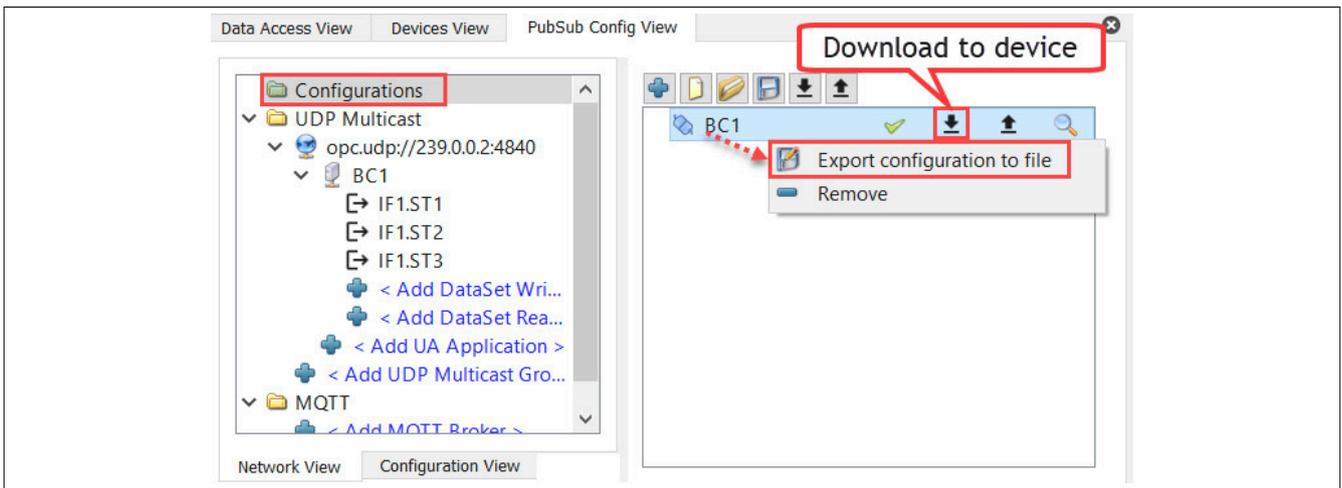
## Information:

Die Parameter */VP* und */VIP* werden nur auf einen evtl. enthaltenen Publisher angewandt.



## Konfiguration auf den Bus Controller übertragen

- In der *PubSub Config View* den Ordner *Configurations* auswählen und auf den Download-Pfeil klicken. Zusätzlich mittels rechten Mausklick auf Bus Controller / "Export configuration to file" eine lokale Sicherungskopie speichern.



Änderungen der Publisher-Daten sollten sich nun auch auf die verknüpften X2X Module auswirken.

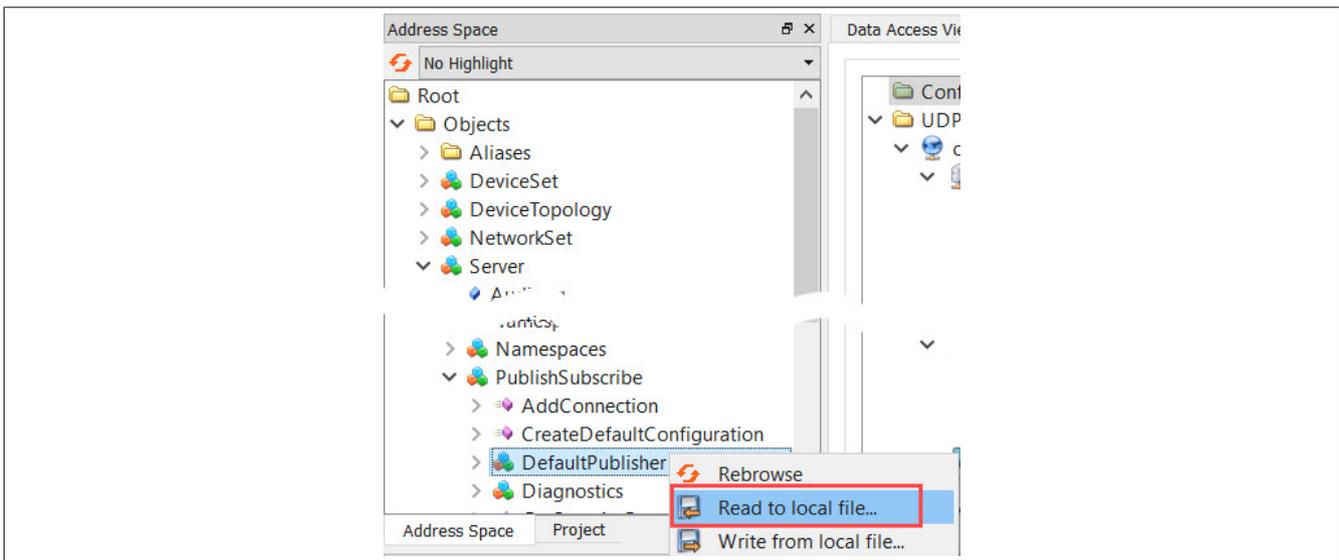
### 6.3.2.3 Kommunikation zwischen Bus Controller und B&R-Steuerung konfigurieren

#### 6.3.2.3.1 Prozessvariablenabbild im Automation Studio

Voraussetzung für dieses Beispiel ist eine Onlineverbindung zwischen Bus Controller und Steuerung, sowie ein bereits angelegtes Automation Studio Projekt.

Um ein Abbild der Modul-Datenpunkte zu erhalten, mit denen kommuniziert werden kann, lassen sich Variablen mit entsprechenden Datentypen für das Automation Studio auf dem Bus Controller erzeugen. Nach erfolgreicher Konfiguration der Module und ausführen der Methode *CreateDefaultConfiguration* kann über das Objekt *Root/Objects/Server/PublishSubscribe/DefaultPublisher* gelesen werden. Darin befindet sich ein Vorschlag für Variablen im Automation Studio, die sich über PubSub mit dem Automation Studio verknüpfen lassen.

- *DefaultPublisher*-Objekt lesen und mit einem Rechtsklick auf *Read to local file ...* als *"\*.zip"*-Datei speichern

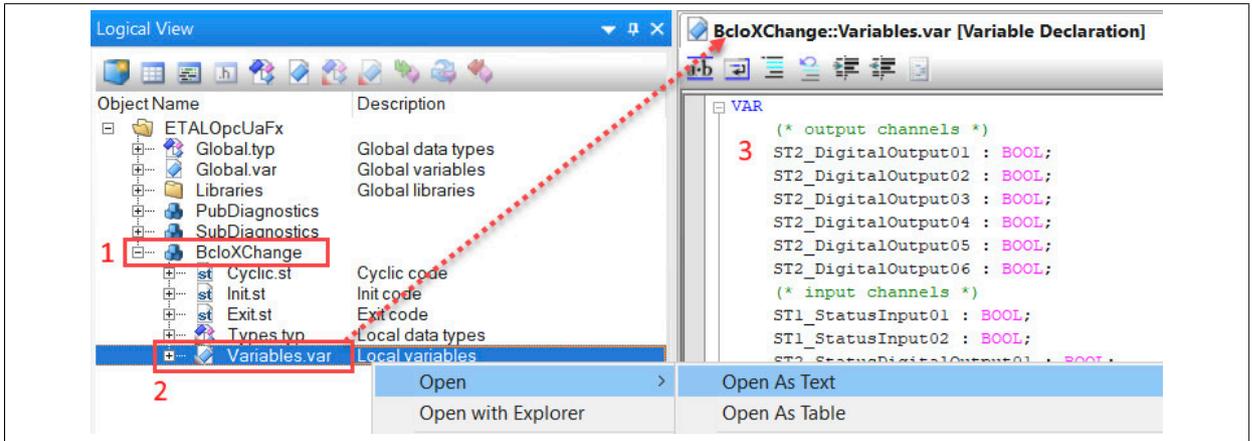


- Gespeicherte *"\*.zip"*-Datei öffnen und die darin enthaltene *"default.var"*-Datei mit Texteditor öffnen.

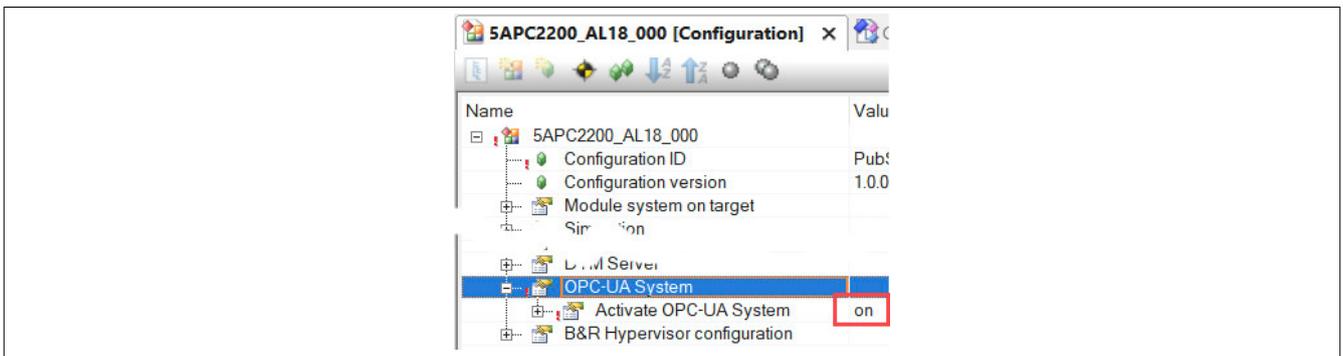
#### Beispiel einer "default.var"-Datei

```
VAR
  (* output channels *)
  ST2_DigitalOutput01 : BOOL;
  ST2_DigitalOutput02 : BOOL;
  ST2_DigitalOutput03 : BOOL;
  ST2_DigitalOutput04 : BOOL;
  ST2_DigitalOutput05 : BOOL;
  ST2_DigitalOutput06 : BOOL;
  (* input channels *)
  ST1_StatusInput01 : BOOL;
  ST1_StatusInput02 : BOOL;
  ST2_StatusDigitalOutput01 : BOOL;
  ST2_StatusDigitalOutput02 : BOOL;
  ST2_StatusDigitalOutput03 : BOOL;
  ST2_StatusDigitalOutput04 : BOOL;
  ST2_StatusDigitalOutput05 : BOOL;
  ST2_StatusDigitalOutput06 : BOOL;
  ST3_AnalogInput01 : INT;
  ST3_AnalogInput02 : INT;
  ST3_StatusInput01 : BYTE;
END_VAR
```

- Um den Inhalt der "default.var"-Datei in Automation Studio einzufügen, folgende Schritte durchführen:
  - 1) Programm in Automation Studio anlegen. (z. B. BcloXchange)
  - 2) Variablendeklaration mit Rechtsklick auf *Open*→*Open as text* öffnen.
  - 3) Den Inhalt der im Texteditor geöffneten Datei "default.var" kopieren und in der Variablendeklaration einfügen und speichern.

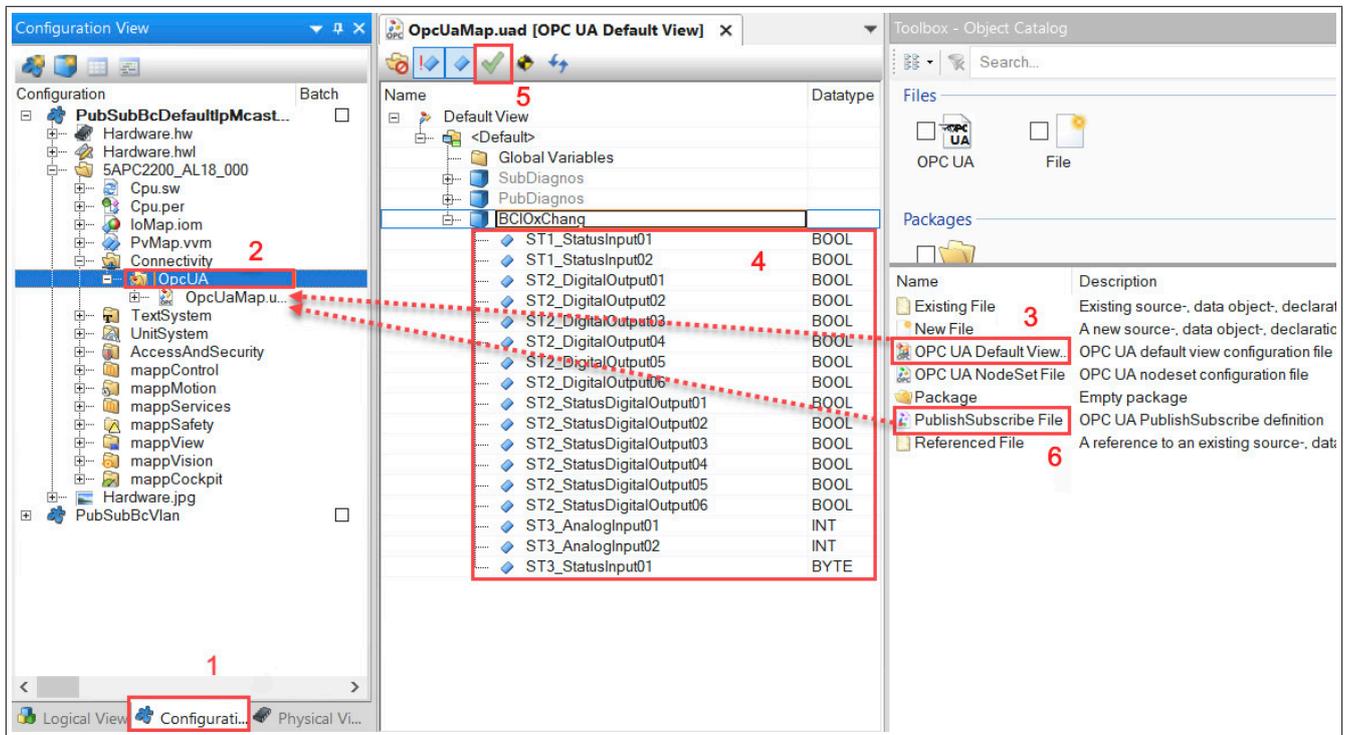


- Um Variablen über OPC UA verfügbar zu machen, in *Physical View* die Konfiguration durch einen Rechtsklick auf *CPU - Configuration* aufrufen. In der Konfiguration bei *OPC-UA System* "on" auswählen.



• Zuletzt müssen die Variablen für PubSub vorbereitet werden:

- 1) Zur *Configuration View* wechseln.
- 2) Ordner *Connectivity - OpcUA* auswählen.
- 3) *OPC UA Default View* durch Doppelklick zu Ordner *OpcUA* hinzufügen.
- 4) Alle Variablen der Bus Controller Datenpunkte markieren.
- 5) Sichtbarkeit des OPC UA Servers durch Klick auf grünen Haken aktivieren und *OPC UA Default View* speichern.
- 6) *PublishSubscribe File* durch Doppelklick zu Ordner *OpcUA* hinzufügen.



Somit ist das X2X-Prozessvariablenabbild des Bus Controllers im Automation Studio vorbereitet. Welche Variablen letztendlich übertragen werden, wird in der PubSub-Konfiguration festgelegt.

### 6.3.2.3.2 Bus Controller sendet zur Steuerung

#### UaExpert Konfiguration für Bus Controller

Bus Controller Publisher konfigurieren und Konfigurationsdatei (\*.uabinary) speichern (siehe 6.3.2.1 "Bus Controller als Publisher mit UaExpert konfigurieren")

#### Automation Studio Konfiguration für Steuerung

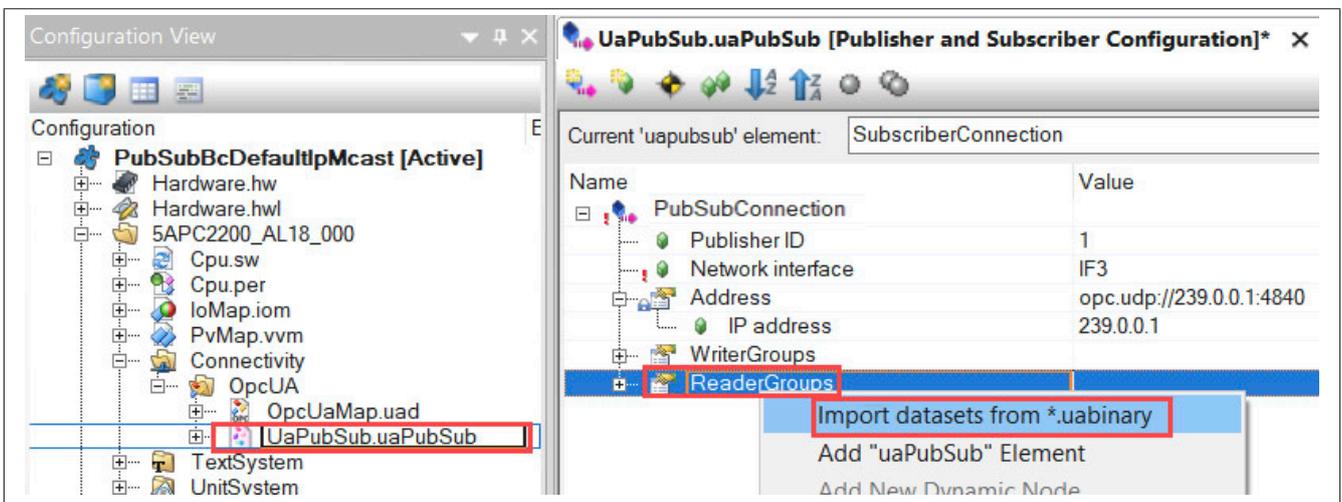
Variablen per OPC UA zugänglich machen und PubSub Konfiguration anlegen (siehe 6.3.2.3.1 "Prozessvariablenabbild im Automation Studio").

