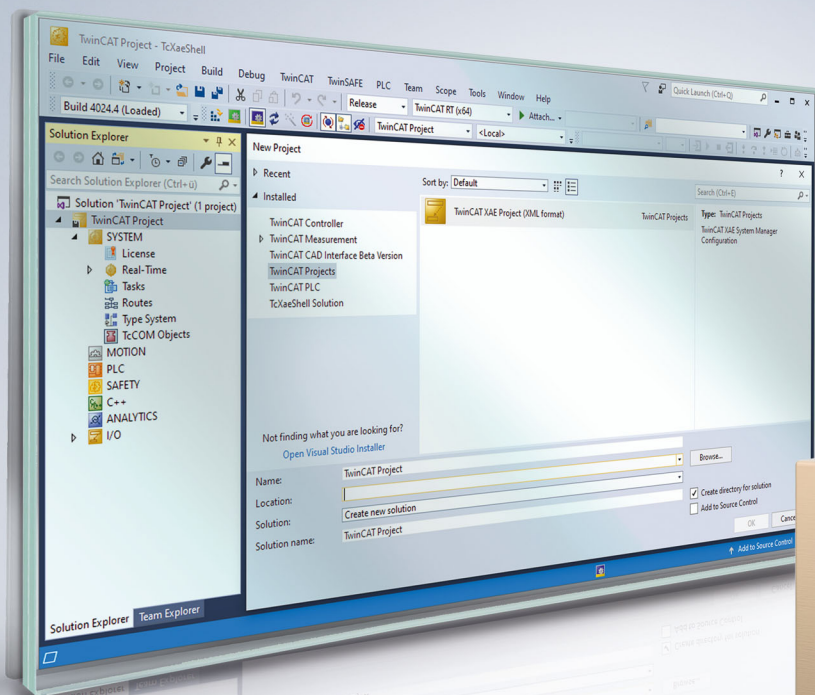


BECKHOFF New Automation Technology

Handbuch | DE

TF8810

TwinCAT 3 | AES70 (OCA) Communication



Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht	8
3	Systemvoraussetzungen	9
4	Installation	10
5	Lizenzierung	14
6	Technische Einführung	17
7	SPS API	18
7.1	Funktionsbausteine	18
7.1.1	FB_OcaDevice	18
7.1.2	FB_OcaRoot	19
7.1.3	Worker-Funktionsbausteine	24
7.2	Datentypen	55
7.2.1	Strukturen, die zur Darstellung der Eigenschaften von OCA-Objekten verwendet werden	55
7.2.2	E_OcaStatus	57
7.2.3	E_OcaMuteState	58
7.2.4	ST_OcaTemperature	58
7.2.5	ST_OcaDeviceInfo	58
7.2.6	E_OcaMessageType	58
8	Beispiele	59
8.1	Beispiel zur Verwendung des Funktionsbausteins FB_OcaDevice	59
8.2	Beispiel zur Verwendung des Funktionsbausteins FB_OcaRoot	59
8.3	Beispiel zur Verwendung des Funktionsbausteins FB_OcaWorker	60
8.4	Beispiel zur Verwendung des Funktionsbausteins FB_OcaGain	61
9	Support und Service	62

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

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

Disclaimer

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

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

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

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

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

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Signalwörter

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

Warnungen vor Personenschäden

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnung vor Umwelt- oder Sachschäden

HINWEIS

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

Information zum Umgang mit dem Produkt



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

1.3 Hinweise zur Informationssicherheit

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

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

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

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

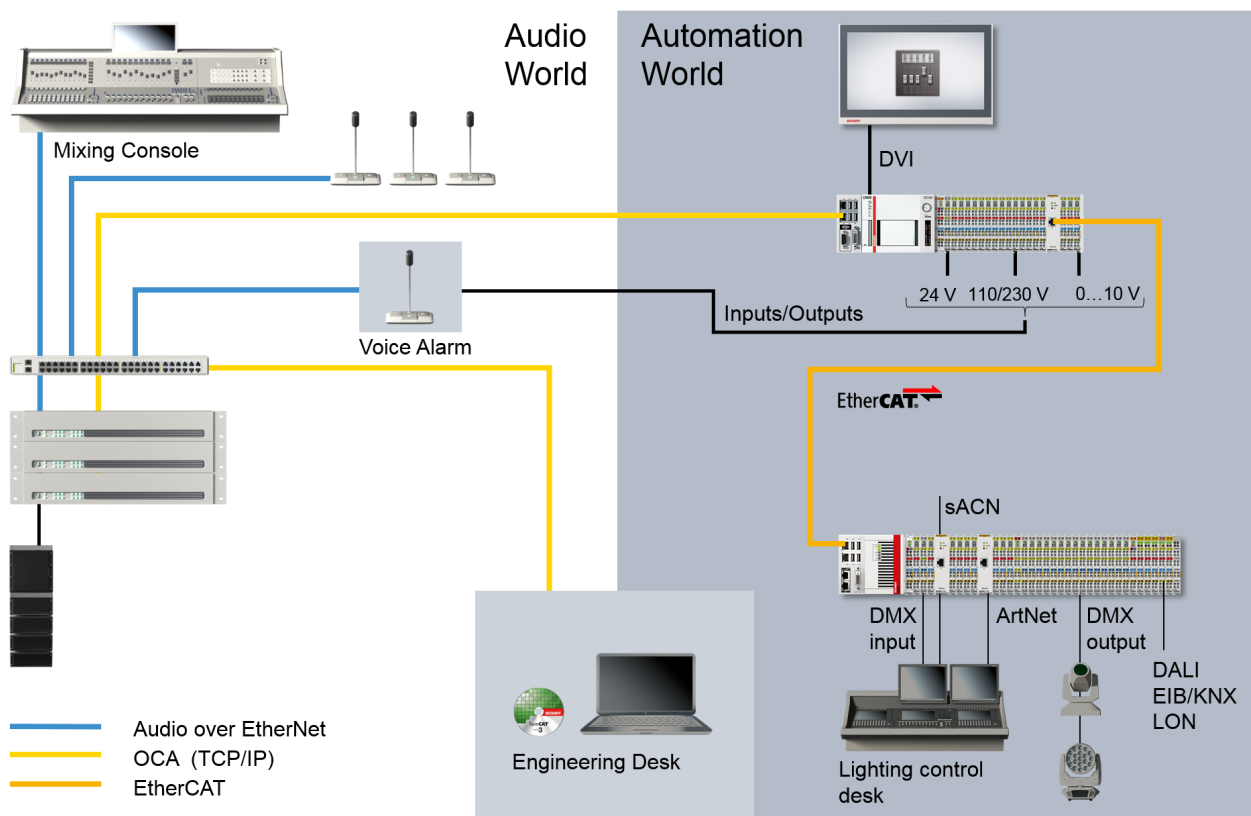
2 Übersicht

Der AES70-Standard wurde von der Audio Engineering Society mit Sitz in New York publiziert und definiert ein skalierbares Steuerungsprotokoll für professionelle Audiogeräte.

Es beschreibt die Überwachung und Steuerung von Geräten, nicht jedoch den Transfer von Mediendaten. Mit den Funktionsbausteinen der SPS-Bibliothek TwinCAT 3 AES70 (OCA) Communication kann ein Datenaustausch zwischen einer TwinCAT SPS und einem Gerät, das den AES70-(OCA)-Standard unterstützt, hergestellt werden. Der Anwender kann Eigenschaften verschiedener Objekte in einem Gerät (Oca Device) abfragen bzw. setzen. Hierzu stehen verschiedene Funktionsbausteine wie FB_OcaGain, FB_OcaMute oder FB_OcaSwitch zur Verfügung.

Somit kann der Anwender OCA-fähige Audioanlagen in die PC-basierte Steuerungstechnik Plattform von Beckhoff einbinden und aus dem breiten Spektrum an Bedien-panels und unterschiedlichen I/Os wählen.

Weitere Informationen finden Sie auf der Beckhoff Homepage im Bereich AV- und Medientechnik > Entertainment-Technologie.



Für die OCA-fähigen Verstärker von d&b audiotechnik steht auf der Homepage von d&b audiotechnik ein Beispiel, bestehend aus einem übergeordneten Funktionsbaustein mit einigen Grundfunktionen wie Gain, Mute oder Presets, zum Download bereit: www.dbaudio.com > Systeme > Netzwerke und Integration > Integration

3 Systemvoraussetzungen

Technische Daten	Voraussetzung
Betriebssystem	Windows 7/10, Windows Embedded Standard 7, Windows CE 7
Zielplattform	PC-Architektur (x86, x64 und ARM)
Minimale TwinCAT-Version	TwinCAT 3.1 Build 4022.2 und höher
Erforderliches TwinCAT-Setup-Level	TwinCAT 3 XAE, XAR
Erforderliche TwinCAT-Lizenz	TF8810 AES70 (OCA) Communication

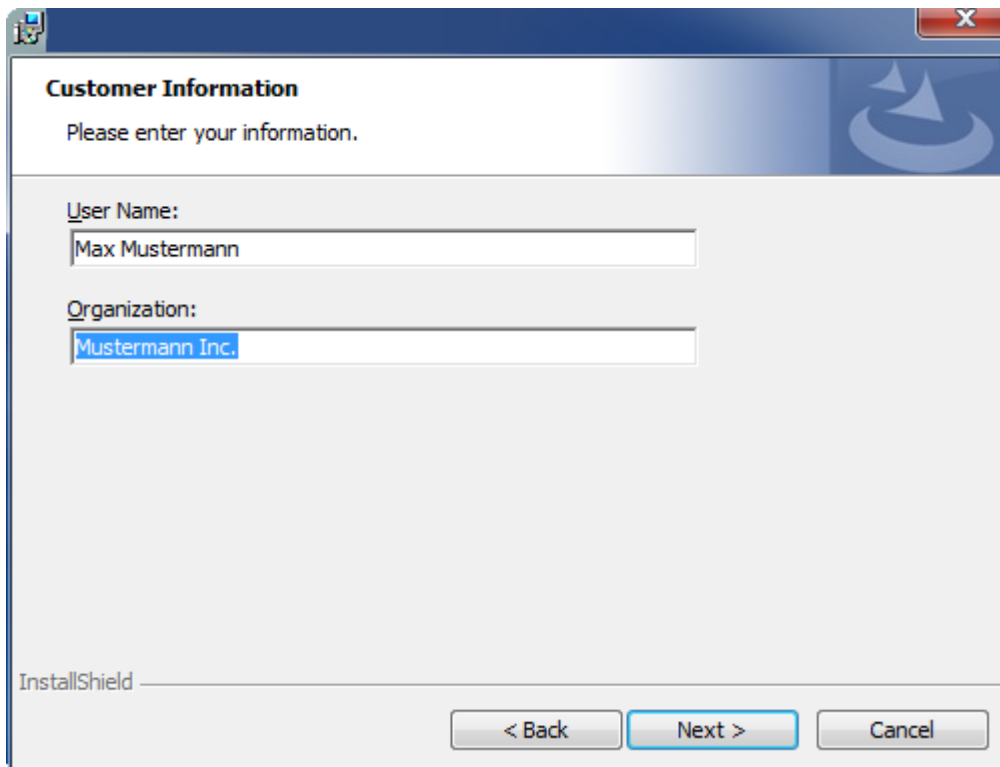
4 Installation

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

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



3. Geben Sie Ihre Benutzerdaten ein.



Customer Information
Please enter your information.

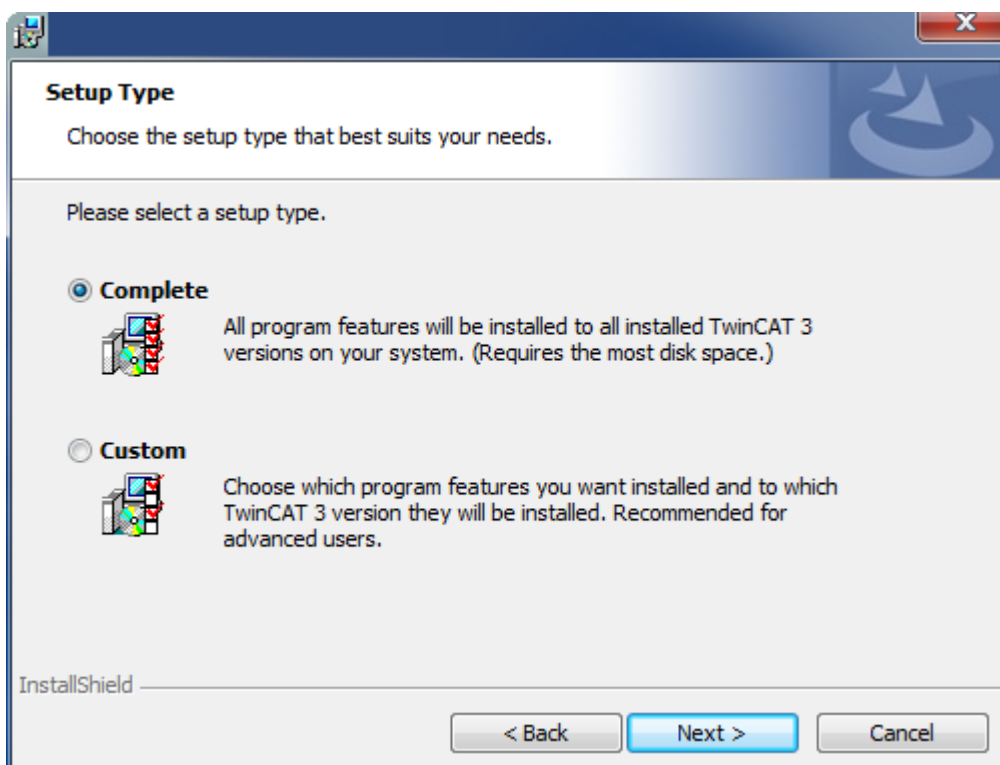
User Name:
Max Mustermann

Organization:
Mustermann Inc.

