

BECKHOFF New Automation Technology

Handbuch | DE

TwinCAT 3

PLC-Bibliothek: PLink

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	Programmierung	10
3.1	FB_PJLink.....	10
3.1.1	Methoden	11
3.2	Strukturen, Enumerationen	13
3.2.1	ST_PJLink_Projector_Status	13
3.2.2	ST_PJLink_AVMT_Status.....	14
3.2.3	ST_PJLink_CLSS_Status	14
3.2.4	ST_PJLink_ERST_Errors	14
3.2.5	ST_PJLink_ERST_Status.....	14
3.2.6	ST_PJLink_FILT_Status	15
3.2.7	ST_PJLink_FREZ_Status	15
3.2.8	ST_PJLink_INF1_Status.....	15
3.2.9	ST_PJLink_INF2_Status.....	15
3.2.10	ST_PJLink_INFO_Status.....	15
3.2.11	ST_PJLink_INNM_Status	15
3.2.12	ST_PJLink_INPT_Status	15
3.2.13	ST_PJLink_INST_Status	16
3.2.14	ST_PJLink_IRES_Status	16
3.2.15	ST_PJLink_MVOL_Status	16
3.2.16	ST_PJLink_LAMP_Status.....	16
3.2.17	ST_PJLink_Lamp_int.....	16
3.2.18	ST_PJLink_NAME_Status	16
3.2.19	ST_PJLink_POWR_Status	17
3.2.20	ST_PJLink_RFIL_Status.....	17
3.2.21	ST_PJLink_RLMP_Status.....	17
3.2.22	ST_PJLink_RRES_Status.....	17
3.2.23	ST_PJLink_SNUM_Status.....	17
3.2.24	ST_PJLink_SVER_Status.....	17
3.2.25	ST_PJLink_SVOL_Status.....	18
3.2.26	E_PJLink_InputType	18
3.2.27	E_PJLink_MuteType	18
3.2.28	E_PJLink_Steps.....	18
3.2.29	E_PJLink_UpdateType	18
3.2.30	PJLink Error Codes.....	19
4	Beispiel	20

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

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

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

Disclaimer

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

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

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

Marken

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

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



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.

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

Fremdmarken

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

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

Signalwörter

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

Warnungen vor Personenschäden

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnung vor Umwelt- oder Sachschäden

HINWEIS

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

Information zum Umgang mit dem Produkt



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

1.3 Hinweise zur Informationssicherheit

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

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

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

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

2 Übersicht

PJLink ist ein Kommunikationsprotokoll, basierend auf TCP/IP-Kommunikation, welches benutzt wird, um eine Überwachung und Steuerung von Projektoren über das Netzwerk zu ermöglichen. Es wurde von der JBMIA (Japan Business Machine and Information System Industries Association) erstellt.

PJLink ist ein Standard, der nicht auf einen Projektorhersteller oder auf ein spezielles Projektormodell limitiert ist. Dadurch ermöglicht PJLink eine zentrale Steuerung von Projektoren verschiedener Hersteller. Es beinhaltet Grundbefehle und Statusabfragen, die zur Steuerung und Überwachung wichtig sind.

Die Implementierung von PJLink deckt den Befehlsumfang der PJLink-Klassen 1 und 2 des „Control Protocol“ ab. Zusätzlich wird die „Security Authorization“- Funktion unterstützt. Das „Search Protocol“ und das „Status Notification Protocol“ (beide basierend auf UDP-Kommunikation und Teil der Class 2 Spezifikation) sind in der Implementierung nicht enthalten.

Befehle:

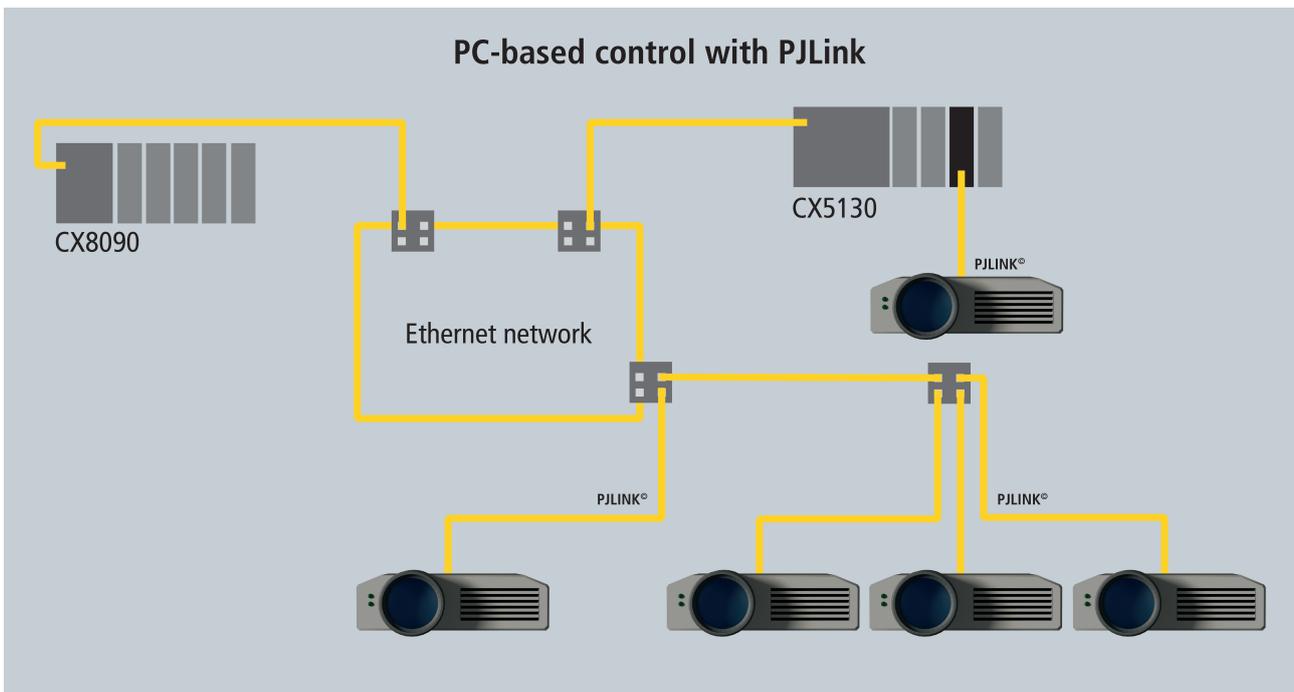
- Projektor an/aus,
- Wechseln des verwendeten Medienanschlusses,
- Video-/Audio Mute.
- Freeze an/aus,*
- Lautstärke Lautsprecher erhöhen/verringern,*
- Lautstärke Mikrofon erhöhen/verringern.*