#### Subscriber konfigurieren

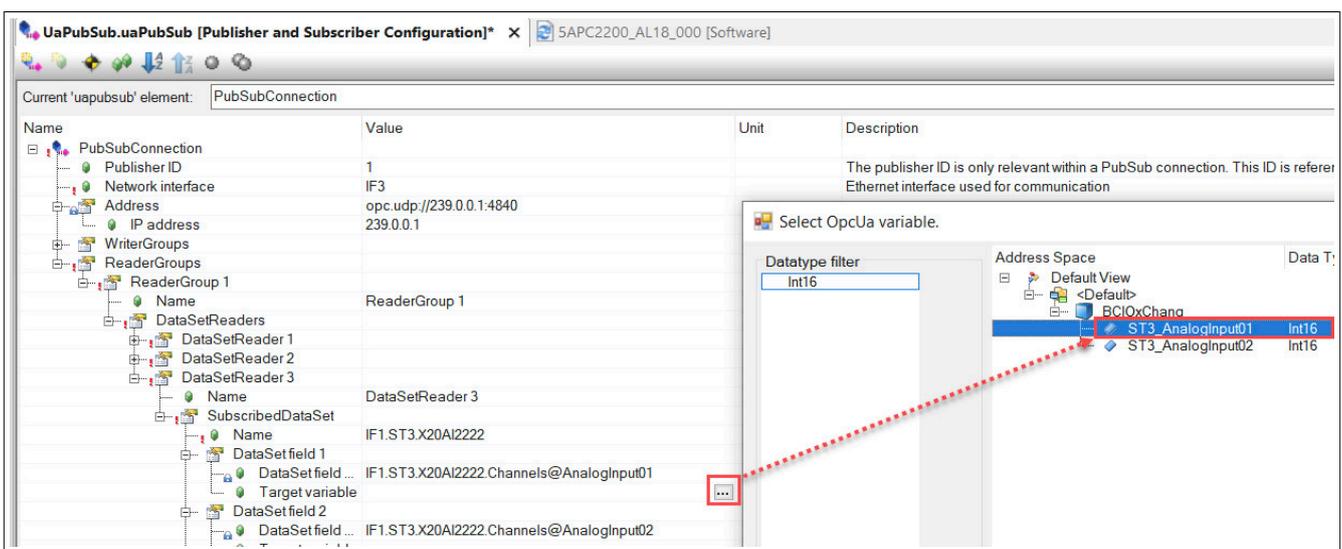
- Nach Rechtsklick auf *ReadersGroups* - *Import datasets from \*.uabinary* die gespeicherte PubSub-Konfigurationsdatei des Bus Controllers auswählen und in Automation Studio importieren.

#### Information:

Dazu eine unbenutzte Verbindung verwenden oder anlegen, um im Bedarfsfall die IP-Adresse unabhängig anpassen zu können.



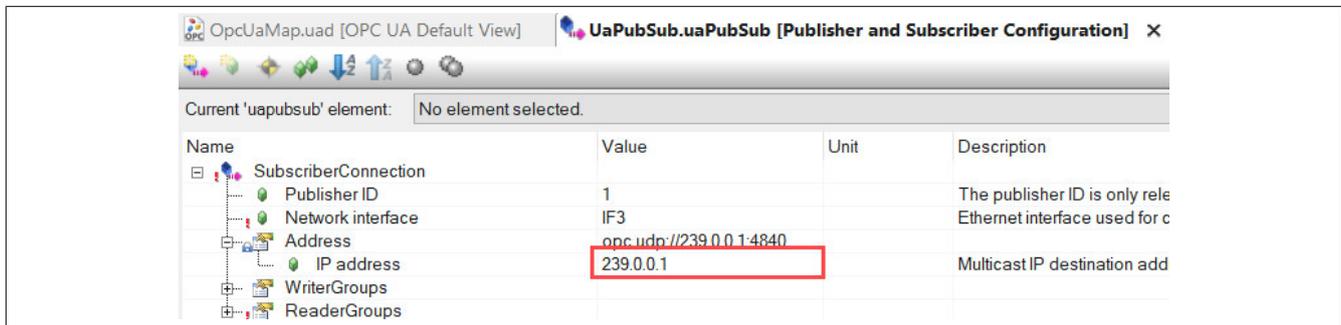
- Nach Doppelklick auf *DataSet field* im Dialog die gewünschte Variable auswählen. Nach Bestätigung der Auswahl wird die Variable in die *PubSubConnection* eingefügt. Schritt für weitere Variablen wiederholen.



- Subscriber IP-Adresse auf Publisher anpassen. Für die Publisher IP-Adresse siehe "Parameter anpassen" auf Seite 41.

## Information:

Diese Einstellung wird nicht automatisch durch den Import der PubSub-Konfigurationsdatei übernommen und muss manuell angepasst werden!

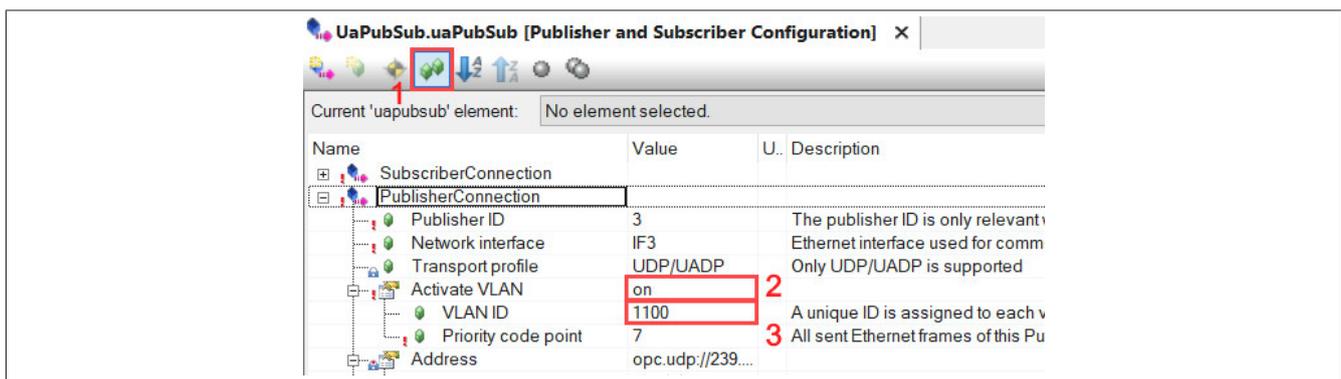


Name	Value	Unit	Description
SubscriberConnection			
Publisher ID	1		The publisher ID is only rele
Network interface	IF3		Ethernet interface used for c
Address	opc.tcp://239.0.0.1:4840		
IP address	239.0.0.1		Multicast IP destination add
WriterGroups			
ReaderGroups			

- VLAN anpassen (optional)

Die VLAN-Parameter sind nicht Teil der importierten "\*.uabinary"-Datei Konfiguration und können daher nachträglich verändert werden.

- 1) Schaltfläche *Advanced Parameter Visibility* aktivieren
- 2) In Parameter *Activate VLAN* "on" auswählen
- 3) VLAN-Parameter anpassen



Name	Value	U.	Description
SubscriberConnection			
PublisherConnection			
Publisher ID	3		The publisher ID is only relevant
Network interface	IF3		Ethernet interface used for comm
Transport profile	UDP/UADP		Only UDP/UADP is supported
Activate VLAN	on		
VLAN ID	1100		A unique ID is assigned to each v
Priority code point	7		All sent Ethernet frames of this Pu
Address	opc.tcp://239...		

## Information:

Der Parameter *Priority Code Point* wird nur auf enthaltene Publisher angewandt.

- Konfiguration speichern und auf die Steuerung übertragen.

## Überprüfung der Kommunikation

Eine Veränderung der Daten am Bus Controller sollten nun auch am entsprechenden Variablenwert der Steuerung sichtbar sein. Die Variablenwerte können im Automation Studio sichtbar gemacht werden unter "Logical View → Task (R-Klick) → Open → Watch"

### 6.3.2.3.3 Steuerung sendet zum Bus Controller

#### Automation Studio Konfiguration für Steuerung

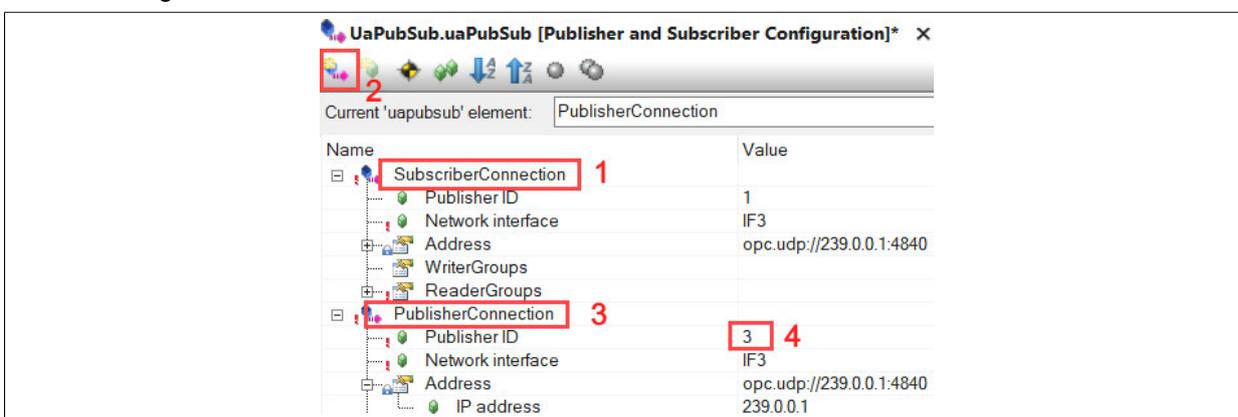
Variablen per OPC UA zugänglich machen und PubSub Konfiguration anlegen (siehe 6.3.2.3.1 "Prozessvariablenabbild im Automation Studio")

#### Publisher konfigurieren

##### Information:

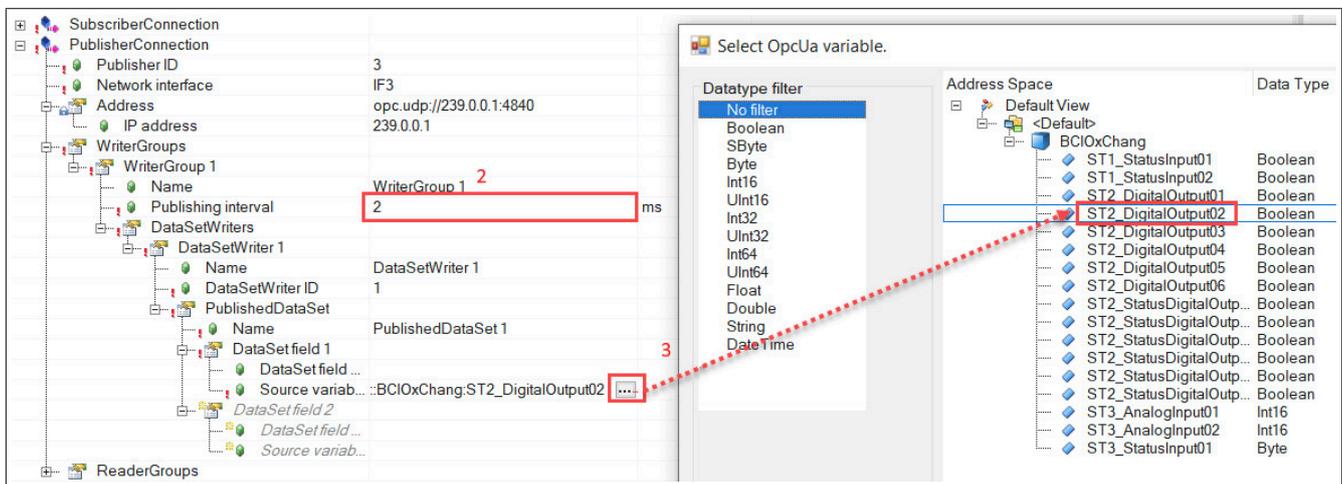
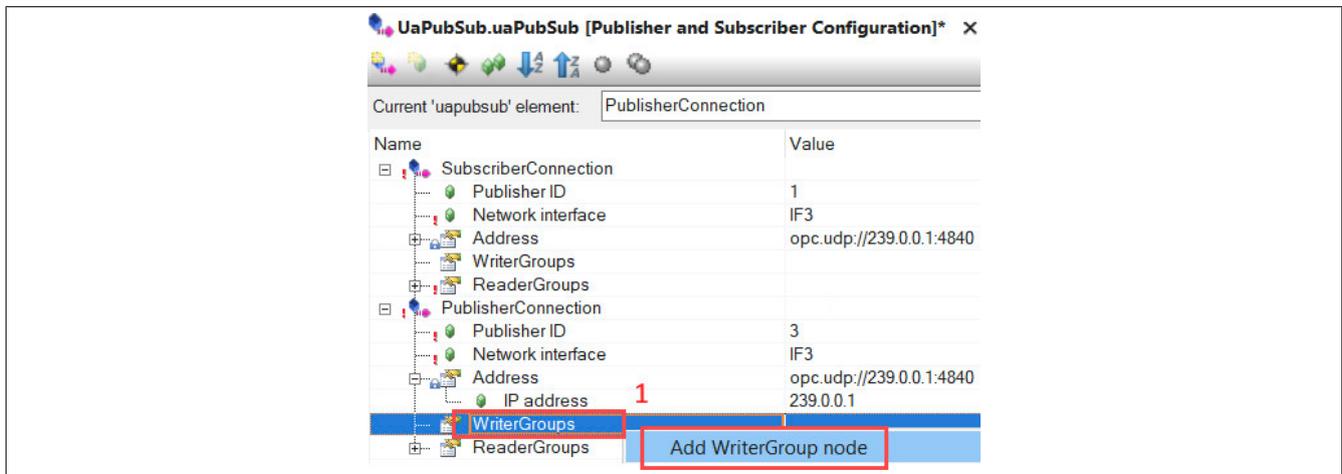
Es sollten getrennte "Connection" für die Publisher und Subscriber Konfiguration angelegt werden, um eine Kombination von *ReaderGroups* und *WriterGroups* innerhalb einer "Connection" zu vermeiden. Eine gemeinsame Verwendung würde die Flexibilität der Verbindungsparameter stark einschränken.

- Um die PublisherConnection hinzuzufügen, folgende Schritte durchführen:
  - 1) Eventuell vorhandenen Verbindung umbenennen, z. B. zu "SubscriberConnection".
  - 2) Durch Klick auf die Schaltfläche eine neue Verbindung hinzufügen
  - 3) Die neue Verbindung umbenennen, z. B. zu "PublisherConnection"
  - 4) Eine eindeutige PublisherId zuweisen. Diese darf im Netzwerk nur einmal vorhanden sein.

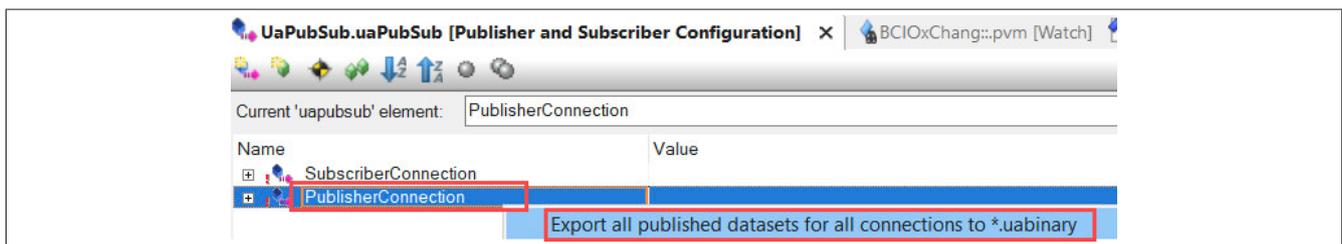


- Um die WriterGroup zu konfigurieren, folgende Schritte durchführen:

- 1) Durch Rechtsklick auf *WriterGroup* → *Add WriterGroup node* eine neue *WriterGroup* hinzufügen.
- 2) Das *Publishing interval* auf Applikationsanforderung anpassen, z. B. 2 ms. Der Wert sollte der Zykluszeit des Automation Runtime Tasks entsprechen, in dem die Variablen definiert sind.
- 3) Task-Variable *PublishedDataSet*-Quelle auswählen.