InstallShield


< Back Next > Cancel


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



Setup Type
Choose the setup type that best suits your needs.

Please select a setup type.

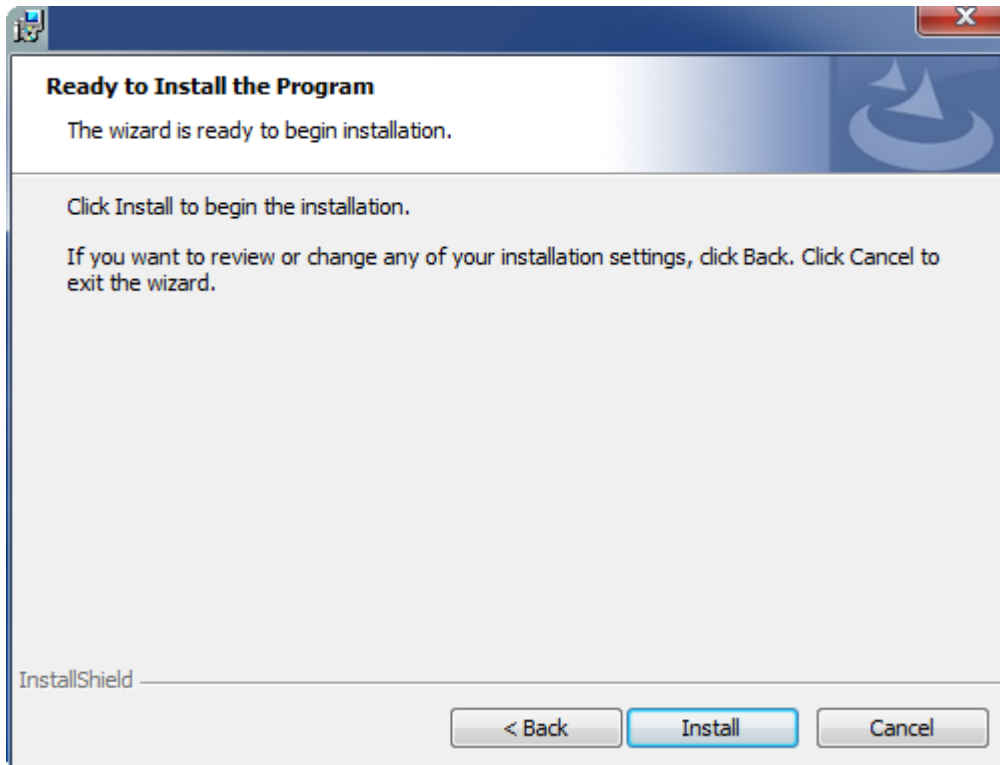
Complete
 All program features will be installed to all installed TwinCAT 3 versions on your system. (Requires the most disk space.)

Custom
 Choose which program features you want installed and to which TwinCAT 3 version they will be installed. Recommended for advanced users.

InstallShield

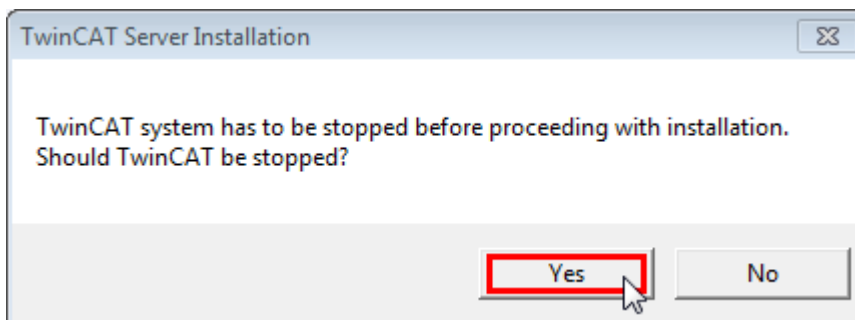
< Back Next > Cancel

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

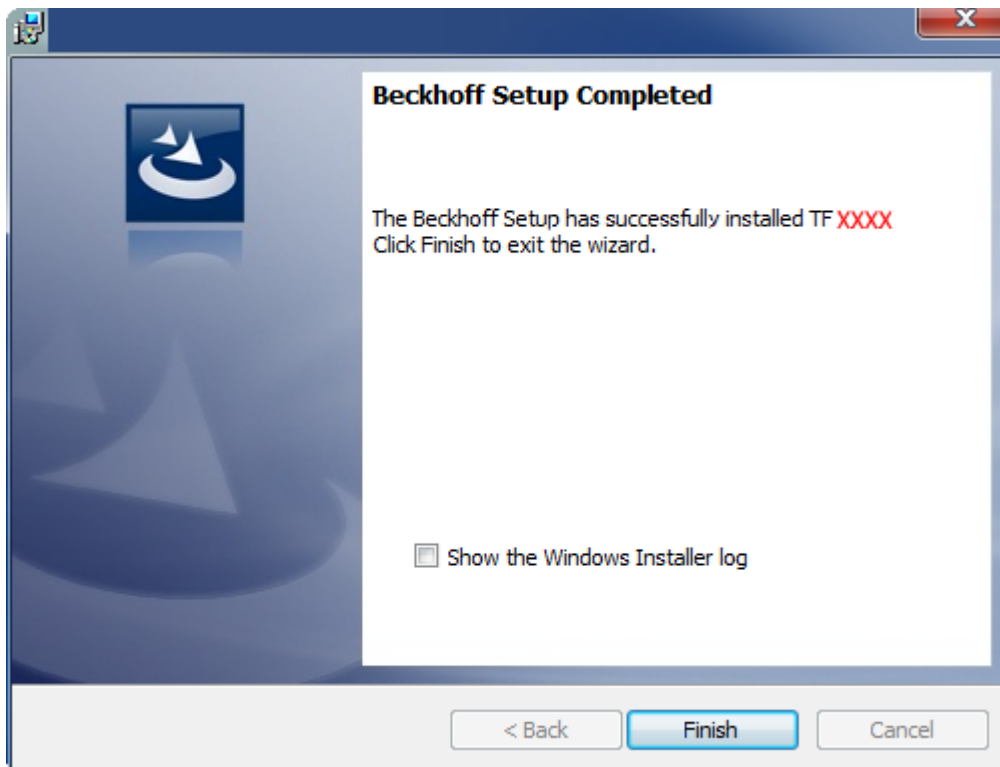


⇒ Ein Dialog weist Sie darauf hin, dass das TwinCAT-System für die weitere Installation gestoppt werden muss.

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



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



⇒ Die TwinCAT 3 Function wurde erfolgreich installiert und kann lizenziert werden (siehe [Lizenzierung \[► 14\]](#)).

5 Lizenzierung

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

Lizenzierung der Vollversion einer TwinCAT 3 Function

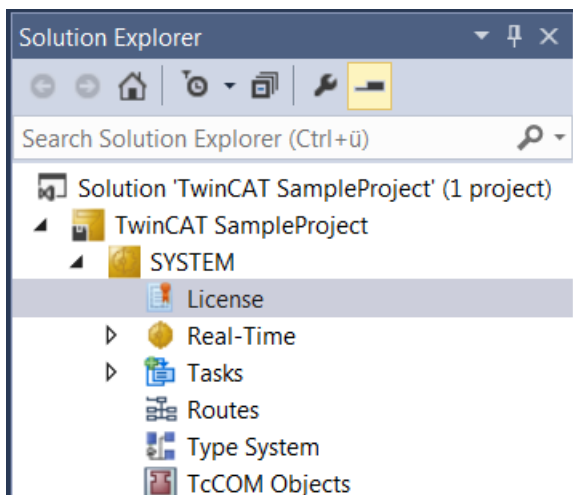
Die Beschreibung der Lizenzierung einer Vollversion finden Sie im Beckhoff Information System in der Dokumentation „[TwinCAT 3 Lizenzierung](#)“.

Lizenzierung der 7-Tage-Testversion einer TwinCAT 3 Function



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

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



⇒ Der TwinCAT-3-Lizenzmanager öffnet sich.

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

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

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

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

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

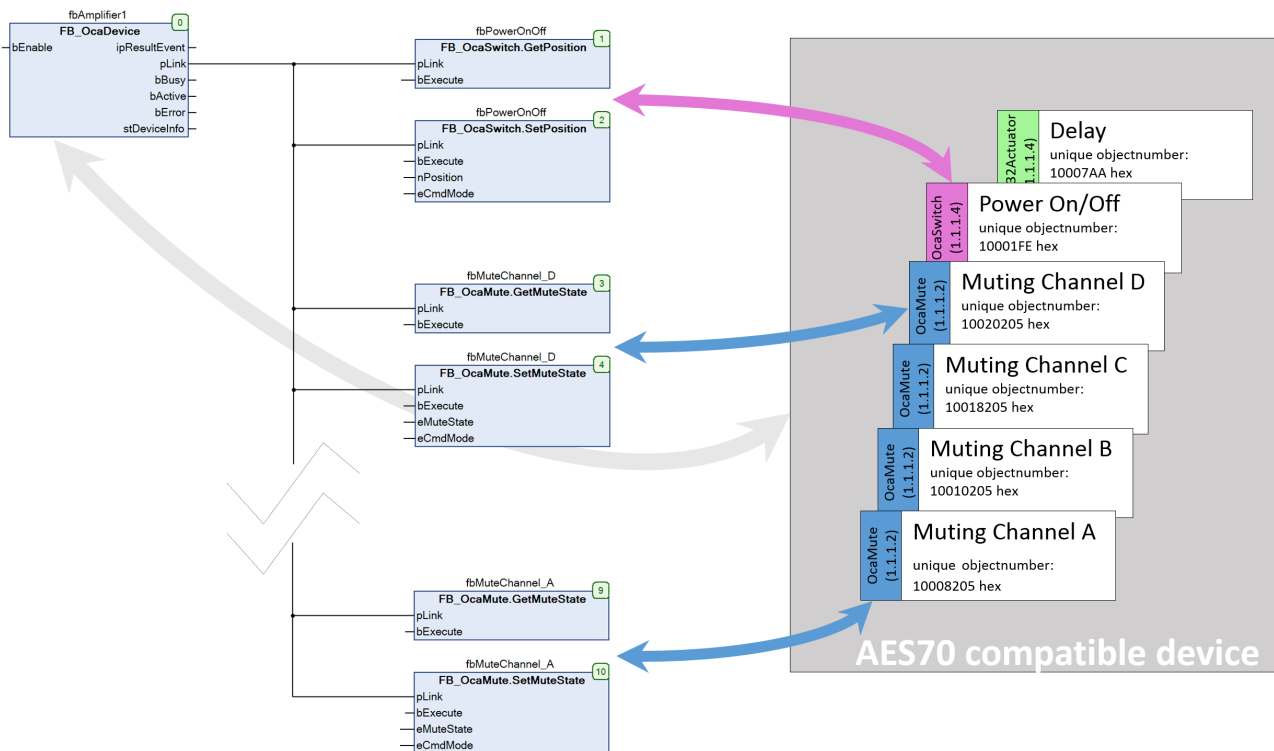
⇒ In der tabellarischen Übersicht der Lizenzen gibt der Lizenzstatus nun das Ablaufdatum der Lizenz an.

10. Starten Sie das TwinCAT-System neu.

⇒ Die 7-Tage-Testversion ist freigeschaltet.

6 Technische Einführung

Die AES70-Spezifikation beschreibt Objekte, die Funktionen und Zustände von Geräten darstellen. Die Handhabung dieser Objekte mit TwinCAT erfolgt über Methoden von Funktionsbausteinen. Jedes Objekt weist eine eindeutige, vom Gerätehersteller definierte, Objektzahl auf.