Abfragen:

- Powerstatus,
- Verfügbare Medienanschlüsse,
- derzeitig benutzter Medienanschluss,
- Mutezustand,
- Errorstatus,
- Lampenstatus und -nutzungsdauer,
- Projektor Name, Modell Name, Hersteller Name, PJLink-Klasse,
- Seriennummer, Softwareversion, Name Medienanschluss,*
- derzeitige Bildauflösung, empfohlene Bildauflösung,*
- Nutzungsdauer Filter, Seriennummer Filter, Seriennummer Lampe(n),*
- Freezestatus.*

**Befehle und Statusabfragen werden nur von Projektoren der PJLink-Klasse 2 unterstützt.*

Jede PJLink-Nachricht wird vom jeweiligen Projektor beantwortet.



Weitere Informationen finden Sie auf unserer Webseite unter:
[PC-based Control für Bühnen- und Showtechnik](#)

Die PLink-Spezifikationen und Informationen zu PLink finden Sie auf der Webseite:
<http://pmlink.jbmia.or.jp/english/>.

PLink trademark and logo are trademarks applied for registration or are already registered in Japan, the United States of America and other countries and areas.

Die Implementierung von PLink stellt lediglich ein Beispiel dar. Es wird kein Anspruch auf Vollständigkeit erhoben.

Systemanforderungen:

Technische Daten	Anforderung
TwinCAT-Version	TwinCAT 3.1 Build 4022.20 oder höher
Visual Studio-Version	Visual Studio 2013 oder höher
Erforderliche TwinCAT-Lizenzen	TF6310-Lizenz

3 Programmierung

3.1 FB_PJLink

FB_PJLink	
bEnable <i>BOOL</i>	<i>BOOL</i> bBusy
sProjector <i>T_IPv4Addr</i>	<i>E_PJLink_step</i> eStep
sServerNetId <i>T_AmsNetId</i>	<i>BOOL</i> bError
sSecurityPW <i>STRING(32)</i>	<i>I_TcMessage</i> ipResultMessage
bKeepAliveSocket <i>BOOL</i>	<i>UDINT</i> nErrId
tStatusUpdateTime <i>TIME</i>	<i>STRING(20)</i> sErrBlock
eTraceLevel <i>TcEventSeverity</i>	<i>ST_PJLink_Projector_Status</i> stProjStatus

Dieser Funktionsbaustein ermöglicht die Kommunikation zu einem Projektor über TCP/IP und kann zum Steuern und Überwachen verwendet werden. Die Kommunikation zu einem Projektor wird durch den Funktionsbaustein automatisch aufgebaut. Jede Information, die von der Steuerung empfangen wird, wird ausgewertet und in der Struktur *ST_PJLink_Projector_Status* (Ausgangsparameter *stProjStatus*) ausgegeben. Darüber hinaus gibt es eine automatische Status-Update Funktionalität, die periodisch ausgeführt wird. Informationen, Warnungen und Errors werden mit Hilfe des TC3 Eventloggers ausgegeben.

Voraussetzungen für FB_PJLink:

- TF6310 „TwinCAT TCP/IP“ ,
- Einzubindende Bibliothek:Tc2_Tcplp (Communication),
- Entwicklungsumgebung: TwinCAT v3.1.0. oder höher.

● **Projektoreinstellungen beachten**

i Folgende Einstellungen müssen beim Projektor eingerichtet werden:

- Aktivierung der PJLink-Funktionalität,
- Deaktivierung der PJLink-Security-Funktionalität (optional),
- Aktivierung der Netzwerk-Funktionalität,
- Deaktivierung der ECO-Funktion (Abschaltung der Netzwerk-Funktionalität im Standby-Modus).