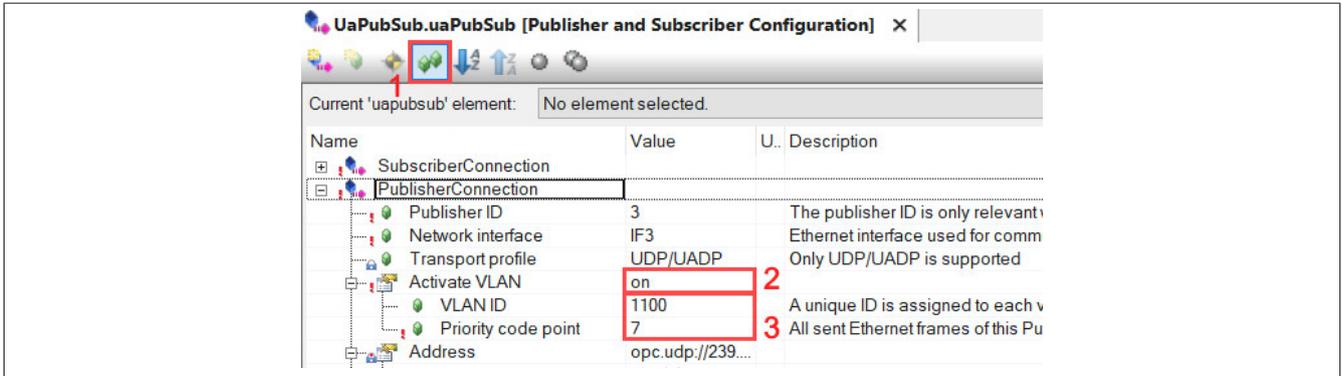
- Konfiguration speichern und Konfigurationsdatei (\*.uabinary) exportieren



- VLAN verwenden (optional)

Die VLAN-Parameter sind nicht Teil der exportierten "\*.uabinary"-Datei Konfiguration und können daher nachträglich verändert werden. Eine entsprechende Anpassung der Subscriber ist ebenfalls erforderlich.

- 1) Schaltfläche *Advanced Parameter Visibility* aktivieren
- 2) In Parameter *Activate VLAN* "on" auswählen
- 3) VLAN-Parameter anpassen



- Konfiguration speichern und auf die Steuerung übertragen.

### **UaExpert Konfiguration für Bus Controller**

Steuerungs-Konfigurationsdatei in UaExpert öffnen und X2X Zielknoten zuweisen (siehe 6.3.2.2 "Bus Controller als Subscriber mit UaExpert konfigurieren")

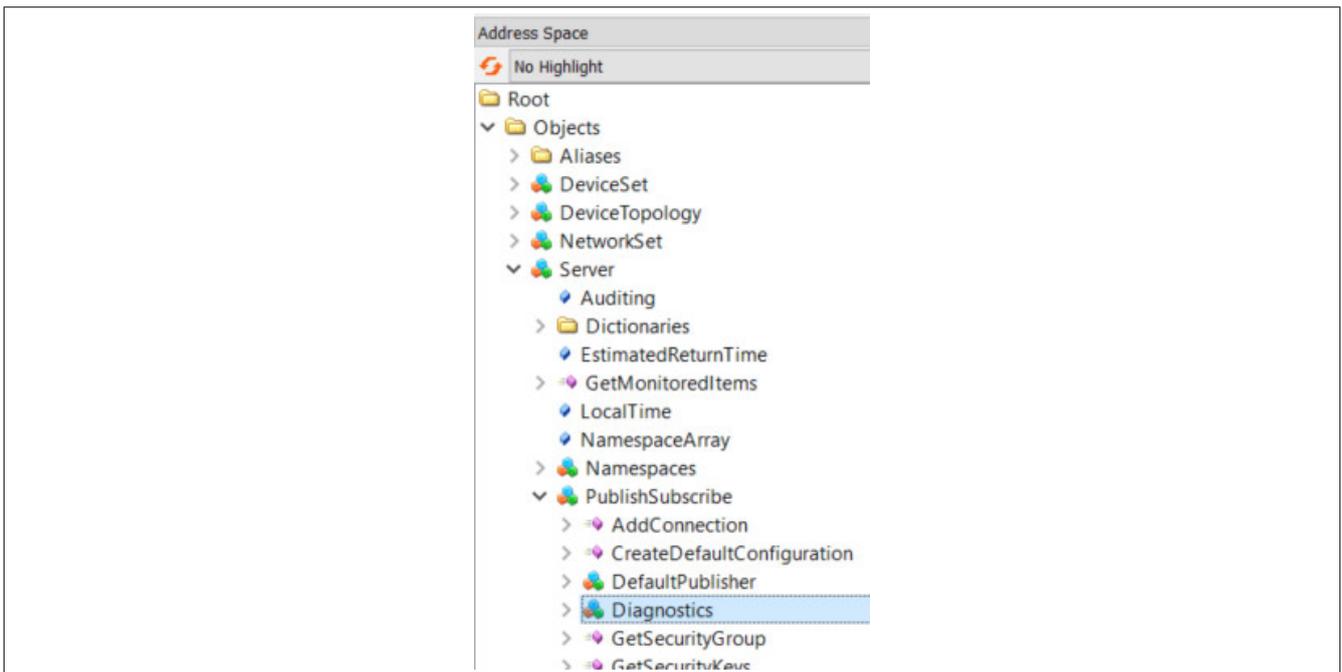
### **Überprüfung der Kommunikation**

Eine Veränderung der Steuerungs-Variablen sollte nun am entsprechenden Bus Controller Modul sichtbar sein. Die Änderung kann im Automation Studio oder UaExpert durchgeführt werden.

## **6.4 PubSub Diagnose**

Eine gültige Konfiguration stellt Diagnosedatenpunkte im Informationsmodell des Geräts zur Verfügung. Damit lässt sich eine fehlerhafte PubSub-Konfiguration diagnostizieren und der Konfigurationsstatus auswerten.

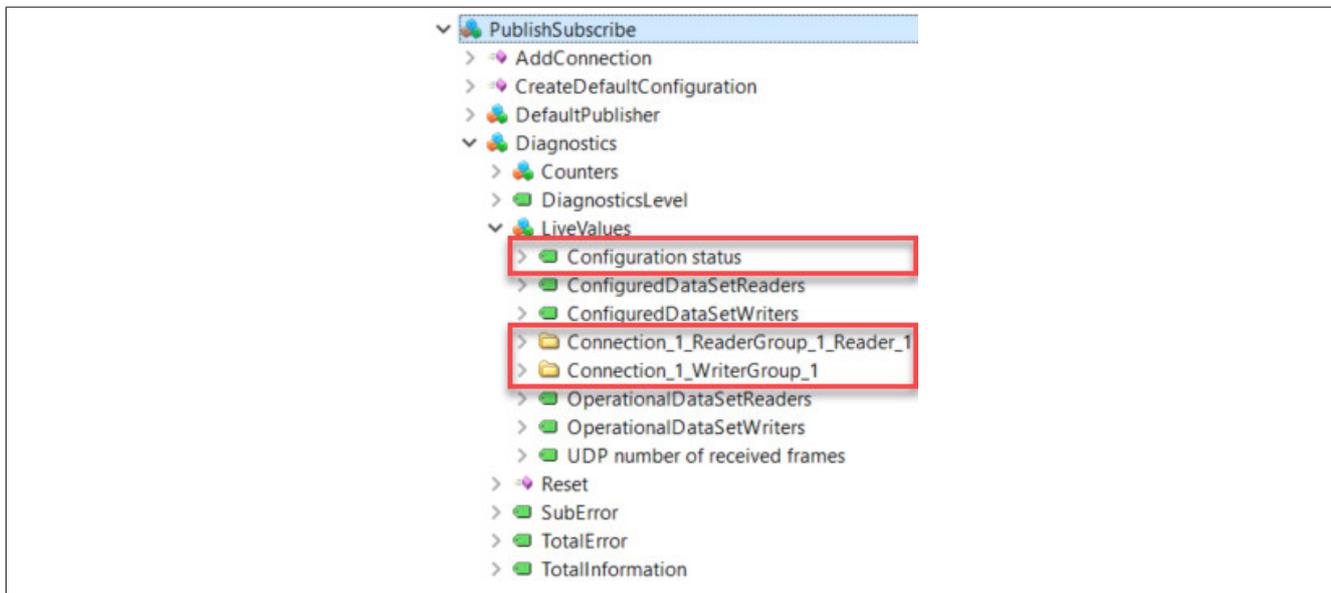
Die erzeugten Diagnosedatenpunkte befinden sich im Pfad *Root/Objects/Server/PublishSubscribe/Diagnostics*.



Im *Diagnostics*-Objekt können nur die rot markierten Unterknoten des *LiveValues*-Objekts zu Diagnosezwecken herangezogen werden.

- Der Knoten "Configuration status" enthält Informationen zur aktuellen Konfiguration.
- Für jede Publisher/Subscriber-Konfiguration wird ein eigener Ordner erzeugt, sofern die PubSub Konfigurationsdatei (PubSub.uabinary) erfolgreich gelesen wurde. Der Name des Ordners wird vom Connection- und Group-Namen der PubSub-Konfiguration abgeleitet z. B. "Connection\_1\_WriterGroup\_1".

Alle anderen Knoten bzw. Objekte werden nicht unterstützt und enthalten keine Werte. Das *LiveValues*-Objekt befindet sich im Pfad *Root/Objects/Server/PublishSubscribe/Diagnostics/LiveValues*.



#### 6.4.1 Konfigurationsstatus

Der Knoten "Configuration status" enthält Informationen zur aktuellen Konfiguration.

Knotenname	Beschreibung
Configuration status	Aktueller Konfigurationsstatus 0: OK -1: Problem bei der Parametrierung der Kernel-Treiber -2: Problem bei der Interpretation der "PubSub.uabinary" Datei (z. B. die gemappten Nodes existieren nicht) -99: Konfiguration wurde noch nicht geladen

#### 6.4.2 Writer Group Diagnoseknoten

Der Diagnoseknoten <Writer Group Name> wird durch eine gültige WriterGroup-Konfigurationen erzeugt und befindet sich im Pfad *Root/Objects/Server/PublishSubscribe/Diagnostics/LiveValues/<Writer Group Name>*.

Publisher haben im Vergleich zu Subscriber eine eingeschränkte Diagnose, da nur das Versenden von Telegrammen überwacht wird.

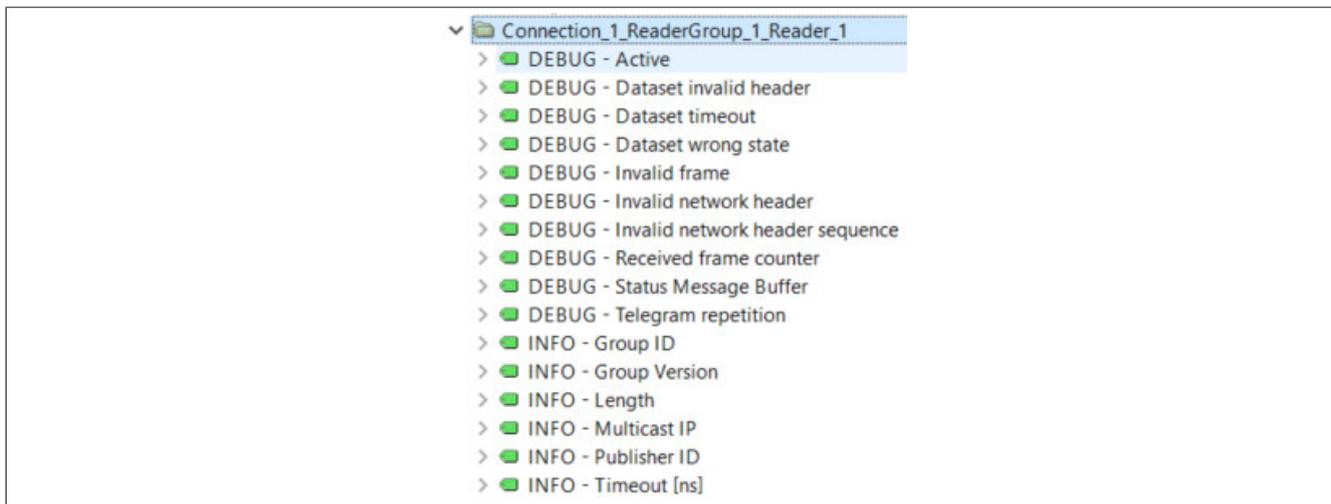


Knotenname	Beschreibung
DEBUG - Active	Paketversand ist aktiv / nicht aktiv
DEBUG - Status Message Buffer	0: kein Fehler -1: Socket konnte nicht geöffnet werden
INFO - Group ID	GroupID der WriterGroup
INFO - Group Version	GroupVersion der WriterGroup
INFO - Interval [ns]	Sendeintervall der WriterGroup in ns
INFO - Length	UADP-Telegrammlänge – ausgenommen Ethernet- und IP/UDP-Header
INFO - Multicast IP	Multicast-IP-Zieladresse des Publishers z. B. 239.0.0.1
INFO - Publisher ID	PublisherID des Publishers

### 6.4.3 Reader Group Diagnoseknoten

Der Diagnoseknoten <Reader Group Name> wird durch eine gültige ReaderGroup-Konfigurationen erzeugt und befindet sich im Pfad *Root/Objects/Server/PublishSubscribe/Diagnostics/LiveValues/<Reader Group Name>*.

Subscriber haben im Vergleich zu Publishern eine erweiterte Diagnose, da als ungültig erkannte Telegramme in verschiedenen Fehlerzählern erfasst werden.



Knotenname	Beschreibung
DEBUG - Active	Paketversand ist aktiv / nicht aktiv
DEBUG – Dataset invalid header	Zähler wird erhöht, wenn die Dataset-Headerflags nicht der Subscriber-Konfiguration entspricht.
DEBUG – Dataset timeout	Zähler wird erhöht, wenn ein Dataset ausfällt; das heißt, es wurden keine neuen Daten vor Ablauf des eingestellten Timeouts empfangen.
DEBUG – Dataset wrong state	Zähler wird erhöht, wenn das Status Feld des DataSetMessage-Headers eine "StatusCode severity" ungleich "Good" enthält.
DEBUG – Invalid frame	Zähler wird erhöht, wenn die Network Header- oder Groupflags nicht der Subscriber-Konfiguration, das heißt, dem erwarteten Telegrammheader, entsprechen. Wurden seit dem Zeitpunkt der PubSub-Konfiguration noch keine Daten empfangen, erhöht sich der Zähler ebenfalls (Empfangspuffer enthält 0 als Telegrammheader).
DEBUG – Invalid network header	Zähler wird erhöht, wenn einer der Felder Extended Flags, PublisherId, WriterGroupId oder GroupVersion nicht der Subscriber-Konfiguration entsprechen.
DEBUG – Invalid network header sequence	Zähler wird erhöht, wenn die SequenceNumber im Networkheader gegenüber dem Vorgänger nicht aufsteigend war.
DEBUG – Status Message Buffer	0: Kein Fehler -1: Socket konnte nicht geöffnet werden
DEBUG – Telegram repetition	Zähler wird erhöht, wenn zweimal in Folge dasselbe Telegramm empfangen wurde, das heißt, die SequenceNumber des NetworkHeaders ist identisch. <b>Information:</b> Bei nicht synchronisierten Systemen ist es wahrscheinlich, dass dieser Fehlerzähler langsam hoch zählt.
INFO – Group ID	Erwartete GroupID
INFO – Group Version	Erwartete GroupVersion
INFO - Length	Erwartete UADP-Telegrammlänge – ausgenommen Ethernet- und IP/UDP-Header
INFO – Multicast IP	Multicast,IP-Adresse, auf welcher der Subscriber Telegramme erwartet z. B. 239.0.0.1
INFO – Publisher ID	Erwartete PublisherId
INFO – Timeout [ns]	Konfiguriertes Message Receive Timeout des DataSet Reader in ns

## 7 Status

Das OPC UA Informationsmodell des Bus Controllers zeigt Statusinformationen, die der Information beziehungsweise Diagnose von auftretenden Funktionsstörungen dienen sollen. Diese sind:

- [Port-Status](#)
- [Zeitsynchronisation](#)
- [Netzwerk](#)

### 7.1 Port-Status

Die entsprechenden Knotennamen befinden sich im OPC UA Informationsmodell unter den Knoten *Root/Objects/DeviceSet/X20BC008T/Status/BridgePorts*. Für jeden Port des Bus Controllers befindet sich unter diesem Pfad ein Eintrag mit der Bezeichnung des Ports selbst (*ETHx*). Die folgende Tabelle listet die für jeden Port verfügbaren Statusinformationen.