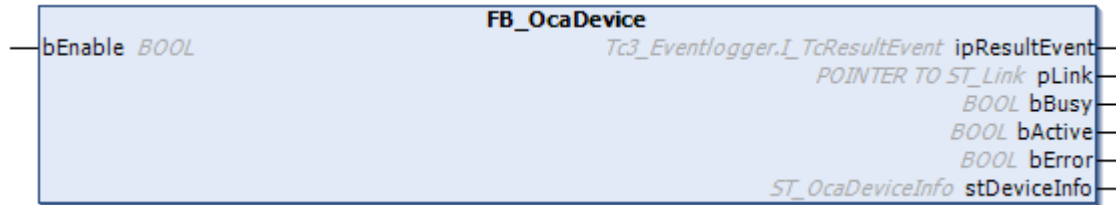


Zu jedem AES70-kompatiblen Gerät muss je eine Instanz des Funktionsbausteins **FB_OcaDevice** [► 18] zyklisch aufgerufen werden. Der Funktionsbaustein stellt die Verbindung per TCP/IP her. Die Parametrierung des Funktionsbausteins (IP-Adresse, Port, ...) wird in dem Beispiel zur Verwendung des Funktionsbausteins FB_OcaDevice [► 59] erläutert. Nach dem Aufruf der Funktionsbausteininstanz müssen die Methoden jener Funktionsbausteine aufgerufen werden, die Objekte im AES70-kompatiblen Gerät darstellen, zum Beispiel sogenannte Worker-Objekte [► 24] wie *OcaMute*, *OcaSwitch*, etc.

7 SPS API

7.1 Funktionsbausteine

7.1.1 FB_OcaDevice



Der Funktionsbaustein FB_OcaDevice stellt die Verbindung per Tcplp zu Geräten her, die den AES70 Standard unterstützen.

● Zyklisch einmaliger Aufruf

i Die Instanz des Funktionsbausteins FB_OcaDevice muss zyklisch einmalig am Anfang des SPS-Programmes aufgerufen werden.

Syntax

```
VAR_INPUT
  bEnable      : BOOL;
END_VAR
VAR_OUTPUT
  ipTcResultEvent : Tc3_EventLogger.I_TcResultEvent;
  pLink          : POINTER TO ST_Link;
  bBusy         : BOOL;
  bActive       : BOOL;
  bError        : BOOL;
  stDeviceInfo   : ST_OcaDeviceInfo;
END_VAR
```

🔌 Eingänge

Name	Typ	Beschreibung
bEnable	BOOL	Aktiviert/deaktiviert die Ausführung des Bausteins und startet den Keep-Alive-Mechanismus. Eine steigende Flanke an diesem Eingang löscht eventuell anstehende Fehler (signalisiert durch bError = TRUE). Das Property sDeviceName sollte vor der Aktivierung festgelegt werden. Ebenso sollten Parameter wie IP-Adresse, Port, etc. in der globalen Variablenliste „GVL_AES70“ bereits definiert sein.

 **Ausgänge**

Name	Typ	Beschreibung
ipTcResultEvent	Tc3_EventLogger.I_TcResultEvent	Ergebnis-Interface, welches den Rückgabewert detailliert angibt
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bBusy	BOOL	TRUE, solange der Funktionsbaustein mit bEnable = TRUE aufgerufen wird.
bActive	BOOL	Zeigt an, dass das Device betriebsbereit ist. Dieser Ausgang wird im Allgemeinen zur Aktivierung von OCA-Objekten verwendet.
bError	BOOL	TRUE, wenn ein Fehler auftritt.
stDeviceInfo	ST_OcaDeviceInfo [► 58]	Diese Struktur bietet Informationen wie zum Beispiel den Zeitpunkt der letzten gesendeten bzw. empfangenen Nachricht sowie die Anzahl der Nachrichten.

 **Eigenschaften**

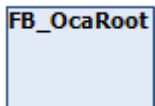
Name	Typ	Zugriff	Definitionsort	Beschreibung
AmsNetId	T_AmsNetID	Set	Lokal	AMS-Netzwerkennung des Gerätes Für den lokalen Rechner (default) kann auch ein Leerstring angegeben werden.
bLocalOcaDevice	BOOL		Lokal	Zurzeit unbenutzt (vorgesehen für zukünftige Erweiterungen)
sDeviceName	STRING	Set		Name des zu verwendenden OCA Devices Parameter wie IP-Adresse und Port werden in dem Array aOcaDevices, welches in der globalen Variablenliste GVL_AES70 zu finden ist, definiert.
tAdsTimeout	TIME	Set		Maximale Zeit, die bei der Ausführung des Funktionsbausteins nicht überschritten werden darf Dieser Eingang ist intern mit <u>DEFAULT ADS TIMEOUT</u> vorbelegt und muss nicht explizit belegt werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

Siehe auch: [Beispiel zur Verwendung des Funktionsbausteins FB_OcaDevice](#) [► 59]

7.1.2 FB_OcaRoot



Der Funktionsbaustein FB_OcaRoot bietet OCA-Basisfunktionalitäten und ist der Funktionsbaustein, von dem alle weiteren Funktionsbausteine erben.



Kein expliziter Aufruf

Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Methoden

Name	Definitionsort	Beschreibung
GetClassIdentification [▶ 21]	Lokal	Mit dieser Methode kann die ClassId und die ClassVersion des OCA-Objektes abgefragt werden. Wenn die Abfrage erfolgreich war, wird das Resultat in den Eigenschaften stClassID und stClassVersion ausgegeben.
GetLockable [▶ 21]	Lokal	Mit dieser Methode wird abgefragt, ob das OCA-Objekt verriegelt werden kann. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stLockable ausgegeben.
LockUnlock [▶ 22]	Lokal	Methode zum Ent- oder Verriegeln eines OCA-Objektes Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stObjectLocked ausgegeben.
GetRole [▶ 22]	Lokal	OCA-Objekte können eine Text-Bezeichnung haben, um deren Bedeutung einfacher zu erkennen. Wenn der Aufruf der Methode erfolgreich war, wird das Resultat in der Eigenschaft stRole ausgegeben.
Subscription [▶ 23]	Lokal	Mit dieser Methode ist es möglich, OCA Objekte aufzufordern, Benachrichtigungen (Notifications) bei Wertänderungen zu generieren oder das Versenden einzustellen. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stSubscribed ausgegeben.

Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
sClassId	STRING	Get	Lokal	Eindeutige Klassenbezeichnung
stClassId	ST_ClassIdProperty [▶ 55]	Get	Lokal	Eindeutige Klassenbezeichnung
stClassVersion	ST_UINT16Property [▶ 55]	Get	Lokal	OCA ClassVersion
stLockable	ST_BooleanProperty [▶ 55]	Get	Lokal	Zeigt an, ob das Objekt gesperrt werden kann
stSubscribed	ST_BooleanProperty [▶ 55]	Get	Lokal	Zeigt an, ob das Objekt aufgefordert wurde bei Wertänderungen Benachrichtigungen (Notifications) zu generieren
stRole	ST_StringProperty [▶ 55]	Get	Lokal	Aufgabe des Objektes in dem Gerät (z.B. Config_InputEnable)
nONo	UDINT	Get	Lokal	Eindeutige Nummer, mit der das instanziierte Objekt eindeutig definiert ist
sOcaObjectDescription	STRING(32)	Set	Lokal	Willkürliche Objektbezeichnung. Anhand dieser werden dem Funktionsbaustein weitere Objekteigenschaften zugeordnet, die in dem Array aOcaDevices (zu finden in der globalen Variablenliste GVL_AES70) definiert wurden.
stObjectLocked	ST_BooleanProperty [▶ 55]	Get	Lokal	Zeigt an, ob das Objekt gesperrt ist.

i AES70-Standard

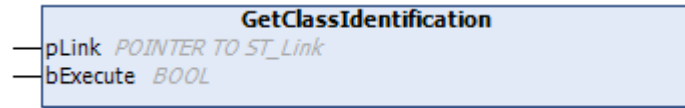
Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

Siehe auch: [Beispiel zur Verwendung des Funktionsbausteins FB_OcaRoot](#) [► 59]

7.1.2.1 GetClassIdentification



Mit der Methode GetClassIdentification kann die ClassId und die ClassVersion des OCA-Objektes abgefragt werden. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stClassID und stClassVersion ausgegeben.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

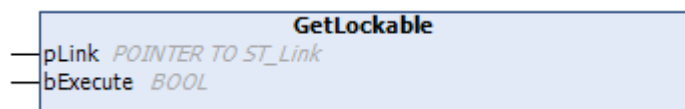
```

VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
END_VAR
    
```

+ Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

7.1.2.2 GetLockable



Mit der Methode GetLockable wird abgefragt, ob das OCA-Objekt verriegelt werden kann. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stLockable ausgegeben.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
END_VAR
    
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

7.1.2.3 LockUnlock

LockUnlock	
— pLink	POINTER TO ST_Link
— bExecute	BOOL
— bLock	BOOL
— eCmdMode	E_OcaMessageTypeCmd

Mit der Methode LockUnlock kann ein OCA-Objekt ent- oder verriegelt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stObjectLocked ausgegeben.

Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  bLock      : BOOL; //If the Method is executed this Boolean Input decides whether the object
should be locked (TRUE) or unlocked (FALSE)
  eCmdMode   : E_OcaMessageTypeCmdbLock;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
bLock	BOOL	Variable, die bestimmt, ob verriegelt (TRUE) bzw. entriegelt (FALSE) werden soll
eCmdMode	E_OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

7.1.2.4 GetRole

GetRole	
— pLink	POINTER TO ST_Link
— bExecute	BOOL

OCA-Objekte können eine Textbezeichnung haben, um deren Bedeutung einfach zu erkennen. Wenn der Aufruf der Methode erfolgreich war, wird das Resultat in der Eigenschaft stRole ausgegeben.

Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

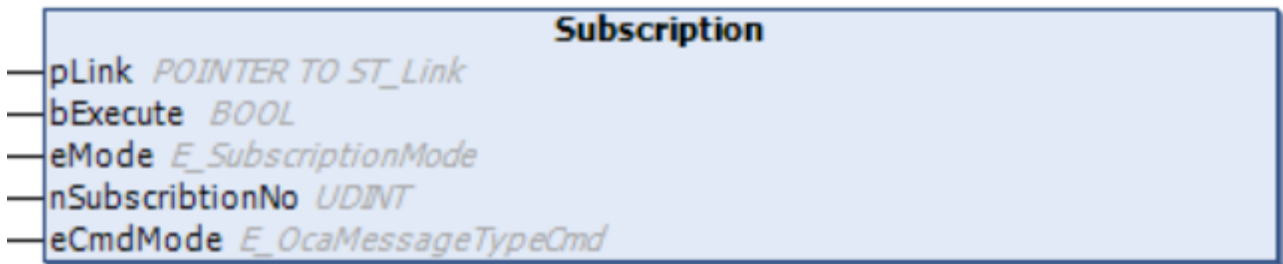
Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

7.1.2.5 Subscription



Mit der Methode Subscription ist es möglich, OCA-Objekte aufzufordern, Benachrichtigungen (Notifications) bei Wertänderungen zu generieren oder das Versenden von Benachrichtigungen einzustellen. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stSubscribed ausgegeben.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  eMode      : E_SubscriptionMode;
  nSubscriptionNo : UDINT;
  eCmdMode   : E_OcaMessageTypeCmd;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
eMode	E_SubscriptionMode	Abhängig davon, ob eine Aufforderung zur Benachrichtigung erstellt oder gelöscht werden soll kann der Eingang der Methode auf E_SubscriptionMode.ADD_Subscription oder E_SubscriptionMode.DELETE_Subscription gesetzt werden.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription
eCmdMode	E_OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

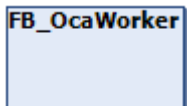
7.1.3 Worker-Funktionsbausteine

Worker-Funktionsbausteine werden zum Steuern von OCA-Worker-Objekten verwendet.

● AES70-Standard

i Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

7.1.3.1 FB_OcaWorker



Der Funktionsbaustein FB_OcaWorker erweitert den Funktionsbaustein FB_OcaRoot um Eigenschaften und Methoden zur Handhabung der OCA-Objekte.

● Kein expliziter Aufruf

i Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie

FB_OcaRoot

 FB_OcaWorker

 **Methoden**

Name	Definitionsort	Beschreibung
GetEnabled / SetEnabled [▶ 26]	Lokal	Methode zur Abfrage bzw. zum Setzen der Enabled-Eigenschaft des OCA-Objekts. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stEnabled ausgegeben.
GetLabel / SetLabel / RNTfLabel [▶ 27]	Lokal	Mit dieser Methode wird abgefragt, ob das OCA-Objekt verriegelt werden kann. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stLockable ausgegeben.
LockUnlock [▶ 22]	Lokal	Methode zum Ent- oder Verriegeln eines OCA-Objektes. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stObjectLocked ausgegeben.
GetRole [▶ 22]	Lokal	OCA-Objekte können eine Text-Bezeichnung haben, um deren Bedeutung einfacher zu erkennen. Wenn der Aufruf der Methode erfolgreich war, wird das Resultat in der Eigenschaft stRole ausgegeben.
Subscription [▶ 23]	Lokal	Mit dieser Methode ist es möglich, OCA-Objekte aufzufordern, Benachrichtigungen (Notifications) bei Wertänderungen zu generieren oder das Versenden einzustellen. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stSubscribed ausgegeben.