Das Senden von Befehlen und Statusabfragen zum Projektor, wird durch Aufrufen von Methoden und Setzen der jeweiligen Input-Parameter durchgeführt:

- *Power* (Ändern des Powerstatus (An-/Ausschalten des Projektors)),
- *Mute* (Ändern des AV-Mutestatus des Projektors (Aktivieren/Deaktivieren von Video-, Audio-, Video/Audio-Mute)),
- *Input* (Ändern des Medieneingangs des Projektors (RGB, Video, Digital, Storage, Network)),
- *Update* (Senden spezifischer Statusabfragen),
- *Freeze* (Ändern des Freezestatus des Projektors (Aktivieren/Deaktivieren der Freeze-Funktion)),
- *Volume* (Erhöhen/Verringern der Lautstärke der Lautsprecher des Projektors),
- *MicVolume* (Erhöhen/Verringern der Lautstärke des Mikrofons des Projektors).

● **Lizenz**

i Für die Verwendung von FB_PJLink wird sowohl eine TwinCAT 3.1-Lizenz, als auch eine TF6310 „TwinCAT TCP/IP“- Lizenz benötigt.

VAR_INPUT

```
bEnable          : BOOL;
sProjector       : T_IPv4Addr;
sServerNetID    : T_AmsNetId;
sSecurityPW     : STRING(32);
bKeepAliveSocket : BOOL := FALSE;
tStatusUpdateTime : TIME := T#5S;
eTraceLevel     : TcEventSeverity;
```

bEnable: Ein Setzen auf `TRUE` aktiviert den Funktionsbaustein. Die Kommunikation zu einem Projektor wird aufgebaut. Bei `FALSE` wird die Kommunikation beendet und der Funktionsbaustein deaktiviert.

sProjector: Die lokale IP-Adresse (IPv4) des TCP/IP Client/Server-Sockets als String (z.B. '2.168.0.5'). Für den Default-Netzwerkadapter kann auch ein Leerstring angegeben werden.

sServerNetID: String mit der Netzwerkadresse des TwinCAT TCP/IP Connection Servers. Für den lokalen Rechner (default) kann auch ein Leerstring angegeben werden.

sSecurityPW: Passwort für die Security-Authorization-Funktion (optional). Das identische (Case sensitive) Passwort muss in den Einstellungen des Projektors eingegeben werden.

bKeepAliveSocket: Aktiviert bei `TRUE` einen Keep-Alive-Modus für die Kommunikation zum Projektor. Es wird lediglich ein Socket geöffnet, der dauerhaft offengehalten wird. Im Fehlerfall verbleibt der Funktionsbaustein in einen Error-Modus. Durch `FALSE` (empfohlen) wird vor dem Senden eines Befehls ein neuer Socket geöffnet und danach wieder geschlossen. Im Fehlerfall wird automatisch ein erneuter Verbindungsvorgang ausgelöst.

tStatusUpdateTime: Updateintervall der Statusinformationsabfrage vom Projektor. (Default=5s; höhere Updatezeit `tStatusUpdateTime` reduziert den Datenverkehr.)

eTraceLevel: Bestimmt die Eventarten, die über den Eventlogger abgeschickt werden. Events die von einer größeren oder gleich großen Wichtigkeit sind, als der Wert am Eingangsparameter, werden gesendet. (Beispiel: Wert = Warning → Abschicken von Warnings, Errors und Criticals; Wert=Warning → unter Anderem werden PJLink-Nachrichten geloggt) Typ der Variablen: TcEventSeverity

VAR_OUTPUT

```
bBusy          : BOOL;
eStep          : E_PJLink_step;
bError        : BOOL;
ipResultMessage : I_TcMessage;
nErrID        : UDINT;
sErrBlock     : STRING(20);
stProjStatus  : ST_PJLink_Projector_Status;
```

bBusy: Dieser Ausgang ist gesetzt, solange die Verbindung zum Socket besteht/aufgebaut wird.

eStep: Dieser Ausgangsparameter gibt den aktuellen Schritt in der `FB_PJLink` State machine an.

bError: Wenn ein Fehler auftritt, wird dieser Ausgang gesetzt. (z. B.: Verbindung zum Socket ist nicht möglich).

ipResultMessage: Mit dieser Schnittstelle wird das Abschicken von Events über den TC3-Eventlogger realisiert.

nErrID: Dieser Parameter liefert bei einem gesetzten `bError`-Ausgang die TwinCAT TCP/IP Fehlernummer.

sErrBlock: Liefert weiterführende Informationen zum aufgetretenen Fehler. (Verursachender Funktionsbaustein)

stProjStatus: Struktur mit den Projektorstatus-Informationen. (Nach dem Senden eines Befehls, wird die Antwort des Projektors ausgewertet und in `stProjStatus` ausgegeben.)