Knotenname	Beschreibung
InternalName	Systeminterner Name der Schnittstelle.
<b>FrameStatistics/</b>	
FcsErrorFrameCount	Anzahl der am jeweiligen Port empfangenen Ethernet-Frames mit fehlerhafter Ethernet FCS (Frame Check Sequence)
GeneralRxErrorFrameCount	Anzahl der am jeweiligen Port eingegangenen Ethernet-Frames, die aufgrund Bus Controller interner Fehler nicht empfangen werden konnten.
GeneralTxErrorFrameCount	Anzahl der am jeweiligen Port zu sendenden Ethernet-Frames, die aufgrund Bus Controller interner Fehler nicht gesendet werden konnten.
RxFrameCount	Anzahl der am jeweiligen Port erfolgreich empfangenen Ethernet-Frames.
SizeErrorFrameCount	Anzahl der am jeweiligen Port empfangenen Ethernet-Frames, die aufgrund ungültiger Länge (< 64 Byte oder > max. Ethernet Framelänge) verworfen wurden.
TxFrameCount	Anzahl der am jeweiligen Port erfolgreich gesendeten Ethernet-Frames.
<b>LinkPartner/</b>	
ChassisId	Bezeichnung der "Chassis"-Komponente des angeschlossenen Geräts, z. B. die MAC-Adresse.
ManagementAddress	Management-Adresse des angeschlossenen Geräts, z. B. die IP-Adresse.
PortId	Bezeichnung der "Port"-Komponente des angeschlossenen Geräts, z. B. interner Schnittstellename.
<b>LinkProperties/</b>	
Duplex	Duplexmodus des Ports. Mögliche Werte: Full                      Port arbeitet im Full-Duplex Modus Half                      Port arbeitet im Half-Duplex Modus Kein Eintrag              Verbindung ist nicht aktiv
LinkStatus	Status der Verbindung. Mögliche Werte: UP                      Verbindung ist aktiv DOWN                    Verbindung ist nicht aktiv
Speed	Geschwindigkeit der Verbindung. Mögliche Werte: 100Mb/s                Verbindung arbeitet mit 100 Mbit/s 1000Mb/s              Verbindung arbeitet mit 1000 Mbit/s Kein Eintrag            Verbindung ist nicht aktiv

## 7.2 Zeitsynchronisation

Der Zustand der Zeitsynchronisierung kann über die Knoten im Objekt *Root/Objects/DeviceSet/X20BC008T/Status/TimeSynchronization* abgefragt werden. Entsprechende Informationen stehen sowohl für die *WallClock*, als auch für die *WorkingClock* zur Verfügung.

Knotenname	Beschreibung
WallClock/NTP/	
SyncOK	Status der NTP-Synchronisation der WallClock. Mögliche Werte: True Die WallClock ist mit einem Zeitserver synchronisiert False Die WallClock ist mit keinem Zeitserver synchronisiert
TimeServer	URL oder IP-Adresse des Zeitservers, mit dem die WallClock synchronisiert wird.
WallClock/PTP/	
ClockIdentity	Eindeutiger Identifikator der PTP-Instanz der WallClock.
GrandmasterIdentity	Identität der PTP-Instanz, die im Netzwerk als Grandmaster für die WallClock dient.
OffsetFromMaster	Berechnete Zeitabweichung in 1/65536 Nanosekunden <sup>1)</sup> der lokalen WallClock zur Uhr des Grandmasters.
ParentPortIdentity	Identität des Ports jenes Nachbargeräts, über den die PTP-Synchronisationsnachrichten zur lokalen WallClock PTP-Instanz gesendet werden. Die Identität ist als Byte-String dargestellt. Wenn die lokale WallClock die Grandmaster-Instanz ist, entspricht dieser String der <i>ClockIdentity</i> gefolgt von 2 Null-Bytes.
WallClock/PTP/ETHx/	
PortIdentity	Eindeutiger Port Identifikator des Ethernet Ports <b>ETHx</b> bei aktivierter PTP-Synchronisation der WallClock an diesem Port. x korrespondiert mit der Nummer des Ports am Gehäuse des Bus Controllers.
PortState	Status der WallClock PTP-Synchronisation am Ethernet Port <b>ETHx</b> . Mögliche Werte: 3 Port ist deaktiviert 6 Port ist Master für die WallClock 7 Port ist passiv 9 Port ist Slave für die WallClock  Detailinformationen siehe IEEE 802.1AS – 2020, Tabelle 14-7.
WorkingClock/PTP/	
ClockIdentity	Eindeutiger Identifikator der PTP-Instanz der WorkingClock.
GrandmasterIdentity	Identität der PTP-Instanz, die im Netzwerk als Grandmaster für die WorkingClock dient.
OffsetFromMaster	Berechnete Zeitabweichung in 1/65536 Nanosekunden <sup>1)</sup> der lokalen WorkingClock zur Uhr des Grandmaster.
ParentPortIdentity	Identität des Ports jenes Nachbargeräts, über den die PTP-Synchronisationsnachrichten zur lokalen WorkingClock PTP-Instanz gesendet werden. Die Identität ist als Byte-String dargestellt. Wenn die lokale WorkingClock die Grandmaster-Instanz ist, entspricht dieser String der <i>ClockIdentity</i> gefolgt von 2 Null-Bytes.
WorkingClock/PTP/ETHx/	
PortIdentity	Eindeutiger Port Identifikator des Ethernet Ports <b>ETHx</b> bei aktivierter PTP-Synchronisation der WorkingClock an diesem Port. x korrespondiert mit der Nummer des Ports am Gehäuse des Bus Controllers.
PortState	Status der WorkingClock PTP-Synchronisation am Ethernet Port <b>ETHx</b> . Mögliche Werte: 3 Port ist deaktiviert 6 Port ist Master für die WorkingClock 7 Port ist passiv 9 Port ist Slave für die WorkingClock  Detailinformationen siehe IEEE 802.1AS – 2020, Tabelle 14-7.

1) Wert 65536 = 1 Nanosekunde.

## 7.3 Netzwerk

Die aktuell verwendete Netzwerkkonfiguration kann über die Knoten im Objekt *Root/Objects/DeviceSet/X20BC008T/Status/Network* ausgelesen werden.

Knotenname	Beschreibung
CurrentDNS	Aktuell verwendete DNS-Server. Der String kann mehrere Einträge enthalten, wenn mehrere DNS-Server zur Verfügung stehen.
CurrentGateway	Aktuell verwendeter Default-Gateway
CurrentHostname	Aktuell verwendeter Hostname
CurrentIPConfig	Aktuelle IP-Konfiguration. Der String kann mehrere Einträge enthalten, falls mehrere IP-Adressen existieren (z. B. eine durch Verstellen des Knotennummerschalters hinzugefügte temporäre IP-Adresse).

## 8 Cyber-Security

Dieses Kapitel gibt eine kurze Einführung in das Thema der Cyber-Security. Die Beschreibung der Begriffe erfolgt dabei nur auf sehr allgemeiner Ebene. Daher können unter einem allgemeinen Begriff je nach Situation eine Reihe unterschiedlicher Aspekte gemeint sein.

Geräte werden mit Werkseinstellungen ausgeliefert. Das bedeutet, dass normalerweise weder Gerätefunktionalität noch Sicherheitseinstellungen konfiguriert sind. Um die Inbetriebnahme dieser Geräte sicher zu gestalten, sollte daher dafür gesorgt werden, dass sie vorerst nur in einer vertrauenswürdigen Umgebung benutzt werden. Das kann z. B. erreicht werden, indem das Maschinennetzwerk vom restlichen Unternehmensnetzwerk getrennt ist, oder die Geräte direkt mit dem zur Konfiguration benutzten PC verbunden werden.

OPC UA over TSN ermöglicht IT-OT konvergente Netzwerke, in denen man nicht davon ausgehen kann, dass alle Netzwerkteilnehmer vertrauenswürdig sind. Das setzt keinen bewussten Angriff voraus, sondern bereits Fehlkonfiguration von Steuerungen außerhalb des eigentlichen Maschinennetzwerks könnten zu unbeabsichtigten Störungen führen.

Fragen der Cyber-Security spielen daher in OPC UA over TSN eine wichtige Rolle und beide Basistechnologien, das heißt, sowohl OPC UA als auch TSN, enthalten alle dafür notwendigen Mechanismen.

### Security-Relevante Fehler und Benachrichtigungen

Cyber-Security lebt von einer offenen Fehlerkultur. Fehler einer Geräte-Firmware, die z. B. unberechtigten Zugriff erlauben, werden von B&R aktiv verfolgt und behandelt. Kritische Sicherheitslücken und deren Behebung werden gesammelt unter <https://www.br-automation.com/en/service/cyber-security/> zur Verfügung gestellt.

#### Information:

**Alle Fehler, die die Sicherheit von B&R Geräten betreffen, sollen unverzüglich an die oben angegebene Webseite gemeldet werden.**

### 8.1 Grundbegriffe und Grundlagen

#### 8.1.1 Verschlüsselung

Ziel der Verschlüsselung ist es, schützenswerte Daten für Außenstehende unlesbar zu machen. Selbst wenn ein Angreifer Zugriff auf die Daten hat, z. B. indem er mit Hilfe von Werkzeugen den Datenverkehr mitverfolgt, sollte es für ihn unmöglich sein, daraus wertvolle Informationen abzuleiten.

Zudem wird bei der Verschlüsselung unterschieden, ob Daten auf einem Computer, Gerät oder Datenträger gespeichert bleiben, oder ob sie über ein Kommunikationsmedium übertragen werden. Die grundlegenden Mechanismen sind in allen Fällen ähnlich.

Der Industriestandard für die Verschlüsselung ist die AES-Familie (Advanced Encryption Standard) und arbeitet mit Schlüssellängen von 128 oder 256 Bit. Sowohl OPC UA, als auch NETCONF unterstützen diesen Standard.

#### 8.1.2 Integrität

Ein Angreifer muss geheime Daten nicht unbedingt entschlüsseln, um z. B. Störungen im Ablauf einer Maschine hervorzurufen. Vielmehr ist es oft schon ausreichend Daten zu verfälschen. Das gelingt selbst dann, wenn Daten verschlüsselt und eigentlich unlesbar sind.

Um diese Bedrohung zu verhindern, werden Daten daher um eine digitale Signatur erweitert. Die ist ähnlich einer CRC-Prüfsumme (Cyclic Redundancy Check). Die Algorithmen sind aber explizit darauf ausgelegt Verfälschungen durch einen Angreifer zu erkennen.

Häufig reicht es aus, lediglich die Integrität der Daten sicherzustellen zu können, ohne sie zu verschlüsseln zu müssen. Anwendungsfälle dafür sind zum Beispiel:

- Die Diagnose des Datenverkehrs mit Hilfe von Werkzeugen wie Wireshark. Die digitale Signatur verhindert die Diagnose nicht, wohingegen eine zusätzliche Verschlüsselung die Daten unlesbar und für eine Diagnose unbrauchbar machen würde.
- Sicherstellung der Integrität der Firmware von Geräten. Die Firmware bleibt auslesbar, es ist aber für einen Angreifer trotzdem nicht möglich Veränderungen durchzuführen.

Der Industriestandard für Signaturalgorithmen ist die SHA-Familie (Secure Hash Algorithm), mit Schlüssellängen von z. B. 256 Bit. Sowohl OPC UA, als auch NETCONF unterstützen diese Algorithmen.

### 8.1.3 Symmetrische und asymmetrische Schlüssel

Algorithmen wie die AES-Familie werden als "symmetrisch" bezeichnet, weil ein einziger Schlüssel sowohl für die Verschlüsselung als auch die Entschlüsselung verwendet wird. Falls 2 Geräte miteinander Daten verschlüsselt austauschen wollen, muss also zuvor sichergestellt sein, dass beide Geräte denselben Schlüssel besitzen. Das ist in der Praxis nicht immer einfach durchzuführen.

Algorithmen wie die RSA-Familie (benannt nach den Erfindern Rivest, Shamir und Adleman) werden dagegen als "asymmetrisch" bezeichnet. Diese Algorithmen verwenden 2 unterschiedliche Schlüsseln, um das Problem des Schlüsselaustauschs zu vereinfachen.

- Ein "privater" Schlüssel dient dazu, Daten zu entschlüsseln, bzw. die Signatur zu erstellen.
- Ein "öffentliche" Schlüssel dient dazu, Daten zu verschlüsseln, bzw. die Authentizität einer Signatur zu prüfen.

Der öffentliche Schlüssel darf – und soll – von jedem lesbar sein. Geräte, die miteinander kommunizieren wollen, stellen einander gegenseitig ihre öffentlichen Schlüsseln zur Verfügung. Da der private Schlüssel nicht übermittelt werden muss, kann er auf einfache Weise im Gerät geheim gehalten werden.

Ablauf der Datenübertragung mit asymmetrischen Algorithmen:

- Der Sender A signiert die Daten mit seinem privaten Schlüssel  $P_A$ .
- Der Sender A verschlüsselt die signierten Daten mit dem öffentlichen Schlüssel  $\ddot{O}_B$  des Empfängers B.
- Der Empfänger B entschlüsselt die verschlüsselten und signierten Daten mit seinem privaten Schlüssel  $P_B$ .
- Der Empfänger B prüfte die Echtheit der signierten Daten mit dem öffentlichen Schlüssel  $\ddot{O}_A$  des Senders A.

Der Nachteil der asymmetrischen Algorithmen besteht im wesentlich größeren Rechenaufwand. Da für RSA Schlüssel mit einer Länge von 2048 Bit verwendet werden, sind sie für den Austausch großer Datenmengen nicht geeignet. Symmetrische Algorithmen wiederum verwenden nur Schlüssellängen von 256 Bit, wodurch sich die Datenübertragung wesentlich einfacher durchführen lässt.

In der Praxis wird daher oft eine Kombination beider Verfahren eingesetzt. Asymmetrische Algorithmen werden verwendet, um einen, bloß temporär für die Kommunikationssitzung erzeugten, symmetrischen Schlüssel auszutauschen. Die eigentliche Kommunikation danach wird mit symmetrischen Algorithmen durchgeführt.

### 8.1.4 Asymmetrischer Schlüsselaustausch

Obwohl ein öffentlicher Schlüssel von jedem gelesen und benutzt werden darf, bedeutet das nicht, dass keine Vorsicht bei dessen Verwendung nötig wäre. Ein Sender A muss z. B. sicher sein, dass der öffentliche Schlüssel  $\ddot{O}_B$  auch tatsächlich dem gewünschten Empfänger B gehört. Ohne eine derartige Versicherung wäre nämlich der folgende, als "Man-in-the-Middle" bekannte, Angriff möglich, bei dem sich der Angreifer in die Kommunikation einklinkt:

Sender A  $\leftrightarrow$  Angreifer C  $\leftrightarrow$  Empfänger B

- Der Sender A signiert die Daten mit seinem privaten Schlüssel  $P_A$ .
- Der Sender A verschlüsselt die signierten Daten fälschlicherweise mit dem öffentlichen Schlüssel  $\ddot{O}_C$  des Angreifers C, an Stelle des öffentlichen Schlüssels  $\ddot{O}_B$  des Empfängers B.
- Der Angreifer C entschlüsselt die verschlüsselten und signierten Daten mit seinem privaten Schlüssel  $P_C$ .
- Der Angreifer C liest die Daten und verfälscht sie eventuell.
- Der Angreifer C signiert die Daten mit seinem privaten Schlüssel  $P_C$ .
- Der Angreifer C verschlüsselt die signierten Daten mit dem öffentlichen Schlüssel  $\ddot{O}_B$  des Empfängers B.
- Der Empfänger B prüfte die Echtheit der signierten Daten fälschlicherweise mit dem öffentlichen Schlüssel  $\ddot{O}_C$  des Angreifers C, an Stelle des öffentlichen Schlüssels  $\ddot{O}_A$  des Senders A.

Der Angreifer kann ebenso die umgekehrte Kommunikationsrichtung mitlesen und verfälschen.

Um sich vor einem Man-in-the-Middle-Angriff zu schützen existieren im Wesentlichen 3 Möglichkeiten:

- 1) Sicherstellen, dass zu Beginn des Kommunikationsaufbaus kein Angreifer anwesend sein kann, z. B. indem die Maschine vom Intra- und Internet getrennt ist. Die ausgetauschten Schlüssel sind danach sicher, auch wenn die Maschine wieder mit dem Intra- und Internet verbunden wird.
- 2) Sicherstellen, dass der empfangene öffentliche Schlüssel  $\tilde{O}_X$  tatsächlich zum Kommunikationsteilnehmer X gehört. Das ist möglich, wenn es eine vertrauenswürdige dritte Stelle gibt, die garantiert, dass dieser Zusammenhang besteht.
- 3) Die öffentlichen Schlüssel auf eine geeignete Weise auf die jeweiligen Geräte verteilen. Dieser Weg bedeutet in der Regel manuelle Arbeit eines Benutzers oder Administrators.