 **Eigenschaften**

Name	Typ	Zugriff	Definitionsort	Beschreibung
stEnabled	ST_BooleanProperty [▶ 55]	Get	Lokal	Zeigt an, ob das OCA-Objekt in dem entsprechenden OCA Device „enabled“ (aktiviert) ist.
stLabel	ST_StringProperty [▶ 55]	Get	Lokal	OCA-Objekten können zur besseren Erkennbarkeit einen Bezeichnungstext besitzen, der in dieser Eigenschaft abgelegt ist.
stLatency	ST_FLOAT32Property [▶ 55]	Get	Lokal	Bearbeitungslatenz des OCA-Objektes
stOwner	ST_UINT32Property [▶ 55]	Get	Lokal	OCA-Objektnummer des übergeordneten Objektes

AES70-Standard

i Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

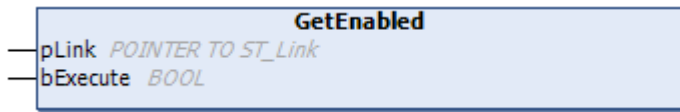
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

Siehe auch: [Beispiel zur Verwendung des Funktionsbausteins FB OcaWorker](#) [[▶ 60](#)]

7.1.3.1.1 GetEnabled / SetEnabled

GetEnabled



Die Methode GetEnabled dient zur Abfrage der Enabled-Eigenschaft des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stEnabled ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

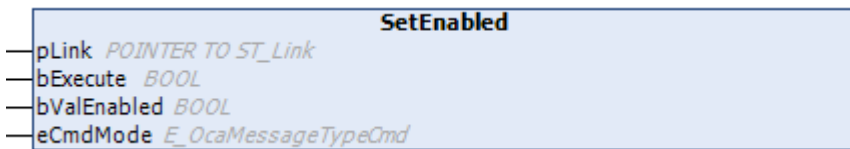
VAR_INPUT
  pLink    : POINTER TO ST_Link;
  bExecute : BOOL;
END_VAR

```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetEnabled



Mit der Methode SetEnabled kann ein OCA-Objekt „enabled“ (aktiviert) oder „disabled“ (deaktiviert) werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Ergebnis in der Eigenschaft stEnabled ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  bValEnabled : BOOL;
  eCmdMode   : E_OcaMessageTypeCmd;
END_VAR

```

 **Eingänge**

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
bValEnabled	BOOL	Abhängig davon, ob das OCA-Objekt aktiviert oder deaktiviert werden soll, ist dieser Eingang mit TRUE oder FALSE zu belegen.
eCmdMode	E_OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

7.1.3.1.2 GetLabel / SetLabel / RntfLabel

GetLabel



OCA-Objekte können zur besseren Erkennbarkeit mit einer Beschriftung versehen werden. Die Methode GetLabel dient zur Abfrage dieser Eigenschaft des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stLabel ausgegeben.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
END_VAR
    
```

 **Eingänge**

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetLabel



Mit der Methode SetLabel kann die Beschriftung eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stLabel ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  sLabel     : STRING;
  eCmdMode   : E_OcaMessageTypeCmd;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
sLabel	STRING	Beschriftung, die das OCA Objekt erhalten soll
eCmdMode	E_OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

RNtfLabel

RNtfLabel	
pLink	POINTER TO ST_Link
bEnable	BOOL
nSubscriptionNo	UDINT

Wenn das OCA-Objekt aufgefordert wird, Änderungen bekanntzugeben (mittels der Methode [Subscription](#) [► 23]), werden mit der Methode RNtfLabel eingehende „Notifications“ (Benachrichtigungen) gelesen.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bEnable    : BOOL;
  nSubscriptionNo : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

7.1.3.1.3 GetLatency / SetLatency / RntfLatency

GetLatency



Die Methode GetLatency dient zur Abfrage der Bearbeitungslatenz des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stLatency ausgegeben.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

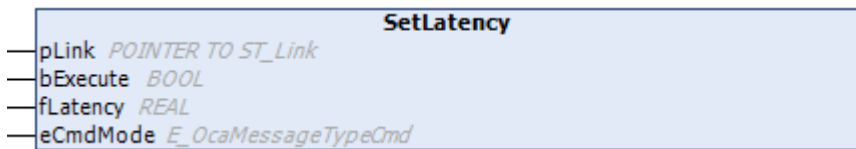
```

VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
END_VAR
    
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetLatency



Mit der Methode SetLatency kann die Bearbeitungslatenz eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stLatency ausgegeben.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  fLatency   : REAL;
  eCmdMode   : E_OcaMessageTypeCmd;
END_VAR
    
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
fLatency	REAL	Wert, den diese Eigenschaft des OCA-Objekt erhalten soll
eCmdMode	E_OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

RNtfLatency



Wenn das OCA-Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode [Subscription](#) [[▶ 23](#)]), werden mit der Methode **RNtfLatency** eingehende „Notifications“ (Benachrichtigungen) gelesen.

Zyklischer Methodenaufruf



Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

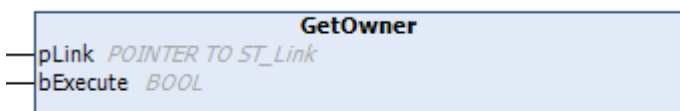
Syntax

```
VAR_INPUT
  pLink          : POINTER TO ST_Link;
  bEnable       : BOOL;
  nSubscriptionNo : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

7.1.3.1.4 GetOwner



Die Methode **GetOwner** dient zur Abfrage des übergeordneten Elementes. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft `stOwner` ausgegeben.

Zyklischer Methodenaufruf



Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

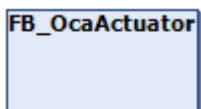
7.1.3.2 Actuator-Funktionsbausteine

Actuator-Objekte sind Worker-Objekte, deren Funktion die Steuerung von elektronischen Funktionen ist.

● AES70-Standard

i Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

7.1.3.2.1 FB_OcaActuator

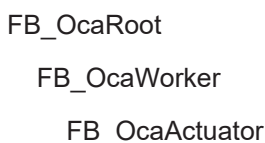


Der Funktionsbaustein FB_OcaActuator erweitert den Funktionsbaustein FB_OcaWorker und ist der Funktionsbaustein, von dem alle Funktionsbausteine erben, die der Kategorie Actuator zugeordnet sind. FB_OcaActuator besitzt weder eigene Eigenschaften noch Methoden.

● Kein expliziter Aufruf

i Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

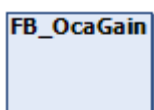
Vererbungshierarchie



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.2.2 FB_OcaGain



Der Funktionsbaustein FB_OcaGain bietet Eigenschaften und Methoden zur Handhabung von OCA-Objekten zur Verstärkungseinstellung.

Kein expliziter Aufruf

Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie

FB_OcaRoot

 FB_OcaWorker

 FB_OcaActuator

 FB_OcaGain

Methoden

Name	Definitionsort	Beschreibung
GetGain [▶ 33]	Lokal	Dient zur Abfrage der Eigenschaften Gain, GainMAX und GainMIN des OCA-Objekts. Wenn die Abfrage erfolgreich war, wird das Resultat in den Eigenschaften stGain, stGainMAX und stGainMIN ausgegeben.
SetGain [▶ 33]	Lokal	Mit dieser Methode kann die Verstärkung in dB eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stGain ausgegeben.
RNtfGain [▶ 33]	Lokal	Wenn das OCA Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode Subscription [▶ 23]), werden mit dieser Methode eingehende „Notifications“ (Benachrichtigungen) gelesen.

Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
stGain	ST_FLOAT32Property [▶ 55]	Get	Lokal	Verstärkung in dB
stGainMAX	ST_FLOAT32Property [▶ 55]	Get	Lokal	Maximalwert der Verstärkung in dB
stGainMIN	ST_FLOAT32Property [▶ 55]	Get	Lokal	Minimalwert der Verstärkung in dB

AES70-Standard

Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.oaalliance.com.

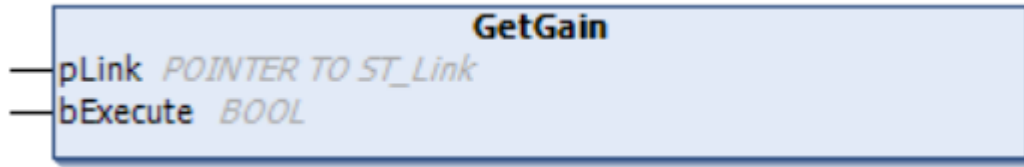
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

Siehe auch: [Beispiel zur Verwendung des Funktionsbausteins FB_OcaGain](#) [[▶ 61](#)]

7.1.3.2.2.1 GetGain / SetGain / RntfGain

GetGain



Die Methode GetGain dient zur Abfrage der Eigenschaften Gain, GainMAX und GainMIN des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in den Eigenschaften stGain, stGainMAX und stGainMIN ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

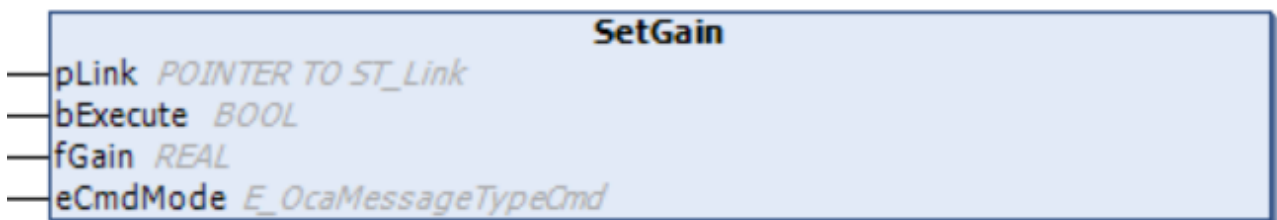
Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
END_VAR
```

 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetGain



Mit der Methode SetGain kann die Verstärkung in dB eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stGain ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

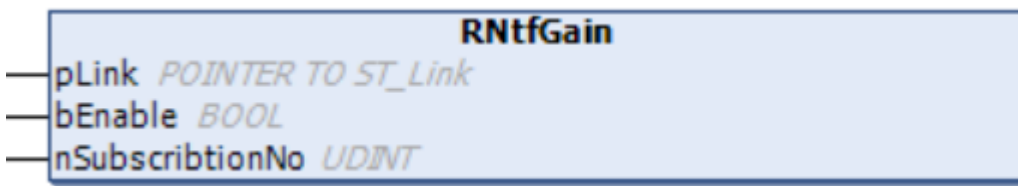
Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  fGain      : REAL;
  eCmdMode   : E_OcaMessageTypeCmd;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
fGain	REAL	Wert der Verstärkung in dB, den das OCA-Objekt erhalten soll.
eCmdMode	E_OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

RNtfGain



Wenn das OCA-Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode [Subscription](#) [► 23]), werden mit der Methode RNtfGain eingehende „Notifications“ (Benachrichtigungen) gelesen.

Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

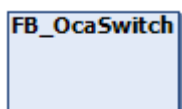
Syntax

```
VAR_INPUT
  pLink          : POINTER TO ST_Link;
  bEnable        : BOOL;
  nSubscriptionNo : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

7.1.3.2.3 FB_OcaSwitch



Der Funktionsbaustein FB_OcaSwitch erweitert den Funktionsbaustein FB_OcaActuator um Eigenschaften und Methoden und zur Handhabung von OCA-Switch-Objekten. Diese Objekte können mehrere Schaltstellungen besitzen. Einzelne Positionen können aktiviert bzw. deaktiviert (enable/disable) und mit Namen versehen werden.

Kein expliziter Aufruf

i Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie

- FB_OcaRoot
 - FB_OcaWorker
 - FB_OcaActuator
 - FB_OcaSwitch

 **Methoden**

Name	Definitionsort	Beschreibung
GetPosition [▶ 36]	Lokal	Diese Methode dient zur Abfrage der Eigenschaften Position, PositionMAX und PositionMIN des OCA-Objektse.
SetPosition [▶ 36]	Lokal	Mit dieser Methode kann die Schalterstellung eines OCA-Objektes gesetzt werden.
RNtfPosition [▶ 36]	Lokal	Diese Methode liest eingehende „Notifications“ (Benachrichtigungen)
GetPositionEnabled [▶ 38]	Lokal	Methode zur Abfrage eine Schalterposition
SetPositionEnabled [▶ 38]	Lokal	Methode zum Aktivieren oder Deaktivieren einer Schalterposition
GetPositionName	Lokal	Methode zur Abfrage des Namens einer Schalterposition
SetPositionName	Lokal	Mit dieser Methode kann einer Schalterposition ein Name zugeordnet werden.

 **Eigenschaften**

Name	Typ	Zugriff	Definitionsort	Beschreibung
stPosition	ST_UINT16Property [▶ 55]	Get	Lokal	Zeigt die Schaltstellung des OCA-Objekts an.
stPosition MAX	ST_UINT16Property [▶ 55]	Get	Lokal	Zeigt die niedrigste Schaltstellung des OCA-Objekts an.
stPosition MIN	ST_UINT16Property [▶ 55]	Get	Lokal	Zeigt die höchste Schaltstellung des OCA-Objekts an.
stPosition Enabled	ST_PositionEnabledProperty [▶ 55]	Get	Lokal	Zeigt an, ob eine bestimmte Schalterstellung aktiviert oder deaktiviert ist.
stPosition Name	ST_PositionNameProperty [▶ 55]	Get	Lokal	Zeigt den Positionsnamen einer bestimmte Schalterstellung .

AES70-Standard

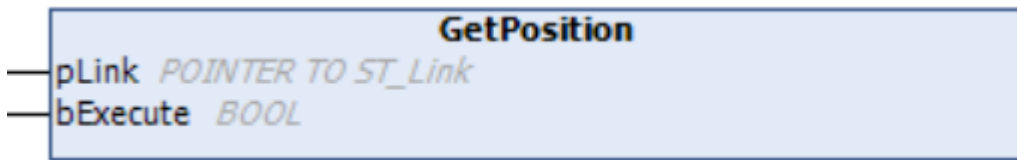
i Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.2.3.1 GetPosition / SetPosition / RNtfPosition

GetPosition



Die Methode `GetPosition` dient zur Abfrage der Eigenschaften `Position`, `PositionMAX` und `PositionMIN` des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in den Eigenschaften `stPosition`, `stPositionMAX` und `stPositionMIN` ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink    : POINTER TO ST_Link; //Pointer to address of the structure which links the OCA objects
to the OCA device
  bExecute : BOOL; //The Method is triggered by a rising edge at this input.
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetPosition



Mit der Methode `SetPosition` kann die Schalterstellung eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft `stPosition` ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink    : POINTER TO ST_Link; //Pointer to address of the structure which links the OCA objects
to the OCA device
```

```
bExecute : BOOL; //The Method is triggered by a rising edge at this input.
nPosition : UINT;
eCmdMode : E_OcaMessageTypeCmdbLock;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nPosition	UINT	Schalterstellung, die das OCA-Objekt erhalten soll.
eCmdMode	E_OcaMessageTypeCmdbLock	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

RNtfPosition



Wenn das OCA-Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode [Subscription](#) [► 23]), werden mit der Methode RNtfPosition eingehende „Notifications“ (Benachrichtigungen) gelesen.

Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

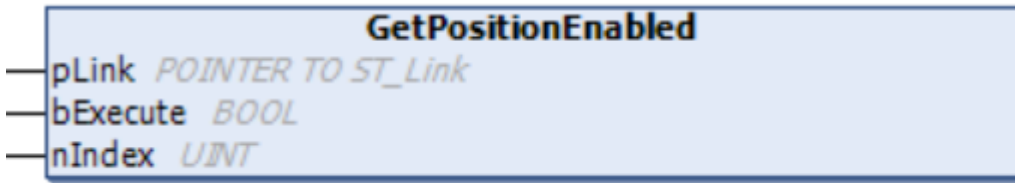
```
VAR_INPUT
  pLink : POINTER TO ST_Link;
  bEnable : BOOL;
  nSubscriptionNo : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

7.1.3.2.3.2 GetPositionEnabled / SetPositionEnabled

GetPositionEnabled



Die Methode GetPositionEnabled dient zur Abfrage, ob die jeweilige, durch nIndex bestimmte Schalterposition aktiviert bzw. deaktiviert ist. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stPositionEnabled ausgegeben.

● Zyklischer Methodenaufwurf



Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

VAR_INPUT
  pLink      : POINTER TO ST_Link; //Pointer to address of the structure which links the OCA objects
to the OCA device
  bExecute   : BOOL; //The Method is triggered by a rising edge at this input.
  nIndex     : UINT; //The Index of the queried Position
END_VAR

```

📌 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nIndex	UINT	Index der abzufragenden Schalterposition

SetPositionEnabled



Mit der Methode SetPositionEnabled kann eine bestimmte, durch nIndex festgelegte Schalterposition aktiviert bzw. deaktiviert werden. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stPositionEnabled ausgegeben.

● Zyklischer Methodenaufwurf



Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

VAR_INPUT
  pLink      : POINTER TO ST_Link; //Pointer to address of the structure which links the OCA
objects to the OCA device
  bExecute   : BOOL; //The Method is triggered by a rising edge at this input.

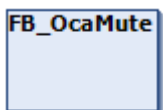
```

```
bValEnabled : BOOL; //Set this Input to TRUE to enable the Position specified by nIndex or FALSE
to disable this Position
nIndex      : UINT; //The Index of the Position which should be modified
eCmdMode    : E_OcaMessageTypeCmdbLock;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
bValEnabled	BOOL	Wenn dieser Eingang auf TRUE gesetzt wird, wird bei Ausführung dieser Methode die durch nIndex gewählte Schalterstellung aktiviert („enabled“), bei FALSE wird diese deaktiviert („disabled“).
nPosition	UINT	Schalterstellung, die das OCA-Objekt erhalten soll.
eCmdMode	E_OcaMessageTypeCmdbLock	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

7.1.3.2.4 FB_OcaMute

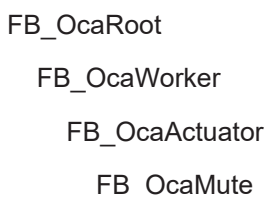


Der Funktionsbaustein FB_OcaMute bietet Eigenschaften und Methoden zur Handhabung von OCA-Objekten zur Stummschaltung.

Kein expliziter Aufruf

i Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie



 **Methoden**

Name	Definitionsort	Beschreibung
GetMuteState [► 40]	Lokal	Diese Methode dient zur Abfrage der Eigenschaften Position, PositionMAX und PositionMIN des OCA-Objektes.
SetMuteState [► 40]	Lokal	Mit dieser Methode kann die Schalterstellung eines OCA-Objektes gesetzt werden.
RNtfMuteState [► 40]	Lokal	Diese Methode liest eingehende „Notifications“ (Benachrichtigungen).

Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
stMuteState	ST_MuteProperty [► 55]	Get	Lokal	Zeigt an, ob das OCA-Objekt stummgeschaltet ist oder nicht.

AES70-Standard

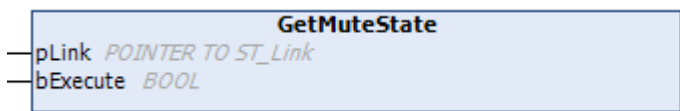
Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.2.4.1 GetMuteState / SetMuteState / RNtfMuteState

GetMuteState



Die Methode GetPosition dient zur Abfrage der Eigenschaft MuteState des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stMuteState ausgegeben.

Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

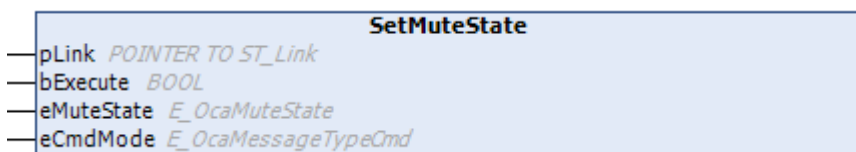
```

VAR_INPUT
    pLink    : POINTER TO ST_Link;
    bExecute : BOOL;
END_VAR
  
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetMuteState



Mit der Methode SetMuteState kann die Stummschaltung eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stMuteState ausgegeben.

Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

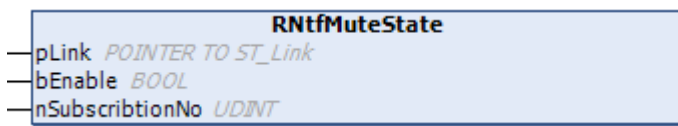
Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  eMuteState : E_OcaMuteState;
  eCmdMode   : E_OcaMessageTypeCmd;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
eMuteState	E_OcaMuteState [► 58]	„Muting“-Einstellung (muted/unmuted), die das OCA-Objekt erhalten soll.
eCmdMode	OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

RNtfMuteState



Wenn das OCA-Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode [Subscription \[► 23\]](#)), werden mit der Methode RNtfMuteState eingehende „Notifications“ (Benachrichtigungen) gelesen.

Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

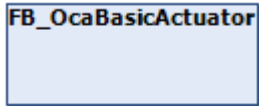
Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bEnable    : BOOL;
  nSubscriptionNo : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

7.1.3.2.5 FB_OcaBasicActuator



Der Funktionsbaustein FB_OcaBasicActuator erweitert den Funktionsbaustein FB_OcaActuator und ist der Funktionsbaustein, von dem alle Funktionsblöcke erben, die der Kategorie BasicActuator zugeordnet sind. FB_OcaBasicActuator besitzt weder eigene Eigenschaften noch Methoden.

Vererbungshierarchie

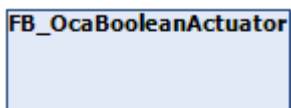
```

FB_OcaRoot
  FB_OcaWorker
    FB_OcaActuator
      FB_OcaBasicActuator
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.2.6 FB_OcaBooleanActuator



Der Funktionsbaustein FB_OcaBooleanActuator erweitert den Funktionsbaustein FB_OcaBasicActuator um Eigenschaften und Methoden zur Handhabung von OcaBooleanActuator-Objekten.

● Kein expliziter Aufruf

i Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie

```

FB_OcaRoot
  FB_OcaWorker
    FB_OcaActuator
      FB_OcaBasicActuator
        FB_OcaBooleanActuator
  
```

☞ Methoden

Name	Definitionsart	Beschreibung
GetSetting [▶ 43]	Lokal	Diese Methode dient zur Abfrage einer booleschen Eigenschaft.
SetSetting [▶ 43]	Lokal	Mit dieser Methode kann der Wert einer booleschen Eigenschaft gesetzt werden.
RNtfSetting [▶ 43]	Lokal	Diese Methode liest eingehende „Notifications“ (Benachrichtigungen)

 **Eigenschaften**

Name	Typ	Zugriff	Definitionsort	Beschreibung
stSetting	ST_BooleanProperty [55]	Get	Lokal	Zeigt die Schaltstellung des OCA-Objektes an.

AES70-Standard

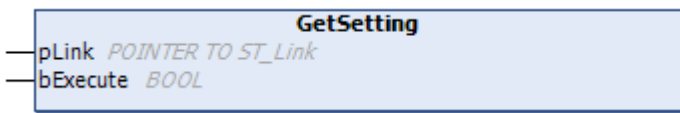
i Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.2.6.1 GetSetting / SetSetting / RNtfSetting

GetSetting



Die Methode GetSetting dient zur Abfrage der booleschen Eigenschaft des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stSetting ausgegeben.

Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

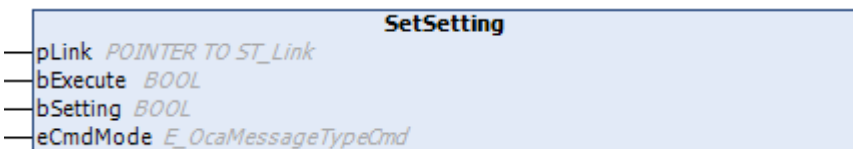
```

VAR_INPUT
    pLink    : POINTER TO ST_Link;
    bExecute : BOOL;
END_VAR
    
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetSetting



Mit der Methode SetSetting kann die Stummschaltung eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stMuteState ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

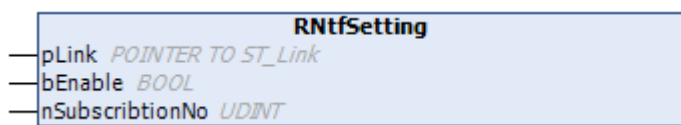
Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  bSetting   : BOOL;
  eCmdMode   : E_OcaMessageTypeCmd;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
bSetting	BOOL	Wert, den die Eigenschaft des OCA-Objektes erhalten soll.
eCmdMode	OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

RNtfMuteState



Wenn das OCA-Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode [Subscription](#) [► 23]), werden mit der Methode RNtfSetting eingehende „Notifications“ (Benachrichtigungen) gelesen.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