Beispiel: Befehl = "%1POWR 1\$R" → Antwort des Projektors = "%1POWR=ERR3" →
`ProjStatus.POWR.Err_UnavailableTime = TRUE.`

3.1.1 Methoden

Power

Mit dieser Methode können einem Projektor Power-Befehle gesendet werden.

```
METHOD Power :hresult
VAR_INPUT
    bOn      : BOOL;
    bExecute : BOOL;
END_VAR
```

bOn: Ist dieser Parameter bei Methodenaufruf `TRUE`, wird ein Power-Einschaltbefehl gesendet. Falls dieser Parameter `FALSE` ist, wird ein Power-Ausschaltbefehl gesendet.

bExecute: Positive Flanke bewirkt bei Methodenaufruf das Senden eines Befehls.

Der Projektor sendet eine Bestätigung, wenn der Befehl erfolgreich umgesetzt werden kann. Durch die Bestätigung wird der Ausgangsparameter `stProjStatus.stPOWR.bSuccessful` für die Dauer eines Zyklus auf `TRUE` gesetzt.

Mute

Mit dieser Methode können einem Projektor Mute-Befehle gesendet werden.

```
METHOD Mute :hresult
VAR_INPUT
    eMuteType : E_PJLINK_MuteType;
    bMuteOn   : BOOL;
    bExecute  : BOOL;
END_VAR
```

eMuteType: Wählen der gewünschten Mute-Art (Audio, Video oder Audio/Video).

bMuteOn: Setzen dieses Parameters auf `TRUE` bewirkt einen Mute-Einschaltbefehl. Falls der Parameter `FALSE` ist, wird der Befehl ein Mute-Ausschaltbefehl.

bExecute: Positive Flanke bewirkt bei Methodenaufruf das Senden eines Befehls.

Der Projektor sendet eine Bestätigung, wenn der Befehl erfolgreich umgesetzt werden kann. Durch die Bestätigung wird der Ausgangsparameter `stProjStatus.stAVMT.bSuccessful` für die Dauer eines Zyklus auf `TRUE` gesetzt.

Input

Mit dieser Methode kann einem Projektor der Befehl zum Umschalten des Medieninputs gesendet werden.

```
METHOD Input :hresult
VAR_INPUT
    eInputType      : E_PJLINK_InputType;
    sInputChannel   : STRING(1);
    bExecute        : BOOL;
END_VAR
```

eInputType: Wählen der gewünschten Input-Art. (RGB, Video, Digital, Storage, Network, Internal)

sInputChannel: Auswählen des jeweiligen Channels. (Werte: 1-9 und A-Z; Siehe Gerätehandbuch.)

bExecute: Positive Flanke bewirkt bei Methodenaufruf das Senden eines Befehls.

Der Projektor sendet eine Bestätigung, wenn der Befehl erfolgreich umgesetzt werden kann. Durch die Bestätigung wird der Ausgangsparameter `stProjStatus.stINPT.bSuccessful` für die Dauer eines Zyklus auf `TRUE` gesetzt.

Update

Mit dieser Methode können einem Projektor Statusabfragen gesendet werden.

```
METHOD Update :hresult
VAR_INPUT
    eUpdateType : E_PJLINK_UpdateType;
    bExecute    : BOOL;
END_VAR
```

eUpdateType: Wählen der gewünschten Statusabfrage.

bExecute: Positive Flanke bewirkt bei Methodenaufruf das Senden eines Befehls.

Antworten des Projektors werden im Ausgangsparameter `stProjStatus` eingetragen.

Freeze

Mit dieser Methode kann einem Projektor der Befehl zum Umschalten des Freezestatus gesendet werden.

```
METHOD Freeze :hresult
VAR_INPUT
    bFreeze    : BOOL;
    bExecute   : BOOL;
END_VAR
```

bFreeze: Ist dieser Parameter bei Methodenaufruf `TRUE`, wird die Freeze-Funktion bei dem Projektor aktiviert. Das Senden des Befehls mit dem Parameterwert `FALSE` hat das Deaktivieren der Freeze-Funktion zur Folge.

bExecute: Positive Flanke bewirkt bei Methodenaufruf das Senden eines Befehls.

Der Projektor sendet eine Bestätigung, wenn der Befehl erfolgreich umgesetzt werden kann. Durch die Bestätigung wird der Ausgangsparameter `stProjStatus.stFREZ.bSuccessful` für die Dauer eines Zyklus auf `TRUE` gesetzt.

Volume

Mit dieser Methode kann die Lautstärke der Lautsprecher des Projektors geändert werden.

```
METHOD Volume :hresult
VAR_INPUT
    bIncrease   : BOOL;
    bExecute    : BOOL;
END_VAR
```

bIncrease: Ist dieser Parameter bei Methodenaufruf `TRUE`, wird ein Befehl zum Erhöhen der Lautstärke gesendet werden. Das Senden des Befehls mit dem Parameterwert `FALSE`, hat ein Verringern der Lautstärke zur Folge.

bExecute: Positive Flanke bewirkt bei Methodenaufruf das Senden eines Befehls.

Der Projektor sendet eine Bestätigung, wenn der Befehl erfolgreich umgesetzt werden kann. Durch die Bestätigung wird der Ausgangsparameter `stProjStatus.stSVOL.bSuccessful` für die Dauer eines Zyklus auf `TRUE` gesetzt. Dies geschieht auch, wenn der Maximal-/Minimalwert der Lautstärke schon erreicht wurde und ein weiterer Befehl gesendet wird.