NETCONF unterstützt – bei Verwendung des Kommunikationsprotokolls SSH (Secure Shell) – den ersten und dritten Weg. OPC UA unterstützt den zweiten und dritten Weg.

### 8.1.5 Vertrauenshierarchie und Autorität

Das auch im Internet angewandte Übertragungsprotokoll HTTPS (HyperText Transport Protocol Secure) basiert darauf, dass eine vertrauenswürdige dritte Stelle dafür bürgt, dass der öffentliche Schlüssel  $\tilde{O}_X$  zu dem Kommunikationsteilnehmer X gehört.

Diese Garantie ist zusammen mit weiteren Informationen in einem sogenannten "Zertifikat" enthalten. Das Format der Zertifikate wurde durch die ITU (International Telecommunication Union) standardisiert und folgt dem Standard X.509. Die "bürgende" Stelle wird dementsprechend als Zertifizierungsstelle (engl. "Certificate Authority" CA) bezeichnet.

Ein Web-Browser akzeptiert z. B. das Zertifikat für <https://www.br-automation.com>, weil es beweisbar von der Zertifizierungsstelle "GlobalSign" ausgestellt wurde und der Web-Browser dieser Zertifizierungsstelle vertraut. Das Zertifikat von <https://www.br-automation.com> muss dafür vor Verfälschung geschützt sein, was wiederum über die [symmetrischen und asymmetrischen Verfahren](#) sichergestellt wird.

Während sich eine CA selbst durch eine höhere CA zertifizieren lässt, gibt es einige CAs, denen Web-Browser und andere Geräte per Default vertrauen und die fest vorgegeben sind; die sogenannten Root-CAs. Diese stellen die höchste Certificate Authority im Internet dar.

- Eine Root-CA R erzeugt ein Zertifikat  $Z_R$  für sich selbst, das ihren öffentlichen Schlüssel  $\tilde{O}_R$  enthält.
- Die Root-CA R signiert das Zertifikat  $Z_R$  mit ihrem privaten Schlüssel  $P_R$  (self-signed Certificate).
- Ein Gerät A (oder Web-Browser) importiert das Zertifikat  $Z_R$  und kann damit überprüfen, ob weitere Zertifikate gegebenenfalls von der Root-CA R ausgestellt wurden.
- Die Root-CA R erstellt ein Zertifikat  $Z_A$  für das Gerät A, das dessen öffentlichen Schlüssel  $\tilde{O}_A$  enthält.
- Die Root-CA R signiert das Zertifikat  $Z_A$  mit ihrem privaten Schlüssel  $P_R$ .
- Ein Gerät B (oder Web-Browser) importiert das Zertifikat  $Z_R$  und kann damit überprüfen, ob weitere Zertifikate gegebenenfalls von der Root-CA R ausgestellt wurden.
- Die Root-CA R erstellt ein Zertifikat  $Z_B$  für das Gerät B, das dessen öffentlichen Schlüssel  $\tilde{O}_B$  enthält.
- Die Root-CA R signiert das Zertifikat  $Z_B$  mit ihrem privaten Schlüssel  $P_R$ .

Sobald Gerät A und Gerät B eine Kommunikationsverbindung eingehen, übermitteln sie einander zuerst ihre Zertifikate:

- Der Sender A sendet sein Zertifikat  $Z_A$  an den Empfänger B.
- Der Empfänger B überprüft die Integrität des Zertifikats  $Z_A$ , an Hand des öffentlichen Schlüssels  $\tilde{O}_R$ , das er dem Zertifikat  $Z_R$  der Root-CA R entnimmt.
- Der Empfänger B sendet sein Zertifikat  $Z_B$  an den Sender A.
- Der Sender A überprüft die Integrität des Zertifikats  $Z_B$ , an Hand des öffentlichen Schlüssels  $\tilde{O}_R$ , das er dem Zertifikat  $Z_R$  der Root-CA R entnimmt.
- Der Sender A signiert die Daten mit seinem privaten Schlüssel  $P_A$ .
- Der Sender A verschlüsselt die signierten Daten mit dem öffentlichen Schlüssel  $\tilde{O}_B$  des Empfängers B, den er aus dessen Zertifikat  $Z_B$  entnimmt.
- Der Empfänger B entschlüsselt die verschlüsselten und signierten Daten mit seinem privaten Schlüssel  $P_B$ .
- Der Empfänger B prüft die Echtheit der signierten Daten mit dem öffentlichen Schlüssel  $\tilde{O}_A$  des Senders A, das er dessen Zertifikat  $Z_A$  entnimmt.

Die Verwendung einer Zertifizierungsstelle bedeutet anfangs einen erhöhten Aufwand. Jedoch entfällt dadurch bei größeren Systemen oder Maschinen die mühsame Verteilung von Zertifikaten auf die einzelnen Geräte.

**Information:**

**In Unternehmen, welche eine eigene IT-Abteilung haben, sind meistens die nötigen Voraussetzungen für eine PKI (Public Key Infrastructure) bereits vorhanden.**

OPC UA verwendet grundsätzlich Zertifikate im X.509-Format. Selbst wenn keine Zertifizierungsstelle verwendet wird, muss der öffentlichen Schlüssel  $\tilde{O}_X$  für das Gerät X in das Zertifikat  $Z_X$  verpackt und vom Gerät mit seinem private Schlüssel  $P_X$  signiert werden.

Beim initialen Verbindungsaufbau kann der Empfänger B nicht sicherstellen, ob das Zertifikat  $Z_A$  tatsächlich vom Sender A stammt, oder von einem Man-in-the-Middle, und muss diesem Zertifikat blind vertrauen. Das ist der Grund für die Warnung, wenn man sich mit einem Programm wie UaExpert zum ersten Mal auf ein Gerät verbinden.

## 8.2 Benutzerzugriffe

### Zugriffsrechte zuweisen

Der Bus Controller verfügt über ein Rechte- und Rollensystem, das festlegt, welche Aktionen ein angemeldeter Benutzer hat. In der Regel sind nicht alle Benutzer gleichberechtigt.

Vielmehr ist es üblich, nur einen oder mehrere Administratoren zu definieren, die sensitive Einstellungen am Bus Controller vornehmen dürfen.

### Benutzer identifizieren

Der Zugriff auf den Bus Controller erfolgt in der Regel authentifiziert. Die Identifizierung erfolgt entweder mit ihrem Benutzernamen und Passwort oder, im Falle von NETCONF, mittels einen SSH-Schlüssel.

Lediglich der erste Zugriff auf den Bus Controller im Konfigurationsmodus erfolgt anonym, da zu diesem Zeitpunkt noch keine bekannten Benutzer am Bus Controller existieren und diese erst angelegt werden müssen.

### Rollenzuweisung

OPC UA bietet 8 "bekannte Rollen", die unter dem Knoten *Root/Objects/Server/ServerCapabilities/RoleSet* aufgeführt sind. Benutzern können eine oder mehrere dieser Rollen zugewiesen werden (siehe "[SecurityAdmin-Rolle zuweisen](#)"). Jeder Knoten im Informationsmodell hat die Attribute *RolePermissions* und *UserRolePermissions*. *UserRolePermissions* zeigt die Berechtigungen für die Rollen eines Benutzers für diesen Knoten an. SecurityAdmins haben die Berechtigung, das Attribut *RolePermissions* zu lesen, das die Berechtigungen aller Rollen auf dem Knoten anzeigt.

### Beispiel für mögliche Werte der Attribute *RolePermissions* und *UserRolePermissions*:

Attributes	
Attribute	Value
WriteMask	0
UserWriteMask	0
RolePermissions	RolePermissionType Array[8]
[0]	RolePermissionType
> RoleId	i=15644 [WellKnownRole_Anonymous]
Permissions	None
[1]	RolePermissionType
> RoleId	i=15656 [WellKnownRole_AuthenticatedUser]
Permissions	None
[2]	RolePermissionType
> RoleId	i=15668 [WellKnownRole_Observer]
Permissions	Browse, Read, ReceiveEvents
[3]	RolePermissionType
> RoleId	i=15704 [WellKnownRole_SecurityAdmin]
Permissions	Browse, ReadRolePermissions, WriteRolePermissions, Read
UserRolePermissions	RolePermissionType Array[2]
[0]	RolePermissionType
> RoleId	i=15716 [WellKnownRole_ConfigureAdmin]
Permissions	Browse, Read, Write, ReceiveEvents, Call
[1]	RolePermissionType
> RoleId	i=15704 [WellKnownRole_SecurityAdmin]
Permissions	Browse, ReadRolePermissions, WriteRolePermissions, Read
AccessRestrictions	BadAttributeIdInvalid (0x80350000)

Die Knoten im Informationsmodell sind Gruppen zugeordnet. Alle Knoten innerhalb einer Gruppe haben die selben Berechtigungseinstellungen. Die Berechtigungseinstellungen sind fest eingestellt und nicht änderbar.

Die folgende Tabelle zeigt die möglichen Zugriffsrechte der Rollen für die Knoten innerhalb der verschiedenen Gruppen:

Gruppe	Knotenpfad	Rolle	Berechtigungen					
			B <sup>1)</sup>	R <sup>2)</sup>	RE <sup>3)</sup>	W <sup>4)</sup>	C <sup>5)</sup>	RP <sup>6)</sup>
Default	Alle Knoten, die in keiner der anderen Gruppen untergeordnet sind	Anonymous						
		AuthenticatedUser						
		Observer	✓	✓	✓			
		Operator	✓	✓	✓			
		Engineer	✓	✓	✓	✓	✓	
		Supervisor	✓	✓	✓		✓	
		ConfigureAdmin	✓	✓	✓	✓	✓	
		SecurityAdmin	✓	✓				✓
Security	Server/ServerConfiguration/* Server/ServerCapabilities/RoleSet/* Server/ServerCapabilities/UserSet/*	Anonymous						
		AuthenticatedUser						
		Observer						
		Operator						
		Engineer	✓	✓	✓			
		Supervisor	✓	✓	✓			
		ConfigureAdmin	✓	✓	✓			
		SecurityAdmin	✓	✓	✓	✓	✓	✓
Configuration	DeviceSet/X20BC008T/Configuration/* DeviceSet/X20BC008T/X2X_IF1/Configuration/* DeviceSet/X20BC008T/X2X_IF1/SubDevices/ST[x]/Configuration/*	Anonymous						
		AuthenticatedUser						
		Observer	✓	✓	✓			
		Operator	✓	✓	✓			
		Engineer	✓	✓	✓	✓	✓	
		Supervisor	✓	✓	✓			
		ConfigureAdmin	✓	✓	✓	✓	✓	
		SecurityAdmin	✓	✓				✓
ProcessData	DeviceSet/X20BC008T/X2X_IF1/SubDevices/ST[x]/ProcessData/*	Anonymous						
		AuthenticatedUser						
		Observer	✓	✓	✓			
		Operator	✓	✓	✓	✓	✓	
		Engineer	✓	✓	✓	✓	✓	
		Supervisor	✓	✓	✓		✓	
		ConfigureAdmin	✓	✓	✓	✓	✓	
		SecurityAdmin	✓	✓				✓
User	Server/ServerCapabilities/CurrentUser/*	Anonymous						
		AuthenticatedUser						
		Observer	✓	✓	✓	✓	✓	
		Operator	✓	✓	✓	✓	✓	
		Engineer	✓	✓	✓	✓	✓	
		Supervisor	✓	✓	✓	✓	✓	
		ConfigureAdmin	✓	✓	✓	✓	✓	
		SecurityAdmin	✓	✓	✓	✓	✓	✓
SoftwareUpdate	DeviceSet/X20BC008T/FirmwareUpdate/*	Anonymous						
		AuthenticatedUser						
		Observer						
		Operator						
		Engineer	✓	✓	✓	✓	✓	
		Supervisor	✓	✓	✓			
		ConfigureAdmin	✓	✓	✓	✓	✓	
		SecurityAdmin	✓	✓	✓	✓	✓	✓
X2XConfigChannels	DeviceSet/X20BC008T/X2X_IF1/SubDevices/ST[x]/ConfigChannels/*	Anonymous						
		AuthenticatedUser						
		Observer	✓	✓	✓			
		Operator	✓	✓	✓			
		Engineer	✓	✓	✓			
		Supervisor	✓	✓	✓			
		ConfigureAdmin	✓	✓	✓			
		SecurityAdmin	✓	✓				✓

- 1) Browse
- 2) Read
- 3) ReceiveEvent
- 4) Write
- 5) Call
- 6) ReadRolePermissions und WriteRolePermissions

### 8.3 Schlüsselverwaltung für NETCONF

Damit ein NETCONF-Client mit dem Bus Controller kommunizieren kann, sollten idealerweise SSH-Schlüssel verwendet werden. Grundsätzlich wäre zwar die Authentifikation über Benutzernamen und Passwort möglich, SSH-Schlüssel bieten aber bessere Sicherheit.

- Zuerst muss am Gerät, auf dem der NETCONF-Client läuft, ein SSH-Schlüsselpaar erzeugt werden. Unter Linux bzw. Windows mit Cygwin geschieht das z. B. über das Kommandozeilen-Tool `ssh-keygen`:

```
$ ssh-keygen -q -N "" -f ~/.ssh/id_rsa
```

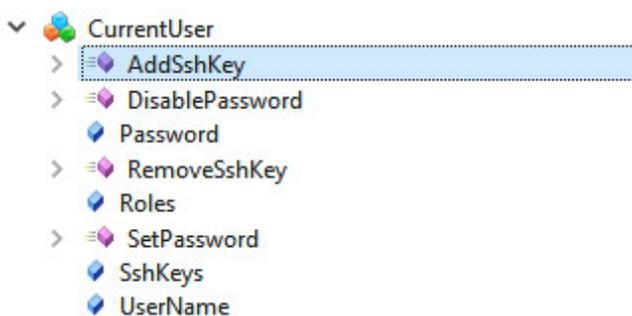
Dieser Aufruf erzeugt 2 Dateien:

```
~/.ssh/id_rsa
~/.ssh/id_rsa.pub
```

Die Datei `~/.ssh/id_rsa` enthält den privaten Schlüssel und muss geschützt am Gerät verbleiben. Die andere Datei `~/.ssh/id_rsa.pub` enthält den öffentlichen Schlüssel, der auf den Bus Controller übertragen wird. Der Inhalt dieser Datei ist eine einzelne ASCII-Textzeile der folgenden Art:

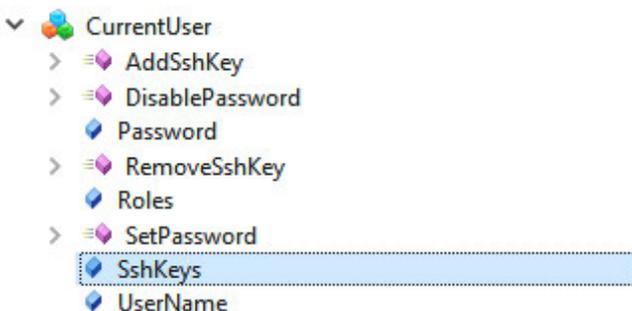
```
ssh-rsa AAAAB3NzaC1yc2EAAAADA...UmUCIxYc68QIw+OSoN admin@client
```

- Falls lediglich ein einziger Benutzer, wie der zuvor angelegte "admin" für sämtliche Verwaltungsaufgaben verwendet werden soll, kann der Schlüssel diesem Benutzer zugewiesen werden. Dazu muss die Methode `Root/Objects/Server/ServerCapabilities/CurrentUser/AddSshKey` aufgerufen und die gesamte Textzeile des öffentlichen SSH-Schlüssels hineinkopiert werden.



Falls unterschiedliche Geräte zur Bus Controller Verwaltung über NETCONF benutzt werden, kann für jedes dieser Geräte ein eigener SSH-Schlüssel hinzugefügt werden. Nicht mehr verwendete SSH-Schlüssel lassen sich analog über die Funktion `Root/Objects/Server/ServerCapabilities/CurrentUser/RemoveSshKey` wieder vom Bus Controller entfernen.

Die Liste der SSH-Schlüssel ist beim jeweiligen Benutzer zu sehen:



Value	
SourceTimestamp	09-Mar-21 11:09:14.036
SourcePicoseconds	0
ServerTimestamp	09-Mar-21 11:09:14.036
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	String Array[1]
[0]	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCSJFg

- Bei Bedarf können unterschiedliche Benutzer mit eigenen Rollen definiert werden, die z. B. für unterschiedliche Verwaltungsaufgaben zuständig sind. Neben einem allgemeinen `SecurityAdmin` für die Benutzer- und Rollenverwaltung, wäre ein weiterer `ConfigureAdmin` denkbar, der für die Verwaltung der TSN-Funktionalität des Bus Controllers zuständig ist. Dieser Benutzer würde mit dem Bus Controller ausschließlich über NETCONF kommunizieren. In diesem Fall kann dessen Passwort deaktiviert und ihm somit der Zugang über OPC UA verwehrt werden.