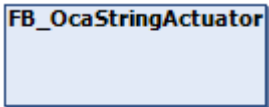
Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bEnable    : BOOL;
  nSubscriptionNo : UDINT;
END_VAR
```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

7.1.3.2.7 FB_OcaStringActuator

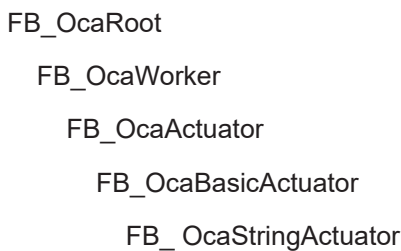


Der Funktionsbaustein FB_OcaStringActuator erweitert den Funktionsbaustein FB_OcaBasicActuator um Eigenschaften und Methoden zur Handhabung von OcaStringActuator-Objekten.

Kein expliziter Aufruf

i Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie



Methoden

Name	Definitionsort	Beschreibung
GetValue [▶ 46]	Lokal	Diese Methode dient zur Abfrage einer vorzeichenbehafteten Eigenschaft (ganzzahliger, vorzeichenbehafteter Datentyp, 32 Bit).
SetValue [▶ 46]	Lokal	Mit dieser Methode kann der Wert der Eigenschaft gesetzt werden.
RNtfValue [▶ 46]	Lokal	Diese Methode liest eingehende „Notifications“ (Benachrichtigungen)
GetMaxLen [▶ 47]	Lokal	Mit dieser Methode kann die maximale Länge der Zeichenkette, die das OCA-Objekt akzeptiert, abgefragt werden.

Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
stSetting	ST_STRINGProperty [▶ 55]	Get	Lokal	Beinhaltet die Zeichenkette
stMaxLen	ST_UINT16Property [▶ 55]	Get	Lokal	Beinhaltet die maximale Länge der Zeichenkette

AES70-Standard

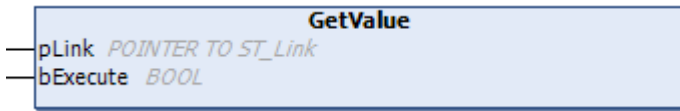
i Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.2.7.1 GetValue / SetValue / RntfValue

GetValue



Die Methode GetValue dient zur Abfrage der Eigenschaft des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stSetting ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

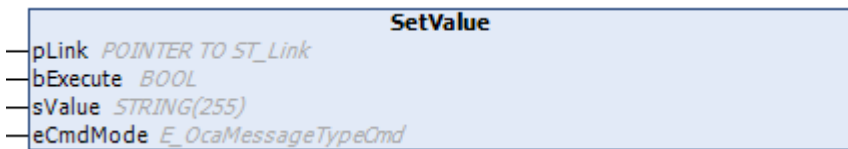
VAR_INPUT
  pLink   : POINTER TO ST_Link;
  bExecute : BOOL;
END_VAR

```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetValue



Mit der Methode SetValue kann die Eigenschaft eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stSetting ausgegeben.

i Zyklischer Methodenaufruf

Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```

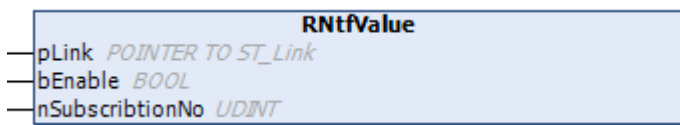
VAR_INPUT
  pLink   : POINTER TO ST_Link;
  bExecute : BOOL;
  sValue  : STRING;
  eCmdMode : E_OcaMessageTypeCmd;
END_VAR

```

 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSetting	UDINT	Wert, den die Eigenschaft des OCA-Objektes erhalten soll.
eCmdMode	OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

RNtfValue



Wenn das OCA-Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode Subscription [► 23]), werden mit der Methode RNtfValue eingehende „Notifications“ (Benachrichtigungen) gelesen.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

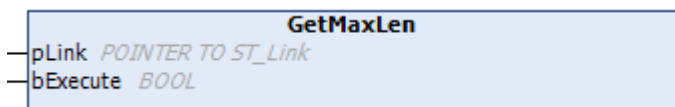
```

VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bEnable    : BOOL;
  nSubscriptionNo : UDINT;
END_VAR
    
```

 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

7.1.3.2.7.2 GetMaxLen



Die Methode GetMaxLen dient zur Abfrage der maximal akzeptierten Länge der Zeichenkette des OCA-Objektes (OcaStringActuator). Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stMaxLen ausgegeben.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

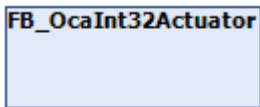
```

VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
END_VAR

```

Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

7.1.3.2.8 FB_OcaInt32Actuator

Der Funktionsbaustein FB_OcaInt32Actuator erweitert den Funktionsbaustein FB_OcaBasicActuator um Eigenschaften und Methoden zur Handhabung von OcaInt32Actuator-Objekten.

Kein expliziter Aufruf

i Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie

```

FB_OcaRoot
  FB_OcaWorker
    FB_OcaActuator
      FB_OcaBasicActuator
        FB_OcaInt32Actuator

```

Methoden

Name	Definitionsort	Beschreibung
GetValue [► 49]	Lokal	Diese Methode dient zur Abfrage einer vorzeichenbehafteten Eigenschaft (ganzzahliger, vorzeichenbehafteter Datentyp, 32 Bit).
SeValue [► 49]	Lokal	Mit dieser Methode kann der Wert der Eigenschaft gesetzt werden.
RNtfValue [► 49]	Lokal	Diese Methode liest eingehende „Notifications“ (Benachrichtigungen)

 **Eigenschaften**

Name	Typ	Zugriff	Definitionsart	Beschreibung
stSetting	ST_INT32Property [▶ 55]	Get	Lokal	Beinhaltet den Wert (ganzzahliger, vorzeichenbehafteter Datentyp, 32 Bit).
stSettingMAX	ST_INT32Property [▶ 55]	Get	Lokal	Beinhaltet den Maximalwert (ganzzahliger, vorzeichenbehafteter Datentyp, 32 Bit).
stSettingMIN	ST_INT32Property [▶ 55]	Get	Lokal	Beinhaltet den Minimalwert (ganzzahliger, vorzeichenbehafteter Datentyp, 32 Bit).

● AES70-Standard

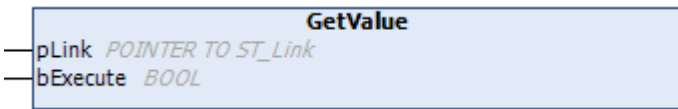
i Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.oaalliance.com.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.2.8.1 GetValue / SetValue / RntfValue

GetValue



Die Methode GetValue dient zur Abfrage der Eigenschaft des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in den Eigenschaften stSetting, stSettingMAX und stSettingMIN ausgegeben.

● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

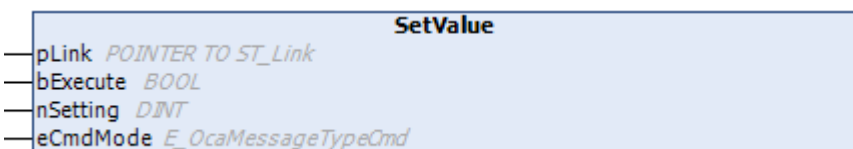
Syntax

```
VAR_INPUT
  pLink   : POINTER TO ST_Link;
  bExecute : BOOL;
END_VAR
```

 **Eingänge**

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

SetValue



Mit der Methode SetValue kann Eigenschaft eines OCA-Objektes gesetzt werden. Wenn die Methode erfolgreich ausgeführt wurde, wird das Resultat in der Eigenschaft stSetting ausgegeben.

● Zyklischer Methodenaufruf

I Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bExecute   : BOOL;
  nSetting   : UDINT;
  eCmdMode   : E_OcaMessageTypeCmd;
END_VAR
```

📌 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSetting	UDINT	Wert, den die Eigenschaft des OCA-Objektes erhalten soll.
eCmdMode	OcaMessageTypeCmd	Abhängig davon, ob eine Bestätigung erforderlich ist oder nicht wird diese Eingangsvariable auf E_OcaMessageTypeCmd.OcaCmdRrq oder E_OcaMessageTypeCmd.OcaCmd gesetzt.

RntfValue

RntfValue	
pLink	POINTER TO ST_Link
bEnable	BOOL
nSubscriptionNo	UDINT

Wenn das OCA-Objekt aufgefordert wird, Änderungen bekanntzugeben (mittels der Methode [Subscription](#) [▶ 23]), werden mit der Methode RntfValue eingehende „Notifications“ (Benachrichtigungen) gelesen.

● Zyklischer Methodenaufruf

I Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

Syntax

```
VAR_INPUT
  pLink      : POINTER TO ST_Link;
  bEnable    : BOOL;
  nSubscriptionNo : UDINT;
END_VAR
```

📌 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

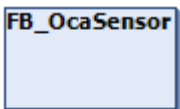
7.1.3.3 Sensor-Funktionsbausteine

Sensoren ermöglichen die Abfrage verschiedener Parameter.

AES70-Standard

i Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.oaalliance.com.

7.1.3.3.1 FB_OcaSensor

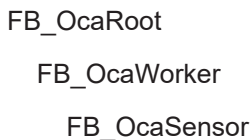


Der Funktionsbaustein FB_OcaSensor erweitert den Funktionsbaustein FB_OcaWorker und ist der Funktionsbaustein, von dem alle Funktionsbausteine erben, die der Kategorie Sensor zugeordnet sind.

Kein expliziter Aufruf

i Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie



Methoden

Name	Definitionsort	Beschreibung
GetReadingState [▶ 51]	Lokal	Mit dieser Methode kann abgefragt werden, ob der gelesene Wert gültig ist oder nicht. Wenn die Abfrage erfolgreich war, wird das Resultat in der Eigenschaft stReadingState ausgegeben.

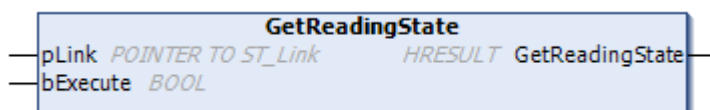
Eigenschaften

Name	Typ	Zugriff	Definitionsort	Beschreibung
stReadingState	ST_SensorReadingState [▶ 55]	Get	Lokal	Zeigt an, ob der abgefragte Wert gültig ist oder nicht.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.3.1.1 GetReadingState



Die Methode GetReadingState dient zur Abfrage der Eigenschaft ReadingState des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in den Eigenschaften stReadingState ausgegeben.

● Zyklischer Methodenaufruf



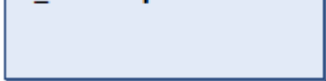
Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

🔌 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

7.1.3.3.2 FB_OcaTemperatureSensor

FB_OcaTemperatureSensor



Der Funktionsbaustein FB_OcaTemperatureSensor bietet Eigenschaften und Methoden zur Abfrage von OCA-Objekten, die einen Temperaturwert abbilden.

● Kein expliziter Aufruf



Da sich kein Code im Rumpf des Funktionsbausteins befindet, sollte dieser nicht explizit aufgerufen werden. Verwenden Sie stattdessen die entsprechenden Methoden des Funktionsbausteins.

Vererbungshierarchie

FB_OcaRoot

 FB_OcaWorker

 FB_OcaSensor

 FB_OcaTemperatureSensor

☰ Methoden

Name	Definitionsort	Beschreibung
GetReading	Lokal	Dient zur Abfrage der Eigenschaften Reading, minReading und maxReading des OCA-Objekts. Wenn die Abfrage erfolgreich war, wird das Resultat in den Eigenschaften stReading, stReadingMAX und stReadingMIN ausgegeben.
RNtfGain	Lokal	Wenn das OCA Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode Subscription [▶ 231]), werden mit dieser Methode eingehende „Notifications“ (Benachrichtigungen) gelesen.

● AES70-Standard



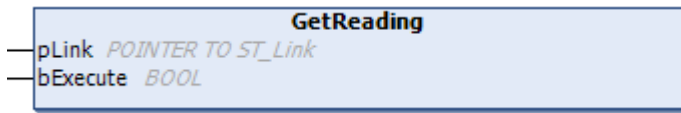
Die Bezeichnung von Variablen und Funktionsbausteinen wurde, sofern möglich, dem AES70-Standard angepasst. Informationen zu diesem Kommunikationsprotokoll finden Sie auf www.aes.org und www.ocaalliance.com.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.3.2.1 GetReading / RntfReading

GetReading



Die Methode GetReading dient zur Abfrage der Eigenschaft des OCA-Objektes. Wenn die Abfrage erfolgreich war, wird das Resultat in den Eigenschaften stReading, stReadingMAX und stReadingMIN ausgegeben

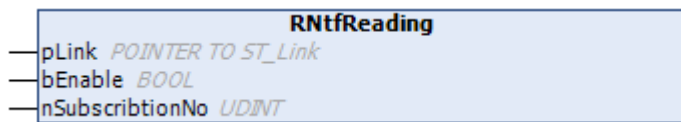
● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

🔌 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.