MicVolume

Mit dieser Methode kann die Lautstärke des Mikrofons des Projektors geändert werden.

```
METHOD MicVolume :hresult
VAR_INPUT
    bIncrease   : BOOL;
    bExecute    : BOOL;
END_VAR
```

bIncrease: Ist dieser Parameter bei Methodenaufruf `TRUE`, wird ein Befehl zum Erhöhen der Lautstärke gesendet werden. Das Senden des Befehls mit dem Parameterwert `FALSE`, hat ein Verringern der Lautstärke zur Folge.

bExecute: Positive Flanke bewirkt bei Methodenaufruf das Senden eines Befehls.

Der Projektor sendet eine Bestätigung, wenn der Befehl erfolgreich umgesetzt werden kann. Durch die Bestätigung wird der Ausgangsparameter `stProjStatus.stMVOL.bSuccessful` für die Dauer eines Zyklus auf `TRUE` gesetzt. Dies geschieht auch, wenn der Maximal-/Minimalwert der Lautstärke schon erreicht wurde und ein weiterer Befehl gesendet wird.

3.2 Strukturen, Enumerationen

3.2.1 ST_PJLink_Projector_Status

```
TYPE ST_PJLink_Projector_Status
STRUCT
    stAVMT : ST_PJLink_AVMT_Status; //Struct with information about the AV-mute status.
    stCLSS : ST_PJLink_CLSS_Status; //Struct with information about the PJLink-class status.
    stERST : ST_PJLink_ERST_Status; //Struct with information about the error status.
    stINF1 : ST_PJLink_INF1_Status; //Struct with information about the manufacturer.
    stINF2 : ST_PJLink_INF2_Status; //Struct with information about the product.
```

```

stINFO : ST_PJLink_INFO_Status; //Struct with further information about the projector.
stINPT : ST_PJLink_INPT_Status; //Struct with information about the current media input.
stINST : ST_PJLink_INST_Status; //Struct with information about all available media inputs.
stLAMP : ST_PJLink_LAMP_Status; //Struct with information about the lamp status.
stNAME : ST_PJLink_NAME_Status; //Struct with information about the projector name.
stPOWR : ST_PJLink_POWR_Status; //Struct with information about the Power status.
stSNUM : ST_PJLink2_SNUM_Status; //Struct with information about the Serialnumber.
stSVER : ST_PJLink2_SVER_Status; //Struct with information about the Software Version.
stINNM : ST_PJLink2_INNM_Status; //Struct with information about the name of Input Terminal.
stIRES : ST_PJLink2_IRES_Status; //Struct with information about the Input Resolution.
stRRES : ST_PJLink2_RRES_Status; //Struct with information about the recommended resolution.
stFILT : ST_PJLink2_FILT_Status; //Struct with information about the Filter usage time.
stRLMP : ST_PJLink2_RLMP_Status; //Struct with information about the lamp part number.
stRFIL : ST_PJLink2_RFIL_Status; //Struct with information about the filter part number.
stSVOL : ST_PJLink2_SVOL_Status; //Struct with information about the Speaker Volume.
stMVOL : ST_PJLink2_MVOL_Status; //Struct with information about the Mic Volume.
stFREZ : ST_PJLink2_FREZ_Status; //Struct with information about the freeze status.
END_STRUCT
END_TYPE

```

Diese Struktur wird benutzt um alle Informationen, die vom Projektor erhalten werden, zu bündeln und dem Benutzer auszugeben (Ausgangsparameter stProjStatus).

3.2.2 ST_PJLink_AVMT_Status

```

TYPE ST_PJLink_AVMT_Status
STRUCT
    bSuccessful          : BOOL; (* TRUE for 1 Cycle if command is successful. *)
    bVideoMuted          : BOOL; (* Current state of Videomute. *)
    bAudioMuted          : BOOL; (* Current state of Audiomute. *)
    bErr_OutOfParameter  : BOOL; (* Error invalid parameter value. *)
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.3 ST_PJLink_CLSS_Status

```

TYPE ST_PJLink_CLSS_Status
STRUCT
    nClass          : BYTE; (* PJLink Class 1/2 *)
    bClassHigher2   : BOOL; (* PJLink Class higher than 2 *)
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure  : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.4 ST_PJLink_ERST_Errors

```

TYPE ST_PJLink_ERST_Errors
STRUCT
    bNoError : BOOL;
    bWarning : BOOL;
    bError   : BOOL;
END_STRUCT
END_TYPE

```

3.2.5 ST_PJLink_ERST_Status

```

TYPE ST_PJLink_ERST_Status
STRUCT
    stFan          : ST_PJLink_ERST_Errors;
    stLamp         : ST_PJLink_ERST_Errors;
    stTemperature  : ST_PJLink_ERST_Errors;
    stCover        : ST_PJLink_ERST_Errors;
    stFilter       : ST_PJLink_ERST_Errors;
    stOther        : ST_PJLink_ERST_Errors;
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure  : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.6 ST_PJLink_FILT_Status