## 8.4 Zertifikatsmanagement

### Information:

Siehe auch [3.8 "Aktualisierung des Self-Signed Zertifikats"](#).

#### 8.4.1 Zertifikatsanforderung erzeugen

Will man Zertifikate verwenden, die von einer Zertifizierungsstelle (Certificate Authority, CA) signiert sind, dann sollte die notwendige Zertifikatsignierungsanforderung (Certificate-Signing-Request, CSR) direkt am Gerät erzeugt werden. Durch die Erzeugung des CSR am Gerät muss der private Schlüssel das Gerät nie verlassen, wodurch die Sicherheit erhöht wird. Der Prozess läuft folgendermaßen ab:

- Zum Erzeugen eines CSR gibt es unter *Root/Objects/Server/ServerConfiguration* die Methode *CreateSigningRequest*. Beim Aufruf der Methode wird optional ein neuer privater Schlüssel erzeugt. Sollte die Option nicht aktiviert sein, dann wird der bestehende Schlüssel verwendet. Der öffentliche Schlüssel, die Information über den Antragsteller, sowie weitere Informationen werden in einen "PKCS #10 DER" codierten Certificate-Request verpackt, der von der Methode zurückgegeben wird. Der Bytestring muss in eine entsprechende ".csr"-Datei gespeichert werden.
- Der CSR muss im Anschluss von einer Zertifizierungsstelle signiert werden, wobei noch zusätzliche Informationen in das Zertifikat eingetragen werden. Das Ergebnis ist ein gültiges Zertifikat.
- Das signierte Zertifikat kann im dann über die Methode *Root/Objects/Server/ServerConfiguration/UpdateCertificate* installiert werden.

### Information:

- **Der private Schlüssel für den CSR bleibt nur so lange am Gerät hinterlegt, bis ein neuer CSR generiert wird oder bis das Gerät neu gestartet wird. Ein signiertes Zertifikat kann nur dann installiert werden, wenn der dazu gehörige private Schlüssel noch vorhanden ist.**
- **UaExpert bietet im GDS Push View eine vereinfachte Möglichkeit den CSR zu erzeugen und zu Speichern. Dadurch muss nicht direkt mit der Methode gearbeitet werden.**

Weiterführende Details zur OPC UA Methode *CreateSigningRequest* finden sich in der OPC UA Spezifikation, Teil 12

#### 8.4.2 Zertifikat mittels UpdateCertificate aktualisieren

Über die Methode *Root/Objects/Server/ServerConfiguration/UpdateCertificate* können signierte Zertifikate auf dem Bus Controller installiert werden. Dabei macht es keinen Unterschied, ob es sich um ein von einer Zertifizierungsstelle signiertes Zertifikat oder um ein selbstsigniertes Zertifikat handelt. Wenn das Zertifikat nicht aus einem CSR erzeugt wurde der vom Bus Controller generiert wurde, dann muss zusätzlich der private Schlüssel übergeben werden.

Damit die Änderungen übernommen werden, muss zusätzlich die Methode *Root/Objects/Server/ServerConfiguration/ApplyChanges* aufgerufen werden. Dabei werden alle verbundenen Clients getrennt. Eine neue Verbindung ist erst wieder möglich, wenn dem neuen Zertifikat vertraut wird.

### Information:

- **Da beim Aufruf dieser Methode möglicherweise ein privater Schlüssel übertragen wird, ist der Aufruf nur möglich, wenn eine verschlüsselte Verbindung zwischen Bus Controller und OPC UA Client besteht.**
- **UaExpert bietet im GDS Push View eine vereinfachte Möglichkeit Zertifikate zu aktualisieren. Dadurch muss nicht direkt mit der Methode gearbeitet werden.**
- **Zertifikate, die von anderen Zertifikaten abgeleitet sind, können nur installiert werden wenn alle übergeordneten Zertifikate bereits installiert wurden, (siehe [8.1.3 "Symmetrische und asymmetrische Schlüssel"](#)) sodass die vollständige Vertrauenskette überprüft werden kann.**

Weiterführende Details zur OPC UA Methode *UpdateCertificate* finden sich in der OPC UA Spezifikation, Teil 12.

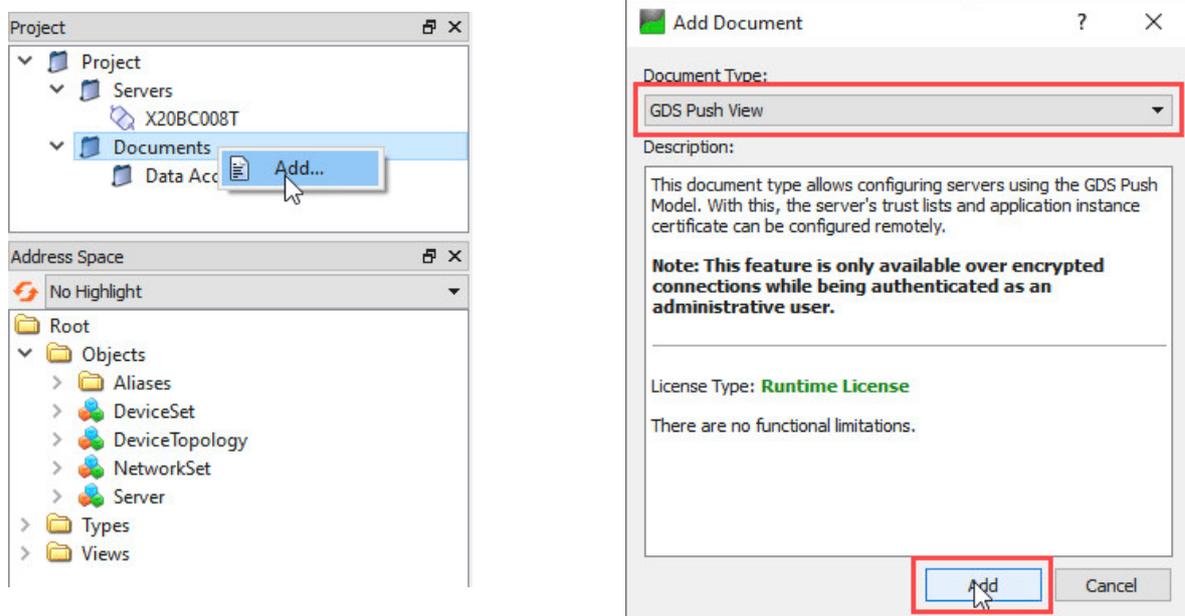
### 8.4.2.1 Aktualisierung des selbstsignierten Zertifikats mittels UaExpert

UaExpert verfügt über Werkzeuge mit dessen Hilfe Zertifikate auf einfache Weise aktualisiert werden können.

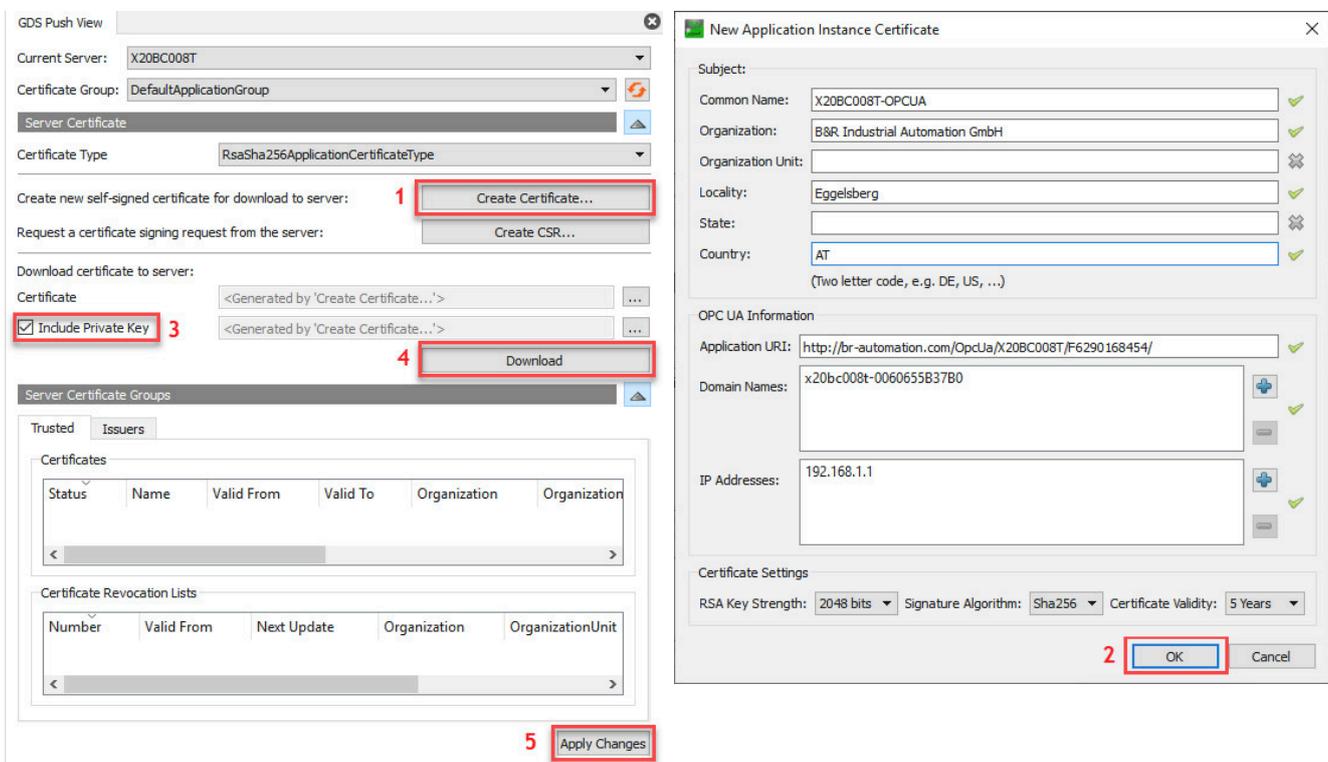
#### Information:

Da beim folgenden Ablauf ein privater Schlüssel übertragen wird funktioniert er nur, wenn eine verschlüsselte Verbindung zum Bus Controller besteht.

- Im Projektfenster des UaExpert im Kontextmenü von *Documents* auf *Add...* klicken. Ein Dialog öffnet sich. In diesem Dialog den Dokument-Typ "GDS Push View" auswählen und durch Klick auf *Add* bestätigen.



- Das Zertifikat erstellen und übertragen.



- 1) Klick auf *Create Certificate*
- 2) Im folgenden Dialog werden die für das Zertifikat erforderlichen Daten eingegeben.

### **Information:**

**Das Eintragen der IP-Adresse ist nur notwendig, wenn die IP-Adresse statisch vergeben ist und Clients mit Hilfe der IP-Adresse auf den Bus Controller zugreifen (z. B. über die URL `opc.tcp://192.168.1.1:4840`). Wird die IP-Adresse über einen DHCP-Server bezogen, dann ist es nicht sinnvoll eine IP-Adresse in das Zertifikat einzutragen, da diese in der Regel dynamisch zugeteilt wird und sich ändern kann.**

- 3) Da für die Aktualisierung der private Schlüssel mit übertragen werden muss, ist die Option "Include Private Key" auszuwählen.
- 4) Durch Klick auf *Download* wird das vorher erstellte Zertifikat auf den Bus Controller übertragen. Die folgende Abfrage, ob Issuer-Zertifikate spezifiziert werden sollen kann mit "Nein" bestätigt werden.
- 5) Durch Klick auf *ApplyChanges* wird das neue Zertifikat übernommen. Dabei werden alle verbundenen Clients getrennt. Eine neue Verbindung ist erst wieder möglich, wenn dem neuen Zertifikat vertraut wird.

## **9 Diagnose**

Auftretende Fehlfunktionen oder das Beobachten von unerwartetem bzw. unerwünschtem Verhalten des Bus Controllers kann vielfältige Ursachen haben. Insbesondere beim Einsatz in größeren Netzwerken im Verbund mit Netzwerkinfrastruktur unterschiedlicher Hersteller, gestaltet sich oft bereits die Lokalisierung möglicher Fehlerquellen schwierig. Das vorliegende Kapitel soll als Hilfestellung bei der Diagnose von Fehlfunktionen und der Lokalisierung ihrer Ursachen dienen. Es beschreibt kontextbezogene Fehler und zeigt mögliche Ursachen und deren Lösung auf. Diese umfassen:

- Fehler im Kontext der Adressierung
- Fehler im Kontext der Datenübertragung
- Fehler im Kontext der Zeitsynchronisierung
- Fehler im Kontext von Cyber-Security

## 9.1 Adressierung

Nr.	Fehlerbild	Mögliche Ursache	Lösung	Siehe
1	Verbindung über Hostname im Werkzustand nicht möglich.	Hostname unbekannt	Der Bus Controller ist standardmäßig unter dem mDNS-Hostnamen "x20bc008t-<MAC-Adresse>.local" erreichbar <sup>1)</sup> .	- 3.2 "Verbindungsaufbau" - 6.1.2 "Allgemeine Netzwerkkonfiguration"
2	Verbindung über IP-Adresse im Werkzustand nicht möglich.	IP-Adresse unbekannt	<p>• <b>DHCP-Server ist im Netzwerk vorhanden</b><sup>2)</sup>: Die dem Bus Controller zugewiesene IP-Adresse beim zuständigen Netzwerkadministrator in Erfahrung bringen.</p> <p>• <b>DHCP-Server ist im Netzwerk nicht vorhanden</b>: Dem Bus Controller zum Betrieb mit Hilfe des Knotennummerschalters die statische IP-Adresse 192.168.1.1 zuweisen (Verstellen des Knotennummerschalters für 1 s) und anschließend eine benutzerdefinierte Konfiguration über das OPC UA Informationsmodell durchführen.</p> <p>Falls der Bus Controller direkt mit einem anderen Switch verbunden ist, welcher bereits über Hostname oder IP-Adresse erreichbar ist, dann kann die gesuchte IP-Adresse von dem Port ausgelesen werden, an dem der Bus Controller angeschlossen ist. <i>Root/Objects/DeviceSet/X20BC008T/Status/SwitchPorts/ETHx/LinkPartner/ManagementAddress</i></p>	- 7.1 "Port-Status"  - 2.2.2 "Nummernschalter" - 2.3 "IP-Adresse einstellen" - 6.1.2 "Allgemeine Netzwerkkonfiguration"
3	In einem Netzwerk mit DHCP-Server ist die Verbindung zum Bus Controller über IP-Adresse nicht möglich, nachdem eine statische IP-Adresse per Konfiguration über das OPC UA Informationsmodell gesetzt wurde.	Es besteht ein IP-Adressenkonflikt mit einem anderen Gerät im Netzwerk, das dieselbe Adresse dynamisch vom DHCP-Server bezogen hat.	<p>Überprüfen, ob die statisch zugewiesene IP-Adresse außerhalb des vom DHCP-Server verwalteten Bereichs liegt.</p> <p>• <b>Außerhalb des Bereichs</b>: Alle Geräte mit statischen Adresseinstellungen auf Adresskonflikte überprüfen.</p> <p>• <b>Innerhalb des Bereichs</b>: Standard-Adresse 192.168.1.1 wiederherstellen und eine statischen IP-Adresse, die außerhalb des vom DHCP-Server verwalteten Bereichs liegt, konfigurieren.</p>	- 2.2.2 "Nummernschalter" - 2.3 "IP-Adresse einstellen" - 6.1.2 "Allgemeine Netzwerkkonfiguration"
		Es wurde bei der Konfiguration über das OPC UA Informationsmodell eine Netzwerkmaske eines Subnetzes vergeben, das vom Client aus nicht erreicht werden kann.	<p>Einstellungen des TCP/IP-Stacks auf dem Betriebssystem des Clients überprüfen.</p> <p>• <b>Innerhalb desselben Subnetzes</b>: Befindet sich der Client im selben Subnetz, müssen die konfigurierten Netzwerkmasken am Client und am Bus Controller identisch sein.</p> <p>• <b>Nicht innerhalb desselben Subnetzes</b>: Befindet sich der Client nicht im selben Subnetz, sind die Routing-Einstellungen des Betriebssystems des Clients zu prüfen.</p>	- 6.1.2 "Allgemeine Netzwerkkonfiguration"

### 1) Information

- Eine Adressierung des Geräts über den oben genannten Hostname erfordert eine entsprechende mDNS-Unterstützung auf dem Betriebssystem des zugreifenden Clients.
- Die zu verwendende MAC-Adresse im Hostname des Bus Controllers, ist dem seitlich am Bus Controller angebrachten Etikett zu entnehmen ("MAC1"), welches auch die Seriennummer enthält.