RntfReading



Wenn das OCA-Objekt aufgefordert wurde, Änderungen bekanntzugeben (mittels der Methode Subscription [▶ 23]), werden mit der Methode RntfReading eingehende „Notifications“ (Benachrichtigungen) gelesen.

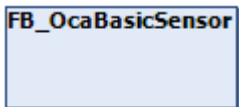
● Zyklischer Methodenaufruf

i Da zwischen dem Absenden und der Antwort des Gerätes einige SPS-Zyklen liegen können, sollte diese Methode zyklisch ausgeführt werden.

🔌 Eingänge

Name	Typ	Beschreibung
pLink	POINTER TO ST_Link	Zeiger zum Herstellen einer Verbindung zwischen OCA-Objekten und dem OCA Device.
bExecute	BOOL	Mit einer positiven Flanke am Eingang bExecute wird die Methode ausgeführt.
nSubscriptionNo	UDINT	Eindeutige Nummer zur Lokalisierung der jeweiligen Subscription.

7.1.3.3.3 FB_OcaBasicSensor



Der Funktionsbaustein FB_OcaBasicSensor erweitert den Funktionsbaustein [FB_OcaSensor](#) [► 51] und ist der Funktionsbaustein, von dem alle Funktionsblöcke erben, die der Kategorie BasicSensor zugeordnet sind. FB_OcaBasicSensor besitzt weder eigene Eigenschaften noch Methoden.

Vererbungshierarchie

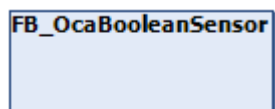
```

FB_OcaRoot
  FB_OcaWorker
    FB_OcaSensor
      FB_OcaBasicSensor
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022.2 oder neuer	PC oder CX (x64, x86, ARM)	Tc3_AES70

7.1.3.3.4 FB_OcaBooleanSensor



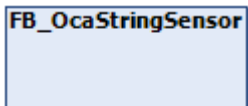
Der Funktionsbaustein FB_OcaBooleanSensor gleicht in der Funktionalität dem Funktionsbaustein [FB_OcaBooleanActuator](#) [► 42]. Da Sensoren nur gelesen werden können, fehlt jedoch die entsprechende Set-Methode.

Vererbungshierarchie

```

FB_OcaRoot
  FB_OcaWorker
    FB_OcaSensor
      FB_OcaBasicSensor
        FB_OcaBooleanSensor
  
```

7.1.3.3.5 FB_OcaStringSensor



Der Funktionsbaustein FB_OcaBooleanSensor gleicht in der Funktionalität dem Funktionsbaustein [FB_OcaStringActuator](#) [► 45]. Da Sensoren nur gelesen werden können, fehlt jedoch die entsprechende Set-Methode.

Vererbungshierarchie

```

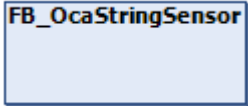
FB_OcaRoot
  FB_OcaWorker
  
```

FB_OcaSensor

FB_OcaBasicSensor

FB_OcaStringSensor

7.1.3.3.6 FB_Ocalnt32Sensor



Der Funktionsbaustein FB_OcaBooleanSensor gleicht in der Funktionalität dem Funktionsbaustein [FB_Ocalnt32Actuator \[► 48\]](#). Da Sensoren nur gelesen werden können, fehlt jedoch die entsprechende Set-Methode.

Vererbungshierarchie

FB_OcaRoot

FB_OcaWorker

FB_OcaSensor

FB_OcaBasicSensor

FB_Ocalnt32Sensor

7.2 Datentypen

7.2.1 Strukturen, die zur Darstellung der Eigenschaften von OCA-Objekten verwendet werden

ST_OcaProperty

Von dieser Struktur erben alle nachfolgend angeführten Strukturen.

```
TYPE ST_OcaProperty :
STRUCT
  eState: E_OcaStatus := E_OcaStatus.Undefined;
  sPropString: STRING(5) := '00p00';
END_STRUCT
END_TYPE
```

Siehe auch: [E_OcaStatus \[► 57\]](#)

ST_BooleanProperty

```
TYPE ST_BooleanProperty EXTENDS ST_OcaProperty :
STRUCT
  bVal: BOOL := FALSE;
END_STRUCT
END_TYPE
```

ST_ClassIdProperty

```
TYPE ST_ClassIdProperty EXTENDS ST_OcaProperty :
STRUCT
  nVal: ST_OcaClassId;
END_STRUCT
END_TYPE
```

ST_FLOAT32Property

```
TYPE ST_FLOAT32Property EXTENDS ST_OcaProperty :
STRUCT
    fVal: REAL;
END_STRUCT
END_TYPE
```

ST_INT16Property

```
TYPE ST_INT16Property EXTENDS ST_OcaProperty :
STRUCT
    nVal: INT;
END_STRUCT
END_TYPE
```

ST_INT32Property

```
TYPE ST_INT32Property EXTENDS ST_OcaProperty :
STRUCT
    nVal: DINT;
END_STRUCT
END_TYPE
```

ST_INT8Property

```
TYPE ST_INT8Property EXTENDS ST_OcaProperty :
STRUCT
    nVal: BYTE;
END_STRUCT
END_TYPE
```

ST_MuteStateProperty

```
TYPE ST_MuteStateProperty EXTENDS ST_OcaProperty :
STRUCT
    eVal: E_OcaMuteState := E_OcaMuteState.Unmuted ;
END_STRUCT
END_TYPE
```

Siehe auch: [E_OcaMuteState](#) [[▶ 58](#)]

ST_PolarityStateProperty

```
TYPE ST_PolarityStateProperty EXTENDS ST_OcaProperty :
STRUCT
    eVal: E_OcaPolarityState := E_OcaPolarityState.NonInverted;
END_STRUCT
END_TYPE
```

ST_PositionEnabledProperty

```
TYPE ST_PositionEnabledProperty EXTENDS ST_BooleanProperty :
STRUCT
    nIndex: UINT;
END_STRUCT
END_TYPE
```

ST_PositionNameProperty

```
TYPE ST_PositionNameProperty EXTENDS ST_StringProperty :
STRUCT
    nIndex: UINT;
END_STRUCT
END_TYPE
```

ST_SensorReadingState

```
TYPE ST_SensorReadingState EXTENDS ST_OcaProperty :
STRUCT
    eVal: E_OcaSensorReadingState := E_OcaSensorReadingState.eUnknown;
END_STRUCT
END_TYPE
```


ST_StringProperty

```

TYPE ST_StringProperty EXTENDS ST_OcaProperty :
STRUCT
  stVal: ST_OcaString;
END_STRUCT
END_TYPE

```

ST_SubscriptionManagerState

```

TYPE ST_SubscriptionManagerState EXTENDS ST_OcaProperty :
STRUCT
  eVal: E_OcaSubscriptionManagerState;
END_STRUCT
END_TYPE

```

ST_TemperatureProperty

```

TYPE ST_TemperatureProperty EXTENDS ST_OcaProperty:STRUCT
  stVal: ST_OcaTemperature;
END_STRUCT
END_TYPE

```

Siehe auch: [ST_OcaTemperature](#) [[▶ 58](#)]

ST_UINT16Property

```

TYPE ST_UINT16Property EXTENDS ST_OcaProperty :
STRUCT
  nVal: UINT;
END_STRUCT
END_TYPE

```

ST_UDINT32Property

```

TYPE ST_UDINT32Property EXTENDS ST_OcaProperty :
STRUCT
  nVal: UDINT;
END_STRUCT
END_TYPE

```

ST_UINT8Property

```

TYPE ST_UINT8Property EXTENDS ST_OcaProperty :
STRUCT
  nVal: SINT;
END_STRUCT
END_TYPE

```

7.2.2 E_OcaStatus

The status code that identifies the result of the method invocation the response belongs to. E_OcaStatus has a size of 1 byte.

```

(*Status codes returned from method calls*)
{attribute 'qualified_only'}
TYPE E_OcaStatus :
(
  Ok := 0,
  ProtocolVersionError := 1,
  DeviceError := 2,
  Locked := 3,
  BadFormat := 4,
  BadOno := 5,
  ParameterError := 6,
  ParameterOutOfRange := 7,
  NotImplemented := 8,
  InvalidRequest := 9,
  ProcessingFailed := 10,
  BadMethod := 11,
  PartiallySucceeded := 12,
  Timeout := 13,
  BufferOverflow := 14,
  DecodingError := 20,

```

```

    Undefined := 21
) BYTE;
END_TYPE

```

7.2.3 E_OcaMuteState

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_OcaMuteState:
(
    Muted:= 1,
    Unmuted:=2
);
END_TYPE

```

7.2.4 ST_OcaTemperature

```

TYPE ST_OcaTemperature :
STRUCT
    fDegreesCelsius: REAL; //Value in Degrees Celsius
END_STRUCT
END_TYPE

```

7.2.5 ST_OcaDeviceInfo

Structure used to show informations about OCA devices.

```

TYPE ST_OcaDeviceInfo:
STRUCT
    tDeviceEnabledSince:    DATE_AND_TIME;
    tClientConnectedSince: DATE_AND_TIME; //yet not used - for further extensions
    tLastSentMsg:          DATE_AND_TIME;
    tLastReceivedMsg:     DATE_AND_TIME;
    aSentMessages:        ARRAY[E_OcaMessageType.OcaCmd..E_OcaMessageType.OcaKeepAlive] OF UDINT;
    aReceivedMessages:    ARRAY[E_OcaMessageType.OcaCmd..E_OcaMessageType.OcaKeepAlive] OF UDINT;
END_STRUCT
END_TYPE

```

7.2.6 E_OcaMessageType

Indicates the type of the message

```

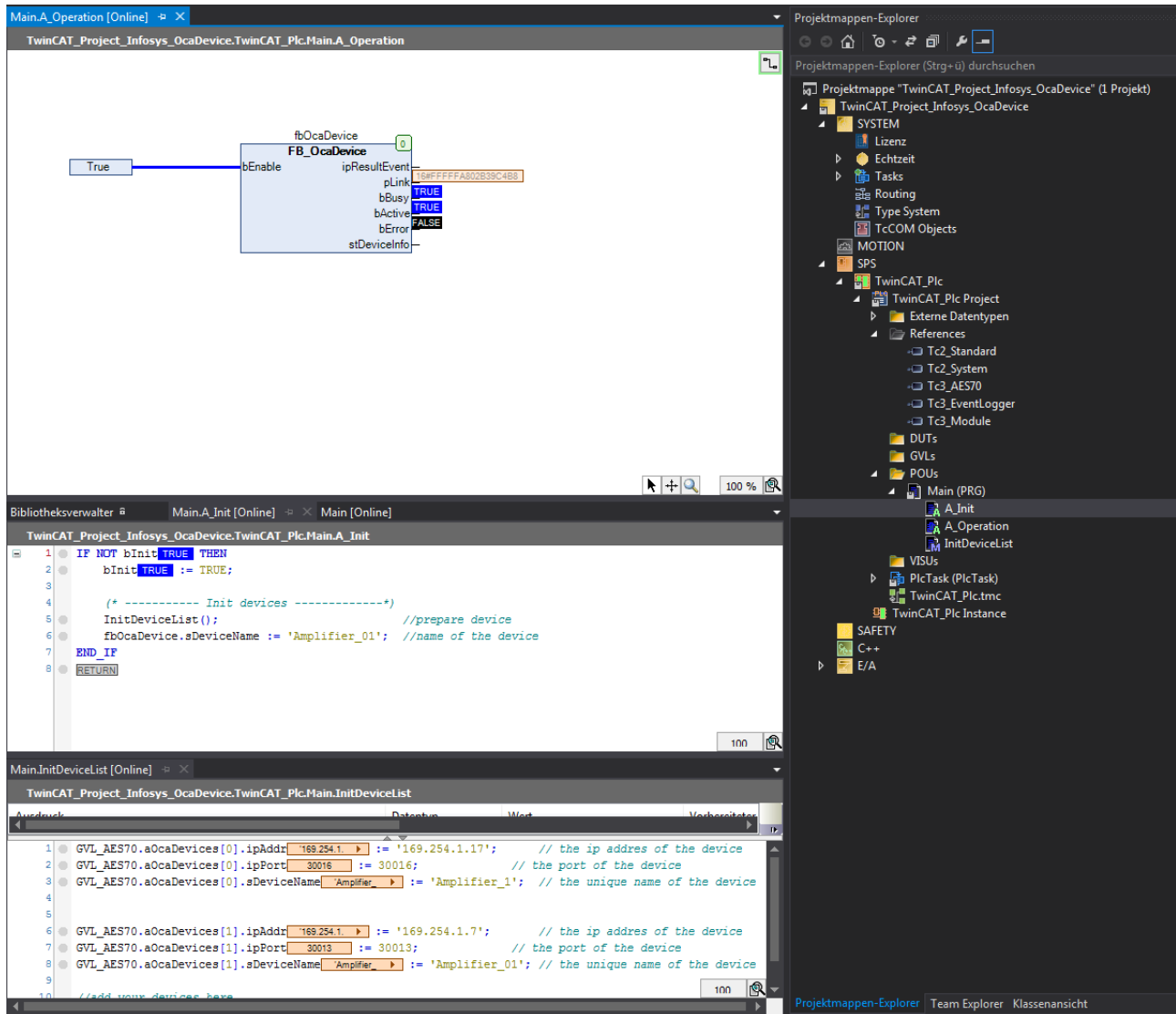
{attribute 'qualified_only'}
TYPE E_OcaMessageType:
(
    OcaCmd:= 0,           // Command - no Response Required
    OcaCmdRrq:= 1,       // Command - Response Required
    OcaNtf:= 2,          // Notification
    OcaRsp:= 3,          // Response (to a command or notification)
    OcaKeepAlive:= 4,    // Keep-alive message used for device supervision.
    Idle:=7
) BYTE;
END_TYPE

```

8 Beispiele

8.1 Beispiel zur Verwendung des Funktionsbausteins FB_OcaDevice

Das Beispiel zeigt, wie der Funktionsbaustein FB_OcaDevice zu handhaben und parametrieren ist. Der Funktionsbaustein FB_OcaDevice bildet die Basis für die Verwendung weiterer Funktionsbausteine, die zum Auslesen und Ändern von OCA-Objekten eines OCA Devices verwendet werden können.