```

TYPE ST_PJLink2_FILT_Status
STRUCT
    nFilterUsageTime      : UDINT;    (* Filter usage Time in Hours. *)
    bErr_NoFilter         : BOOL;     (* No Filter. *)
    bErr_UnavailableTime  : BOOL;     (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL;     (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.7 ST_PJLink_FREZ_Status

```

TYPE ST_PJLink2_FREZ_Status
STRUCT
    bSuccessful           : BOOL;     (* TRUE for 1 Cycle if command is successful. *)
    bFreezeStatus        : BOOL;     (* Freeze status 1--> Freeze ON 0 --> Freeze OFF. *)
    bErr_NotSupported    : BOOL;     (* Freeze is not supported by projector. *)
    bErr_OutOfParameter  : BOOL;     (* Invalid Parameter. *)
    bErr_UnavailableTime  : BOOL;     (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL;     (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.8 ST_PJLink_INF1_Status

```

TYPE ST_PJLink_INF1_Status
STRUCT
    sManufName           : String;    (* Manufacturer Name. *)
    bErr_UnavailableTime  : BOOL;     (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL;     (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.9 ST_PJLink_INF2_Status

```

TYPE ST_PJLink_INF2_Status
STRUCT
    sProdName           : STRING;    (* Product Name. *)
    bErr_UnavailableTime  : BOOL;     (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL;     (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.10 ST_PJLink_INFO_Status

```

TYPE ST_PJLink_INFO_Status
STRUCT
    sAdditionalInfo      : String;    (* Additional Info. *)
    bErr_UnavailableTime  : BOOL;     (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL;     (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.11 ST_PJLink_INNM_Status

```

TYPE ST_PJLink2_INNM_Status
STRUCT
    sInputTerminalName   : STRING(128); (* Name of Input Terminal. *)
    bErr_OutOfParameter  : BOOL;     (* Invalid Parameter. *)
    bErr_UnavailableTime  : BOOL;     (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL;     (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.12 ST_PJLink_INPT_Status

```

TYPE ST_PJLink_INPT_Status
STRUCT
    bSuccessful           : BOOL;     (* TRUE for 1 Cycle if command is successful. *)
    sVideo_Channel       : STRING(1); (* Number of current Video Channel. *)
    sDigital_Channel     : STRING(1); (* Number of current Digital Channel. *)

```

```

sStorage_Channel      : STRING(1); (* Number of current Storage Channel. *)
sNetwork_Channel      : STRING(1); (* Number of current Network Channel. *)
sInternal_Channel     : STRING(1); (* Number of current Internal Channel. *)
bErr_NonExistInptSrc  : BOOL;    (* Input does not Exist. *)
bErr_UnavailableTime  : BOOL;    (* Wrong time for projector to handle command. *)
bErr_ProjFailure      : BOOL;    (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.13 ST_PJLink_INST_Status

```

TYPE ST_PJLink_INST_Status
STRUCT
  nRGB_Channel          : BYTE; (* Number of available RGB Channels. *)
  nVideo_Channel        : BYTE; (* Number of available Video Channels. *)
  nDigital_Channel      : BYTE; (* Number of available Digital Channels. *)
  nStorage_Channel      : BYTE; (* Number of available Storage Channels. *)
  nNetwork_Channel      : BYTE; (* Number of available Network Channels. *)
  nInternal_Channel     : BYTE; (* Number of available Internal Channels. *)
  bErr_UnavailableTime  : BOOL; (* Wrong time for projector to handle command. *)
  bErr_ProjFailure      : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.14 ST_PJLink_IRES_Status

```

TYPE ST_PJLink2_IRES_Status
STRUCT
  nHorizontalRes        : UINT;  (* Current Horizontal Resolution. *)
  nVerticalRes          : UINT;  (* Current Vertical Resolution. *)
  bErr_UnavailableTime  : BOOL;  (* Wrong time for projector to handle command. *)
  bErr_ProjFailure      : BOOL;  (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.15 ST_PJLink_MVOL_Status

```

TYPE ST_PJLink2_MVOL_Status
STRUCT
  bSuccessful           : BOOL;  (* True for 1 Cycle if command is successful. *)
  bErr_MicNotInstalled  : BOOL;  (* No Mic Installed. *)
  bErr_OutOfParameter   : BOOL;  (* Invalid Parameter. *)
  bErr_UnavailableTime  : BOOL;  (* Wrong time for projector to handle command. *)
  bErr_ProjFailure      : BOOL;  (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.16 ST_PJLink_LAMP_Status

```

TYPE ST_PJLink_LAMP_Status
STRUCT
  astLamps              : ARRAY [0..GVL_PJLink.byMaxLamps-1] OF ST_PJLink_Lamp_int;
  bErr_UnavailableTime  : BOOL; //wrong time for projector to handle command
  bErr_ProjFailure      : BOOL; //Projector/Display failure
END_STRUCT
END_TYPE

```

3.2.17 ST_PJLink_Lamp_int

```

TYPE ST_PJLink_LAMP_int
STRUCT
  nLamp_LightingTime    : UDINT; //Hours of Lamp Lighting Time
  bLamp_On              : BOOL; //Lamp Status on/off
END_STRUCT
END_TYPE

```

3.2.18 ST_PJLink_NAME_Status