Ein benutzerdefinierter Hostname ist über das OPC UA Informationsmodell konfigurierbar.

- 2) Der Bus Controller wird standardmäßig mit aktiviertem DHCP-Client ausgeliefert und eine IP-Adresse muss von einem DHCP-Server im Netzwerk zugewiesen werden.

## 9.2 Datenübertragung

Nr.	Fehlerbild	Mögliche Ursache	Lösung	Siehe
1	Kompletter Kommunikationsausfall zwischen 2 oder mehr an den Bus Controller angeschlossenen Geräten <sup>1)</sup> .	Ethernet Auto-Negotiation zwischen Bus Controller und einem oder mehreren angeschlossenen Geräten ist fehlgeschlagen.	<p><i>LinkStatus</i> der betroffenen Ports im OPC UA Informationsmodell prüfen:  <i>Root/Objects/DeviceSet/X20BC008T/Status/SwitchPorts/ETHx/LinkProperties/LinkStatus</i></p> <p>Steht dieser Wert nicht auf <i>UP</i>, besteht auf Netzwerkebene kein Link zwischen Bus Controller und angeschlossenem Gerät.  Kabelverbindung kontrollieren:</p> <ul style="list-style-type: none"> <li>- Fester Sitz der Steckverbindung</li> <li>- max. Länge (100m) des Kabels nicht überschritten</li> <li>- Eventuelle Kabelschäden</li> </ul>	- 7.1 "Port-Status"
			Eventuell vorhandene Log-Ausgaben des Bus Controllers bzw. des angeschlossenen Geräts kontrollieren. Diese können Aufschluss über Hardwareprobleme geben.	
		An der Kommunikation beteiligte Geräte senden keine oder fehlerhafte Ethernet-Frames, die vom Bus Controller erkannt und verworfen werden.	<p>Fehlerzähler der betroffenen Ports im OPC UA Informationsmodell prüfen:  <i>Root/Objects/DeviceSet/X20BC008T/Status/SwitchPorts/ETHx/FrameStatistics/</i></p> <ul style="list-style-type: none"> <li>• <i>RxFrameCount</i> und <i>TxFrameCount</i>  Weisen diese Zähler den Wert 0 auf, werden von den angeschlossenen Geräten keine oder fehlerhafte Ethernet-Frames versendet.</li> <li>• <i>FcsErrorFrameCount</i>, <i>GeneralRxErrorFrameCount</i>, <i>GeneralTxErrorFrameCount</i> und <i>SizeErrorFrameCount</i>  Bei fehlerfreiem Betrieb weisen diese Zähler den Wert 0 auf. Werte ungleich 0 deuten in den meisten Fällen auf Fehler in Netzwerkkomponenten der angeschlossenen Geräte oder dem Bus Controller selbst hin.</li> </ul>	
	Ethernet-Frames die mit einem VLAN-Tag versehen sind und/oder Multicast DMAC-Adressen wurden nicht am Bus Controller konfiguriert.	Konfiguration zur Weiterleitung von Ethernet-Frames mit VLAN-Tag und/oder Multicast DMAC-Adressen im Konfigurationstool überprüfen.		
2	Datenempfang an einem an den Bus Controller angeschlossenen Empfänger erfolgt nicht zum erwarteten Zeitpunkt <sup>2)</sup> .	Link-Geschwindigkeit zwischen Bus Controller und angeschlossenem Gerät entspricht nicht Erwartungshaltung (100 Mbit/s statt 1 Gbit/s).	<p>Knoten <i>Speed</i> und <i>Duplex</i> der betroffenen Ports im OPC UA Informationsmodell prüfen.  <i>Root/Objects/DeviceSet/X20BC008T/Status/SwitchPorts/ETHx/LinkProperties/Speed</i> bzw. <i>Duplex</i></p> <p>Entspricht die Geschwindigkeit oder der Duplex-Modus nicht der Erwartungshaltung, dann folgende Punkte überprüfen:</p> <ul style="list-style-type: none"> <li>- Unterstützt das angeschlossene Gerät die erwartete Geschwindigkeit/Duplex-Modus?</li> <li>- Wurde der korrekte Kabeltyp verwendet?  (Für Gigabit-Ethernet ist mindestens Cat 5 erforderlich)</li> <li>- Kabelverbindung kontrollieren: <ul style="list-style-type: none"> <li>• Fester Sitz der Steckverbindung</li> <li>• Eventuelle Kabelschäden</li> <li>• Verlegung in der Nähe von potenziellen Einstreuungsquellen</li> </ul> </li> </ul>	- 7.1 "Port-Status"
		Duplex-Mode zwischen Bus Controller und angeschlossenem Gerät entspricht nicht Erwartungshaltung (Half-Duplex anstatt Full-Duplex).		
		Geräte wurden nicht an den laut Systemdesign vorgesehenen Bus Controller-Ports angeschlossen. Dadurch entsprechen die Übertragungslatenzen nicht der Erwartungshaltung.	<p>Knoten unter <i>LinkPartner</i> der betroffenen Ports im OPC UA Informationsmodell prüfen.  <i>Root/Objects/DeviceSet/X20BC008T/Status/SwitchPorts/ETHx/LinkPartner/</i></p>	

1) Beispielsweise, wenn Nachrichten eines OPC UA Publishers an einem OPC UA Subscriber nicht empfangen werden.

2) Beispielsweise, wenn ein OPC UA Subscriber verspätet eingetroffene oder ausgefallene Ethernet-Frames feststellt.

### 9.3 Zeitsynchronisierung

Nr.	Fehlerbild	Mögliche Ursache	Lösung	Siehe
1	Die PTP-Zeitsynchronisation am Bus Controller angeschlossener Geräte funktioniert nicht.	Die PTP-Zeitsynchronisation am Bus Controller ist nicht aktiviert.	<p>PTP-Zeitsynchronisation im OPC UA Informationsmodell aktivieren<sup>1)</sup>.</p> <ul style="list-style-type: none"> <li>● <b>WallClock:</b> <i>Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WallClock/TimeSyncProtocol</i></li> <li>● <b>WorkingClock:</b> <i>Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WorkingClock/TimeSyncProtocol</i></li> </ul>	- 6.1.4 "Zeitsynchronisation"
		Es wurde die Zeitsynchronisation der falschen PTP-Domäne am Bus Controller oder dem angeschlossenen Gerät konfiguriert.	<p>Identische PTP-Domäne an allen beteiligten Geräten konfigurieren<sup>2)</sup>. Am Bus Controller kann diese Einstellung im OPC UA Informationsmodell erfolgen.</p> <ul style="list-style-type: none"> <li>● <b>WallClock:</b> <i>Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WallClock/PTP/DomainNumber</i></li> <li>● <b>WorkingClock:</b> <i>Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WorkingClock/PTP/DomainNumber</i></li> </ul>	- 6.1.4 "Zeitsynchronisation"
		Es gibt keinen PTP-Grandmaster im Netzwerk.	<p>Wenn der Bus Controller als PTP Grandmaster fungieren sollte, Option <i>SlaveOnly</i> im OPC UA Informationsmodell deaktivieren.</p> <ul style="list-style-type: none"> <li>● <b>WallClock:</b> <i>Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WallClock/PTP/SlaveOnly</i></li> <li>● <b>WorkingClock:</b> <i>Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WorkingClock/PTP/SlaveOnly</i></li> </ul>	- 6.1.4 "Zeitsynchronisation"
2	Der Bus Controller sollte PTP-Grandmaster sein, es wurde aber ein anderes Netzwerkgerät ausgewählt.	Die Priorität der PTP-Uhr des Bus Controllers ist im Vergleich zum gewählten Netzwerkgerät zu niedrig.	<p>Die <i>Priority1</i> Einstellung der PTP-Uhr im OPC UA Informationsmodell anpassen. Je niedriger der Wert eingestellt wird, desto höher ist die Priorität.</p> <ul style="list-style-type: none"> <li>● <b>WallClock:</b> <i>Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WallClock/PTP/Priority1</i></li> <li>● <b>WorkingClock:</b> <i>Root/Objects/DeviceSet/X20BC008T/Configuration/TimeSynchronization/WorkingClock/PTP/Priority1</i></li> </ul>	- 6.1.4 "Zeitsynchronisation"
3	Datenempfang an einem anderen Bus Controller angeschlossenen Empfänger erfolgt nicht zum erwarteten Zeitpunkt <sup>3)</sup> .	Die an der Kommunikation beteiligten Geräte sind nicht, oder nicht den Anforderungen entsprechend, zeitsynchronisiert.	<p>Überprüfen der korrekten Zeitsynchronisation der PTP-Domäne der WorkingClock aller Netzwerkgeräte zwischen Sender und Empfänger. Der Zustand der Zeitsynchronisation des Bus Controllers kann im OPC UA Informationsmodell überprüft werden:</p> <p><i>Root/Objects/DeviceSet/X20BC008T/Status/TimeSynchronization/WorkingClock/PTP</i></p>	- 7.2 "Zeitsynchronisation"

- 1) Standardmäßig ist am Bus Controller die Zeitsynchronisation deaktiviert.
- 2) Standardmäßig wird die WallClock über Domäne 0 und die WorkingClock über die Domäne 20 synchronisiert.
- 3) Beispielsweise, wenn ein OPC UA Subscriber verspätet eingetroffene oder ausgefallene Ethernet-Frames feststellt.

## 9.4 Cyber-Security

Nr.	Fehlerbild	Mögliche Ursache	Lösung	Siehe
1	Der Aufbau einer sicheren Verbindung zum OPC UA Server des Bus Controllers wird mit Hinweis auf ein abgelaufenes Zertifikat abgelehnt.	Gültigkeitsbereich des vom Client verwendeten Zertifikats liegt außerhalb des aktuellen Datums bzw. der aktuellen Uhrzeit des Bus Controllers <sup>1)</sup> .	<ul style="list-style-type: none"> <li>• <b>Bei Verwendung von NTP</b> Sicherstellen, dass mindestens einer der konfigurierten NTP-Server erreichbar ist und die korrekte Uhrzeit verteilt.</li> <li>• <b>Bei Verwendung von PTP</b> Sicherstellen, dass der PTP-Grandmaster aktiv ist und mit der entsprechenden PTP-Domäne (für die WallClock) die korrekte Uhrzeit verteilt.</li> </ul> <p>Ist keine Quelle für die WallClock vorhanden oder funktional, Sicherheitseinstellungen des Geräts zurücksetzen. Nur in diesem Zustand kann eine unverschlüsselte Verbindung hergestellt werden.</p>	<ul style="list-style-type: none"> <li>- 3.5 "Allgemeine Netzwerkeinstellungen über OPC UA"</li> <li>- 5.4 "Zeitsynchronisation und Zeitdomänen"</li> <li>- 6.1.4 "Zeitsynchronisation"</li> <li>- 7.1 "Port-Status"</li> <li>- 2.2.2 "Nummernschalter"</li> </ul>

- 1) Maßgeblich ist hierfür der Wert der WallClock.  
Beim sicheren Verbindungsaufbau unter Verwendung von SSL/TLS, erfolgt sowohl Client- als auch Serverseitig eine Überprüfung der verwendeten Zertifikate.

## 10 Lizenzen

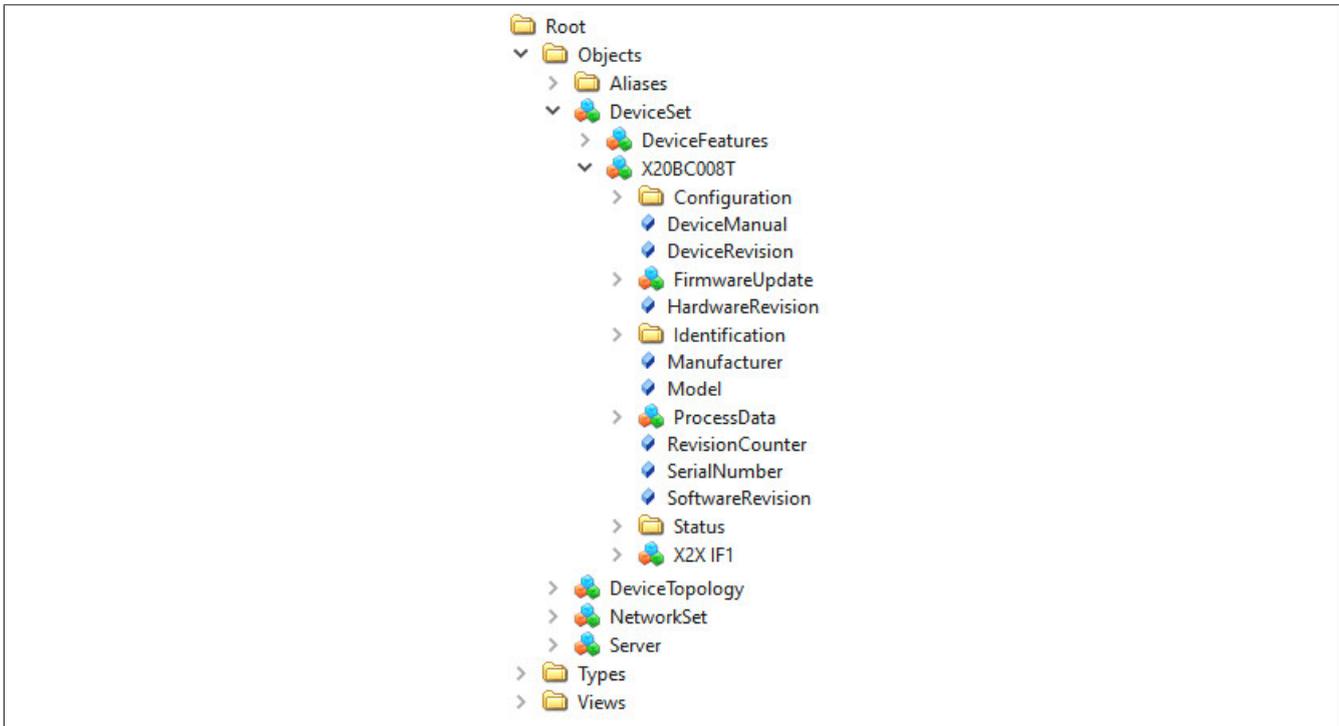
Mit Hilfe der Firmware-Upgrades, welche von der B&R Homepage ([www.br-automation.com](http://www.br-automation.com)) herunter geladen werden können, ist es möglich die Lizenzinformationen abzurufen.

1. Firmwareupgrade (ZIP-Datei) des Moduls von B&R Homepage herunterladen.
2. Das Firmwareupgrade in einen neuen Ordner entpacken.  
Es sollte danach eine ZIP-Datei licenses.zip vorhanden sein.
3. Die ZIP-Datei entpacken.  
Aus technischen Gründen können in der ZIP-Datei Dateien mit gleichem Namen enthalten sein. Dies sollte beim Entpacken der ZIP-Datei beachtet werden.
4. Nach dem Entpacken können die Lizenzdateien im Ordner ...*licenses* eingesehen werden.

# 11 Anhang

## 11.1 OPC UA Informationsmodell

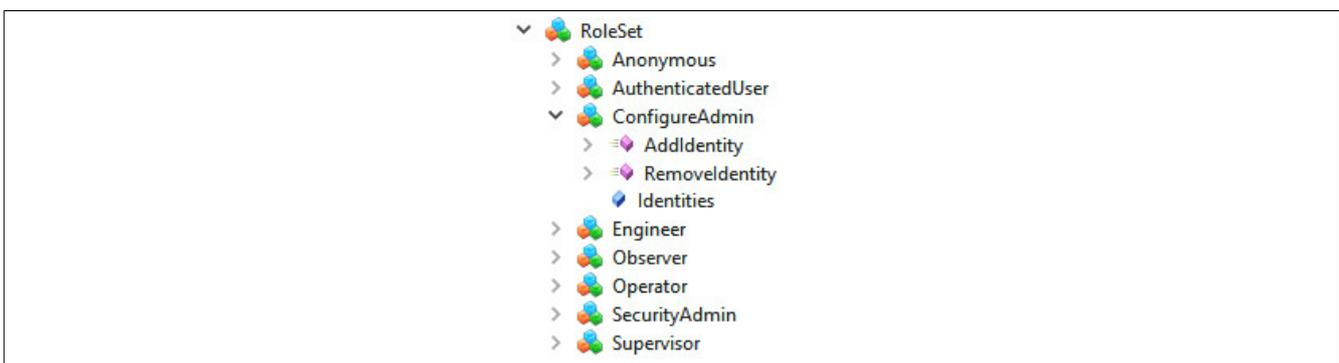
Der Bus Controller bietet über das OPC UA Informationsmodell Zugang zur Konfiguration und Daten der I/O-Module und des Bus Controllers. Auch OPC UA Clients können sich über dieses Informationsmodell Zugriff auf die vorhandenen Daten verschaffen.