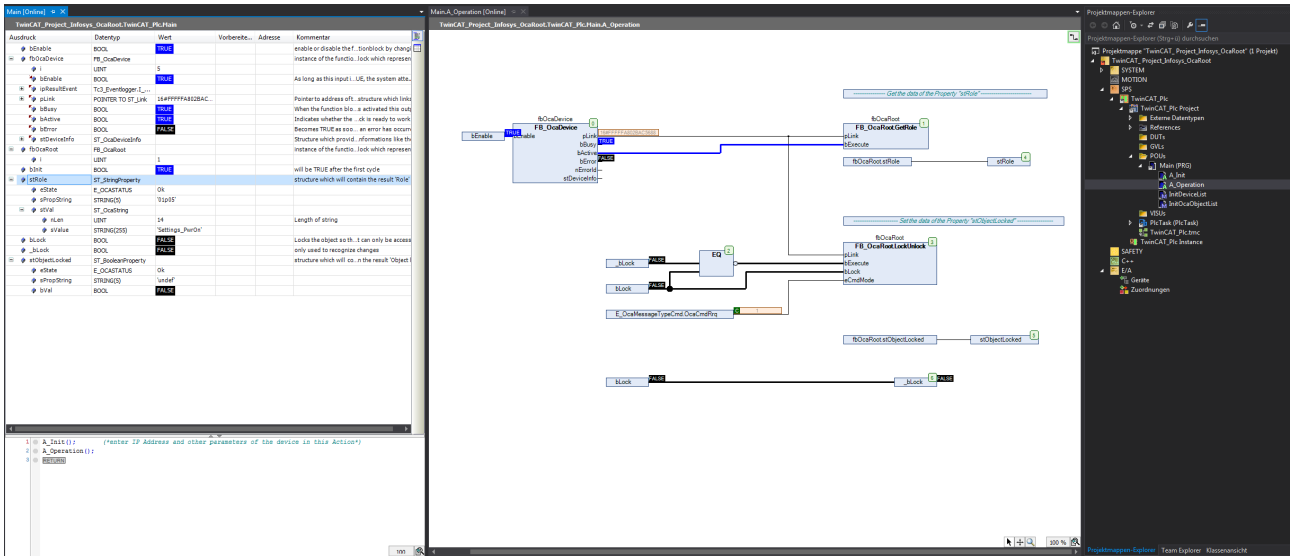


Bei diesem Beispiel wird angenommen, dass ein Gerät, das den AES70 Standard unterstützt, angeschlossen ist.

Download: https://infosys.beckhoff.com/content/1031/tf8810_tc3_aes70/Resources/4223793163.zip

8.2 Beispiel zur Verwendung des Funktionsbausteins FB_OcaRoot

Das Beispiel zeigt, wie der Funktionsbaustein FB_OcaRoot verwendet werden kann. Der Funktionsbaustein FB_OcaRoot bietet Basisfunktionalitäten und vererbt diese an alle Funktionsbausteine der Bibliothek, die zum Auslesen und Ändern von OCA-Objekten werden, zum Beispiel FB_OcaMute, FB_OcaSwitch, ...

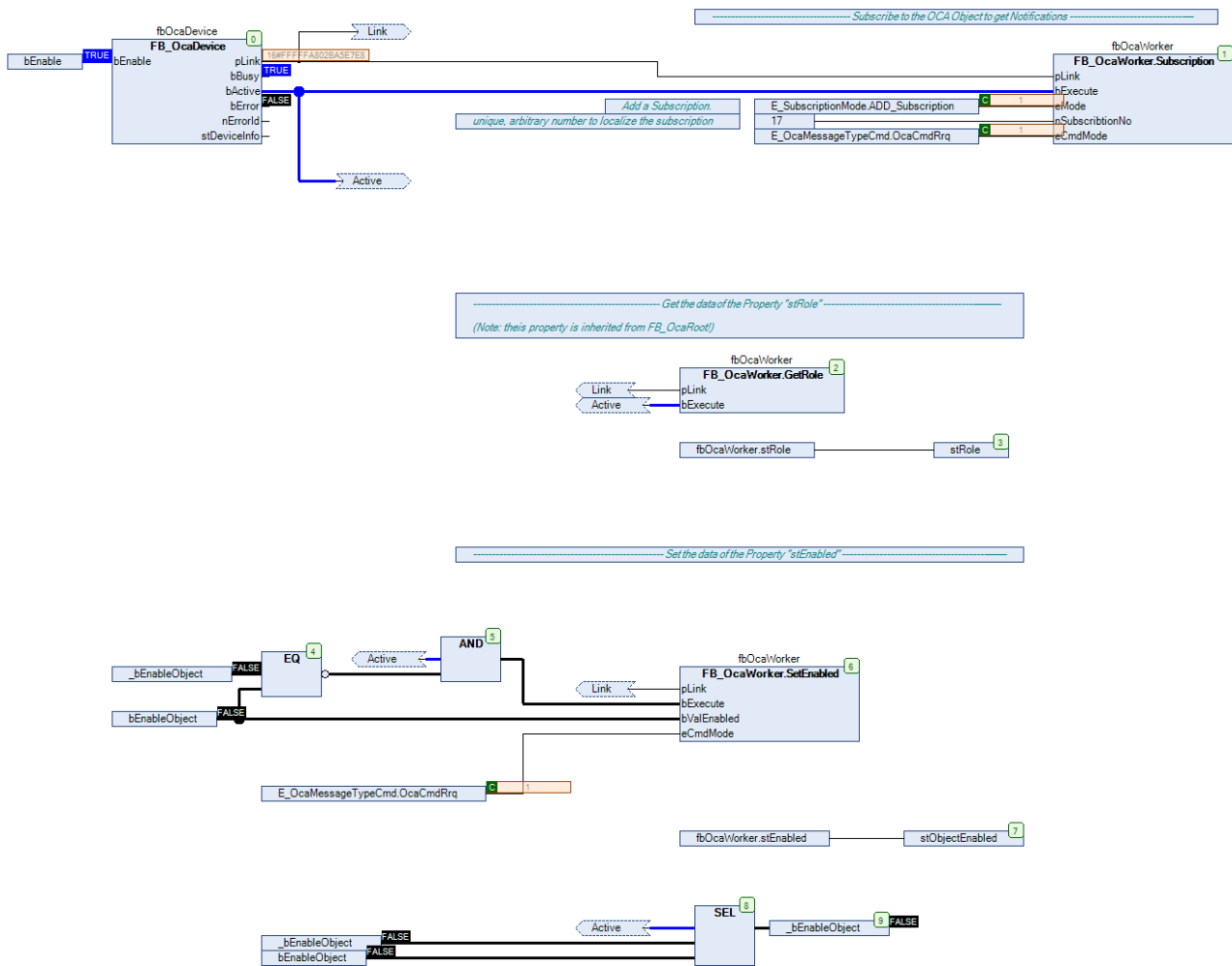


Bei diesem Beispiel wird angenommen, dass ein Gerät, das den AES70 Standard unterstützt, angeschlossen ist.

Download: https://infosys.beckhoff.com/content/1031/tf8810_tc3_aes70/Resources/4223020171.zip

8.3 Beispiel zur Verwendung des Funktionsbausteins FB_OcaWorker

Das Beispiel zeigt, wie der Funktionsbaustein FB_OcaWorker verwendet werden kann.

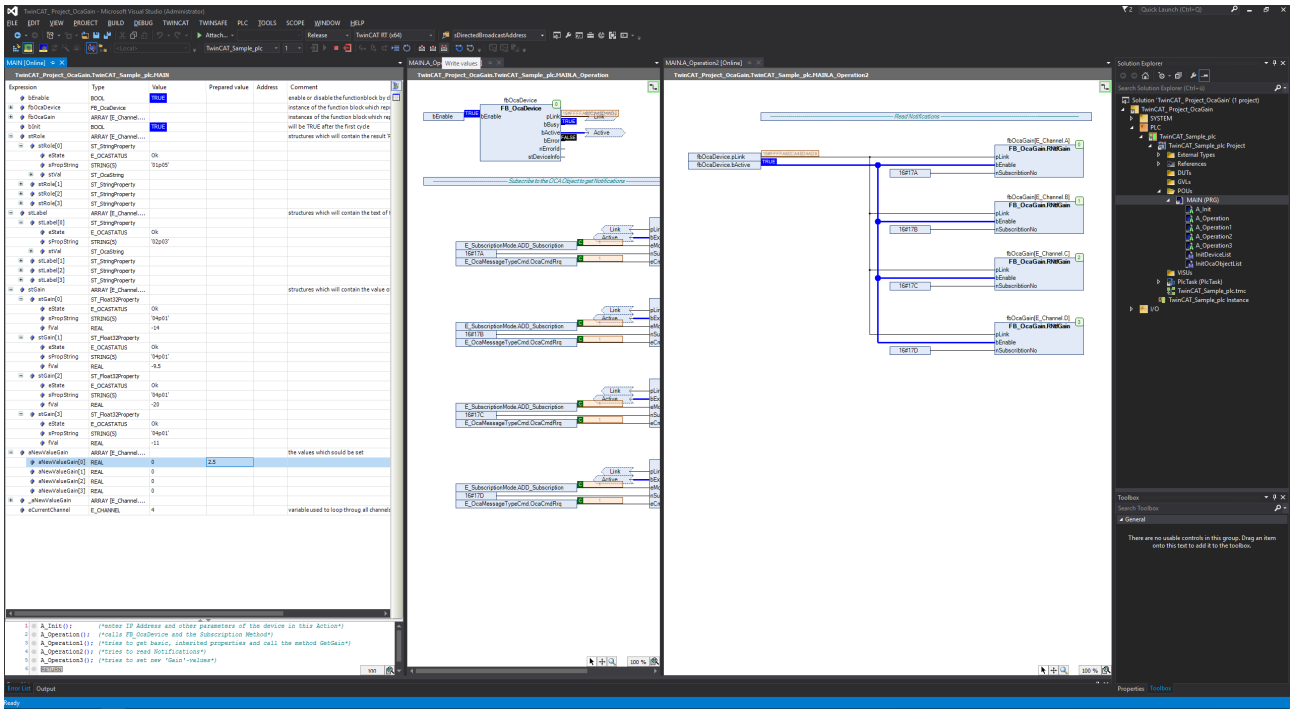


Bei diesem Beispiel wird angenommen, dass ein Gerät, das den AES70 Standard unterstützt, angeschlossen ist.

Download: https://infosys.beckhoff.com/content/1031/tf8810_tc3_aes70/Resources/4223021835.zip

8.4 Beispiel zur Verwendung des Funktionsbausteins FB_OcaGain

Das Beispiel zeigt, wie der Funktionsbaustein FB_OcaGain zu handhaben und parametrieren ist. Zum Einsatz kommen auch Methoden und Eigenschaften, die von FB_OcaWorker und somit auch von FB_OcaRoot geerbt werden.



Bei diesem Beispiel wird angenommen, dass ein Gerät, das den AES70 Standard unterstützt, angeschlossen ist.

Download: https://infosys.beckhoff.com/content/1031/tf8810_tc3_aes70/Resources/4223791499.zip

9 Support und Service

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

Downloadfinder

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

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den [lokalen Support und Service](#) zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0
E-Mail: info@beckhoff.com
Internet: www.beckhoff.com

Mehr Informationen:
www.beckhoff.de/tf8810

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