```

TYPE ST_PJLink_NAME_Status
STRUCT
  sProjName             : String; (* Projector name. *)

```

```

    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.19 ST_PJLink_POWR_Status

```

TYPE ST_PJLink_POWR_Status
STRUCT
    bSuccessful          : BOOL; (* TRUE for 1 Cycle if command is successful. *)
    bOn                  : BOOL; (* Projector is on. *)
    bOff                 : BOOL; (* Projector is off. *)
    bCooling             : BOOL; (* Projector is cooling. *)
    bWarmup              : BOOL; (* Projector is starting. *)
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.20 ST_PJLink_RFIL_Status

```

TYPE ST_PJLink2_RFIL_Status
STRUCT
    sFilterModelNumber  : STRING(128); (* Filter replacement model number. *)
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.21 ST_PJLink_RLMP_Status

```

TYPE ST_PJLink2_RLMP_Status
STRUCT
    sLampModelNumber    : STRING(128); (* Lamp Replacement model number. *)
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.22 ST_PJLink_RRES_Status

```

TYPE ST_PJLink2_RRES_Status
STRUCT
    nHorizontalRes      : UINT; (* Recommended Horizontal Resolution. *)
    nVerticalRes        : UINT; (* Recommended Vertical Resolution. *)
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.23 ST_PJLink_SNUM_Status

```

TYPE ST_PJLink2_SNUM_Status
STRUCT
    sSerialNumber       : STRING (32); (* Serial Number of Projector. *)
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.24 ST_PJLink_SVER_Status

```

TYPE ST_PJLink2_SVER_Status
STRUCT
    sSoftwareVersion    : STRING (32); (* Serial Number of ProjectorsProdName. *)
    bErr_UnavailableTime : BOOL; (* Wrong time for projector to handle command. *)
    bErr_ProjFailure     : BOOL; (* Projector/Display failure. *)
END_STRUCT
END_TYPE

```

3.2.25 ST_PJLink_SVOL_Status

```

TYPE ST_PJLink2_SVOL_Status
STRUCT
    bSuccessful          : BOOL;      (* True for 1 Cycle if command is successful. *)
    bErr_SpeakerNotInstalled : BOOL;  (* No Speaker Installed. *)
    bErr_OutOfParameter   : BOOL;    (* Invalid Parameter. *)
    bErr_UnavailableTime  : BOOL;    (* Wrong time for projector to handle command. *)
    bErr_ProjFailure      : BOOL;    (* Projector/Display failure. *)
END_STRUCT
END_TYPE
    
```

3.2.26 E_PJLink_InputType

```

TYPE E_PJLink_InputType :
(
    RGB,          (* RGB Input *)
    Video,       (* Video Input *)
    Digital,     (* Digital Input *)
    Storage,     (* Storage Input *)
    Network,     (* Network Input *)
    _Internal    (* Internal Input, only Class 2 *)
);
END_TYPE
    
```

3.2.27 E_PJLink_MuteType

```

TYPE E_PJLink_MuteType :
(
    Audio,       (* Audio Mute *)
    Video,      (* Video Mute *)
    AudioVideo  (* Audio/Video Mute *)
);
END_TYPE
    
```

3.2.28 E_PJLink_Steps

```

TYPE E_PJLink_Step :
(
    Init                , //init-step
    SocketConnect       , //Connect to Socket - init
    SocketConnecting    , //Connect to Socket - connecting
    ListenForStart      , //PJLink communication - Listen init
    ListeningForStart    , //PJLink communication - Listening
    PJLinkSendForAuthorization , //PJLink communication - Start Sending Authorization Message to
    Projector
    PJLinkSendingForAuthorization , //PJLink communication - Sending Authorization Message to
    Projector
    PJLinkReceiveForAuthorization , //PJLink communication - Listen for Authorization-Response of
    Projector
    PJLinkReceivingForAuthorization, //PJLink communication - Receiving Authorization-Response of
    Projector
    PJLinkConnectedIdle , //PJLink communication - Idle Mode (wait for commands to Send)
    PJLinkSend          , //PJLink communication - Send Command
    PJLinkSending       , //PJLink communication - Sending Command
    PJLinkReceive       , //PJLink communication - Listen for Response of Projector
    PJLinkReceiving     , //PJLink communication - Receiving Response of Projector
    PJLinkReceived      , //PJLink communication - Response Received --> Analyse Response
    SocketClose         , //Socket close - init
    SocketClosing       , //Socket close - closing
    SocketClosed        , //Socket close - Closed
    SocketConnectErr    , //Connect to Socket - Error
    SocketSendErr       , //PJLink communication - Send Error
    SocketCloseErr      , //Socket close - Error
);
END_TYPE
    
```

3.2.29 E_PJLink_UpdateType

```

TYPE E_PJLink_UpdateType :
(
    StatusUpdate , (* Update all status. *)
    Power        , (* Update Power status. *)
    Input        , (* Update Input status. *)
    Muting       , (* Update Muting status. *)
);
    