Dem Hauptknoten *Root/Objects/DeviceSet/X20BC008T* sind dabei über hierarchische Referenzen alle Knoten untergeordnet, die für den Bus Controller verfügbar sind. Dies beinhaltet unter anderem Knoten für die Konfiguration und den Zugriff auf die Prozessdaten der I/O-Module.

### 11.1.1 Benutzerverwaltung

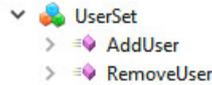
Der Zugriff auf den Bus Controller ist im normalen Betrieb auf autorisierte Benutzer beschränkt. Benutzer wiederum haben unterschiedliche Rechte, entsprechend den ihnen zugewiesenen Rollen. Die dafür nötige Benutzer- und Rollenverwaltung erfolgt über das OPC UA Informationsmodell.



Der Bus Controller kommt mit vordefinierten, standardisierten Rollen. Die Rollen unterscheiden sich strukturmäßig im Informationsmodell nicht, sondern besitzen alle die gleichen Methoden und Attribute. Beispielhaft wird hier die für Administrationsaufgaben zuständige Rolle *ConfigureAdmin* dargestellt.

Position der Daten im Informationsmodell: *Root/Objects/Server/ServerCapabilities/RoleSet*

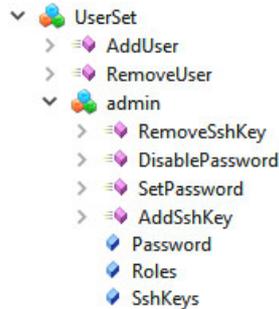
Knotenname	Beschreibung
AddIdentity	Hinzufügen eines Benutzers zu dieser Rolle
RemoveIdentity	Entfernen eines Benutzers aus dieser Rolle
Identities	Liste aller Benutzer in dieser Rolle



In der Werkseinstellung kommt der Bus Controller ohne vordefinierte Benutzer. Diese können frei vergeben werden, bis auf wenige reservierte Namen, wie z. B. "root". Diese erscheinen dann unterhalb des *UserSet*-Objekts.

Position der Daten im Informationsmodell: *Root/Objects/Server/ServerCapabilities/UserSet*

Knotenname	Beschreibung
AddUser	Hinzufügen eines Benutzers
RemoveUser	Entfernen eines Benutzers



Angelegte Benutzer unterscheiden sich strukturmäßig im OPC UA Informationsmodell nicht, sondern besitzen alle die gleichen Methoden und Attribute. Beispielhaft wird hier der häufig verwendete Benutzer "admin" dargestellt.

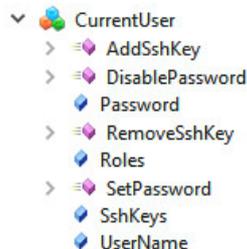
Benutzer können sich am Bus Controller auf unterschiedliche Weise authentifizieren. Der Zugriff über OPC UA wird durch Passwörter, der Zugriff über NETCONF hingegen über SSH-Schlüssel gesichert. Es ist erlaubt, beide Arten gleichzeitig zu aktivieren.

Ein Benutzer kann höchstens ein Passwort haben. Es ist sinnvoll, ein Passwort von hinreichender Komplexität zu wählen. Der Bus Controller überprüft allerdings das eingestellte Passwort nicht, wodurch auch das leere Passwort "" möglich ist.

Ein Benutzer kann beliebig viele SSH-Schlüssel haben. Das bietet sich an, wenn der Zugriff auf den Bus Controller von unterschiedlichen Geräten aus erwünscht ist. Im Gegensatz zu Passwörtern sind SSH-Schlüssel grundsätzlich sicher und nicht zu erraten. Der Bus Controller erlaubt die Verwendung von SSH-Schlüsseln allerdings nur für NETCONF. Der OPC UA Standard unterstützt SSH nicht.

Position der Daten im Informationsmodell: *Root/Objects/Server/ServerCapabilities/UserSet*

Beschreibung	Knotenname
Roles	Liste aller Rollen, die der Benutzer inne hat
Password	Anzeige, ob Passwortauthentifizierung für diesen Benutzer aktiv ist
SetPassword	Setzen eines Passworts
DisablePassword	Löschen des Passworts und Deaktivierung der Passwortauthentifizierung dieses Benutzers
SshKeys	Liste der öffentlichen SSH-Schlüssel
AddSshKey	Hinzufügen eines öffentlichen SSH-Schlüssels
RemoveSshKey	Entfernen eines öffentlichen SSH-Schlüssels



Der Zugriff auf die allgemeine Benutzer- und Rollenverwaltung ist beschränkt auf privilegierte Benutzer. Jeder Benutzer hat dagegen die nötigen Berechtigungen, um z. B. das eigene Passwort zu ändern. Der aktuelle Benutzer der Sitzung ist im Informationsmodell extra repräsentiert; dieser spiegelt dynamisch die Attribute und Methoden eines auch über die allgemeine Benutzerverwaltung erreichbaren Benutzers.

Position der Daten im Informationsmodell: *Root/Objects/Server/ServerCapabilities/CurrentUser*

Knotenname	Beschreibung
Roles	Liste aller Rollen, die der Benutzer dieser Sitzung inne hat
Password	Anzeige, ob Passwortauthentifizierung für diesen Benutzer aktiv ist
SetPassword	Setzen eines Passworts
DisablePassword	Löschen des Passworts und Deaktivierung der Passwortauthentifizierung dieses Benutzers

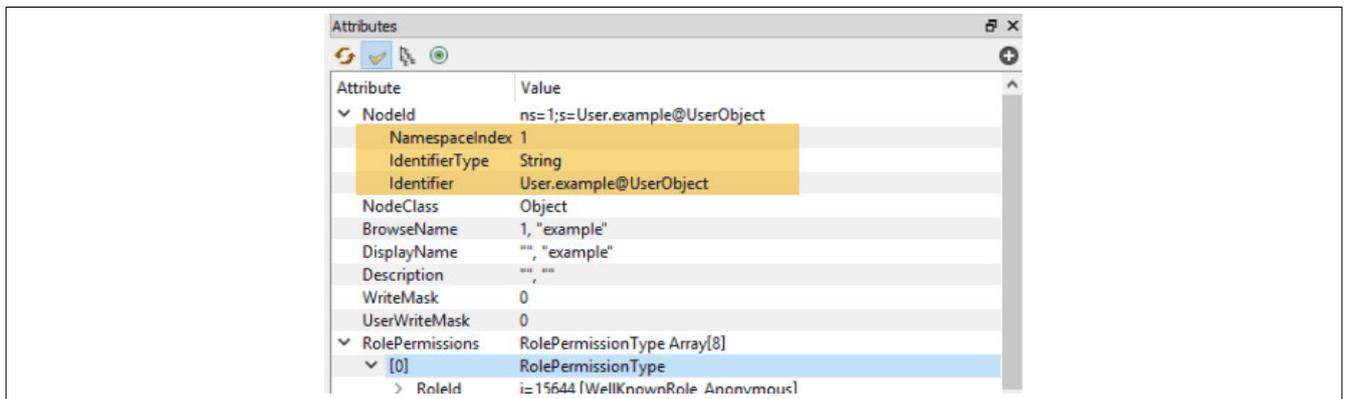
Knotenname	Beschreibung
SshKeys	Liste der öffentlichen SSH-Schlüssel
AddSshKey	Hinzufügen eines öffentlichen SSH-Schlüssels
RemoveSshKey	Entfernen eines öffentlichen SSH-Schlüssels
UserName	Name des aktuellen Benutzers der Sitzung

### 11.1.1.1 Angelegten Benutzer löschen

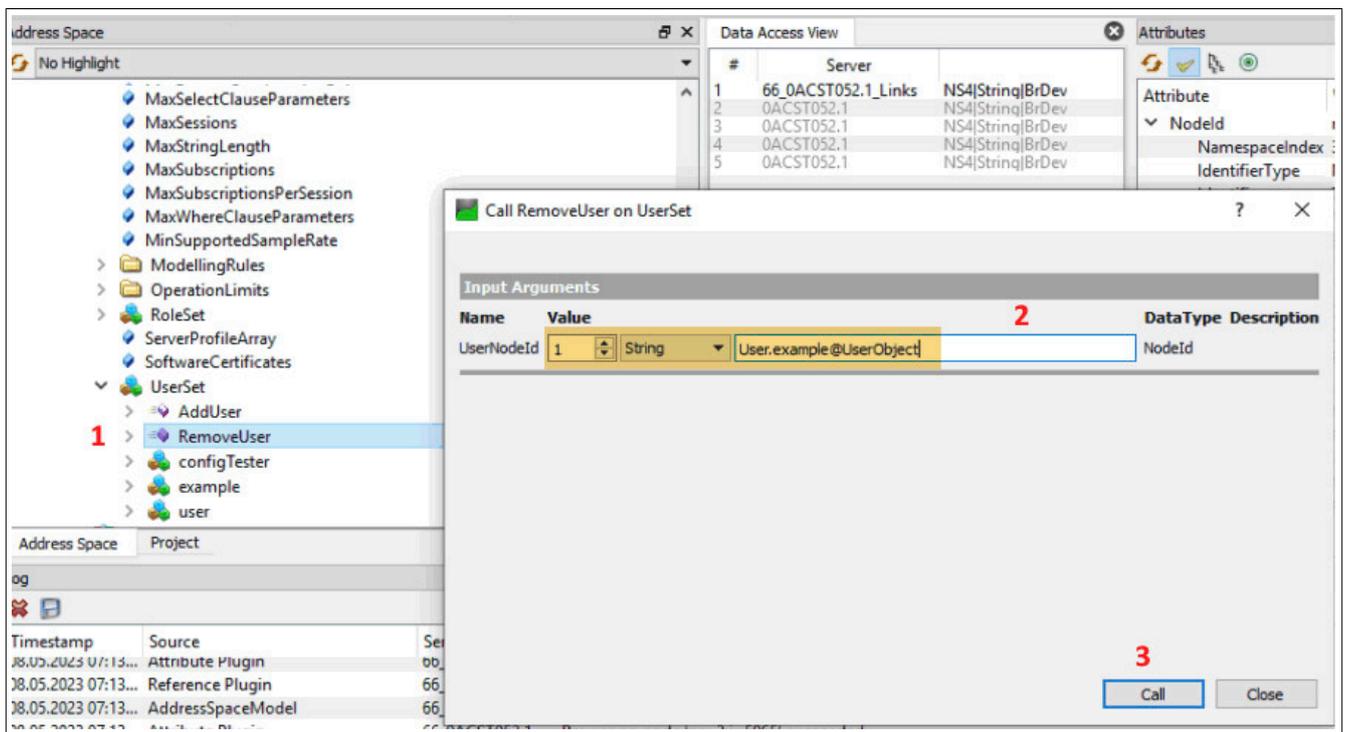
Um bereits angelegte Benutzer zu löschen, muss folgendes beachtet werden:

- Nur ein Benutzer mit *SecurityAdmin*-Rechten kann einen Benutzer löschen
- Benutzer können sich nicht selbst löschen

In diesem Beispiel soll der Benutzer "Example" gelöscht werden.



- 1) Zuerst muss die Methode 11.1.1 "RemoveUser" aufgerufen werden.
- 2) Als Eingangsargumente werden der *NamespaceIndex* und der *Identifier* des Benutzers übergeben
- 3) Zuletzt wird der Benutzer durch Klick auf "Call" gelöscht.



## 11.1.2 Firmwareupdate

Die Firmwareupdate-Funktionalität wird im OPC UA Informationsmodell dem Bus Controller durch den Knoten *Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate* bereitgestellt. Die folgende Tabelle beschreibt die Hierarchie aller Unterknoten und deren Bedeutung:

Knotenname	Datentyp	Beschreibung
DefaultInstanceBrowseName	QualifiedName	Name des verwendeten Objekts. Festgelegter Standardname = "SoftwareUpdate"
<b>Installation</b>		
CurrentState	LocalizedText	Nutzer-lesbarer Installationsstatus Mögliche Werte Idle Installing Error
Id	Nodeld	Maschinen-lesbarer Installationsstatus Mögliche Werte 271 Idle Objekt 273 Installation Objekt 275 Error Objekt
InstallSoftwarePackage		Startet die Installation des zuvor geladenen Firmwarepakets
InputArguments	String ManufacturerUri String SoftwareRevision String[] PatchIdentifiers ByteString Hash	ManufacturerUri und SoftwareRevision; dient zur Identifikation des zu installierenden Firmwarepakets. Die Argumente <i>PatchIdentifiers</i> und <i>Hash</i> sind ohne Funktion und müssen leer bleiben. <b>Information:</b> Möchte man eine Firmware-Installation erzwingen, bei der eine identische, bereits installierte Version überschrieben werden soll, muss dem Argument <i>PatchIdentifiers[0]</i> der String "force" zugewiesen werden.
PercentComplete	Byte	Zeigt den Installationsfortschritt an. <b>Information:</b> Diese Methode ist nicht für die Erkennung einer abgeschlossenen Installation geeignet.
Resume		Setzt den Installationsstatus von "Error" zurück auf "Idle"
<b>Loading</b>		
CurrentVersion		Zeigt Eigenschaften der aktiven Firmware
Manufacturer	LocalizedText	Hersteller
ManufacturerUri	String	Hersteller-URI
SoftwareRevision	String	Version des Firmwarepakets
ErrorMessage	LocalizedText	Nutzerinformation für den Dateitransfer - siehe <a href="#">Fehlernachrichten</a>
FileTransfer		Gibt Informationen zum Dateitransfer des Firmwarepakets
ClientProcessingTimeout		Zeit in Sekunden, nach der der Server den Transfer beendet, sollte der Client keine für den Transfer erforderlichen Methodenaufrufe mehr ausführen.
CloseAndCommit		Beendet den Dateitransfer
GenerateFileForRead		Wird nicht unterstützt
GenerateFileForWrite		Generiert eine FileType Instanz, die für den Dateitransfer genutzt wird.
TransferState		Transferstatus Objekt
GetUpdateBehavior		Zeigt Update-Eigenschaften der geladenen Firmware
InputArguments	String ManufacturerUri String SoftwareRevision String[] PatchIdentifiers	ManufacturerUri und SoftwareRevision; dient zur Identifikation des vorher geladenen Firmwarepakets. Das Argument <i>PatchIdentifiers</i> ist ohne Funktion und muss leer bleiben.
OutputArguments	UInt32	Beschreibt, wie der Bus Controller ein Update durchführen kann. Mögliche Werte 4 RequiresPowerCycle
PendingVersion		Zeigt Eigenschaften der geladenen Firmware <sup>1)</sup>
Manufacturer	LocalizedText	Hersteller
ManufacturerUri	String	Hersteller-URI
SoftwareRevision	String	Version des Firmwarepakets
<b>PowerCycle</b>		
CurrentState	LocalizedText	Nutzer-lesbarer Rebootstatus Mögliche Werte NotWaitingForPowerCycle WaitingForPowerCycle
Id	Nodeld	Maschinen-lesbarer Rebootstatus Mögliche Werte 299 NotWaitingForPowerCycle Objekt 301 WaitingForPowerCycle Objekt
UpdateStatus	LocalizedText	Nutzerinformation und Rückmeldung für den gesamten Updatevorgang, siehe <a href="#">Updatestatus</a>

1) Nur wenn die entsprechende Firmwareupdate-Datei bereits auf das Zielgerät übertragen wurde und bereit zur Installation ist.

### Fehlernachrichten

Nr.	Text	Bedeutung
0	[ERROR] File invalid or not loaded	Wird angezeigt, wenn der Knoten <i>Root/Objects/DeviceSet/X20BC008T/FirmwareUpdate&gt;Loading/PendingVersion</i> ausgelesen wird, das Firmwarepaket jedoch ungültig ist oder fehlt.

## Updatestatus

Nr.	Text	Bedeutung
0	[ERROR] Requested version not present or file invalid	Eingabeparameter der Methode <i>GetUpdateBehavior</i> oder <i>InstallSoftwarePackage</i> passen nicht zum geladenen Firmwarepaket. Die angefragte Version ist nicht vorhanden, oder das Firmwarepaket ist ungültig.
1	[ERROR] Installation failed. See FirmwareIn-stall.log in system dump archive.	Die von der Methode <i>InstallSoftwarePackage</i> ausgelöste Installation des Firmwarepakets wurde gestartet, ist jedoch fehlgeschlagen. Weiterführende Informationen sind im "SystemDump" Objekt, in der Datei "FirmwareInstall.log", zu finden.
2	[ERROR] Action not allowed in current state	Methodenaufruf wurde verwehrt, da dieser im aktuellen Status nicht erlaubt ist.
3	[INFO] Installation successful, reboot required	Der Bus Controller benötigt einen Neustart, um die installierte Firmware zu aktivieren. Dieser kann durch Aufruf der Methode <i>Root/Objects/DeviceSet/X20BC008T/Configuration/Control/Reboot</i> oder durch ein Aus- und Einschalten der Spannungsversorgung erfolgen.