```

```

Lamp                , (* Update Lamp status. *)
ErrorStates         , (* Update ErrorStates. *)
InputList           , (* Update the InputList. *)
Name                , (* Update projector name. *)
Manufacturer        , (* Update manufacturer. *)
Product             , (* Update Product information. *)
Info                , (* Update further information. *)
ClassInformation    , (* Update PJLink-Class information. *)

//Class 2
SerialNumber        , (* Update Serialnumber. *)
SoftwareVersion     , (* Update Software version. *)
InputTerminal       , (* Update Name of Input Terminal. *)
InputResolution     , (* Update Input Resolution. *)
RecommendedResolution , (* Update recommended Resolution. *)
Filter              , (* Update Filter usage time. *)
LampPartNumber      , (* Update Lamp part number. *)
FilterPartNumber    , (* Update Filter part number. *)
Freeze              , (* Update Freeze status. *)
);
END_TYPE

```

3.2.30 PJLink Error Codes

PJLink Error Codes

Value (hex)	Value (dec)	Description
0x8006	32774	PJLink-Session-Timeout (30 secs in Idle Mode)
0x8007	32775	Timeout Statemachine
0x8008	32776	Authorization Error

4 Beispiel

Das https://infosys.beckhoff.com/content/1031/TF6310_pjlink/Resources/9947745163.zip beinhaltet ein SPS-Programm, um die Verwendung von FB_PJLink zu verdeutlichen. Einem Projektor können unterschiedliche Befehle gesendet werden, sobald die Kommunikation erfolgreich aufgebaut wurde.

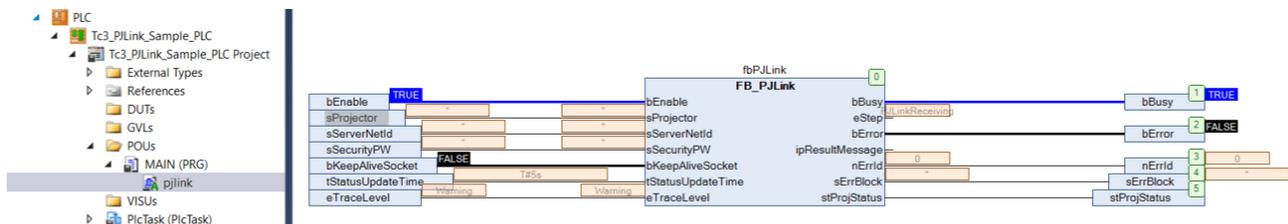


Schritte zur Inbetriebnahme:

- Der Eingangsparameter `sProjector` muss auf die IP-Adresse des anzusteuernenden Projektors angepasst werden.
- Der Eingangsparameter `sSecurityPW` muss mit dem PJLink Passwort in den Einstellungen des Projektors übereinstimmen. (Optional: Nur wenn die PJLink Security Authorization aktiviert ist.)
- Den Eingangsparameter `bEnable` auf `TRUE` setzen, sodass die Kommunikation zum Projektor aufgebaut wird.

Kommentare zum Programmcode:

- Aufruf von FB_PJLink in der `pjlink`-Methode im MAIN-Programm:



- Sobald die Verbindung zu einem Projektor erfolgreich hergestellt wurde, können im MAIN-Programm Methoden aufgerufen werden, um Kommandos an den Projektor zu senden:

```
pjlink();
//Please Write Variables
//switch Projector on/off by setting bPowerExecute and choose wanted Power State
IF bPowerExecute FALSE THEN
  fbPJLink.Power (bOn:= bWantedPowerState TRUE, bExecute:=bPowerExecute FALSE);
  bPowerExecute FALSE :=FALSE;
fbPJLink.Power (bOn:= bWantedPowerState TRUE, bExecute:=FALSE); // call second time for Reset Execute --> needed
END_IF
```

Beispiel für das Power-On-Kommando. Das Kommando wird durch Setzen von `bPowerExecute` getriggert.

- Sollte folgende Fehlermeldung geloggt werden, stimmt der Parameter `sSecurityPW` nicht mit dem eingestellten Sicherheits-Passwort des Projektors überein.

07.01.2021 10:54:29 822 ms | 'MAIN.fbPJLink': PJLink Authorization Failed (Wrong sSecurityPW)

Weitere Hinweise:

- Das Beispiel kann auch zusammen mit der PJLink Testsoftware („TEST4CNT“) getestet werden. Mit dieser wird ein Projektor simuliert. Die Software lässt sich im Downloadbereich unter <http://pjlink.jbma.or.jp/english/> herunterladen.
- Die TwinCAT Function TF6310 „TwinCAT TCP/IP“ wird für dieses Beispiel vorausgesetzt.



Der Download enthält ein TwinCAT-Projekt und die kompilierte Bibliothek „Tc3_PJLink“. Diese Bibliothek muss zunächst über das Bibliotheks-Repository installiert werden. (Siehe: [PLC > Bibliotheken verwenden](#))



Die bereits eingebundenen Bibliotheken „Tc2_Tcplp“, „Tc3_Module“ und „Tc2_Uutilities“ sind für die Verwendung der PJLink-Implementierung erforderlich.

Trademark statements

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Third-party trademark statements

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

Mehr Informationen:

www.beckhoff.com/entertainment-industrie

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

