BECKHOFF New Automation Technology

Manual | EN

TwinCAT 3

PLC Library: PJLink

2023-07-04 | Version: 1.3.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

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.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

PJLink is a communication protocol, based on TCP/IP communication, which is used to enable monitoring and control of projectors over the network. It was created by the JBMIA (Japan Business Machine and Information System Industries Association).

PJLink is a standard that is not limited to a projector manufacturer or a specific projector model. Thus, PJLink enables central control of projectors from different manufacturers. It includes basic commands and status queries that are important for control and monitoring.

The implementation of PJLink covers the command scope of PJLink classes 1 and 2 of the Control Protocol. The Security Authorization function is also supported. The "Search Protocol" and the "Status Notification Protocol" (both based on UDP communication and part of the Class 2 specification) are not included in the implementation.

Commands:

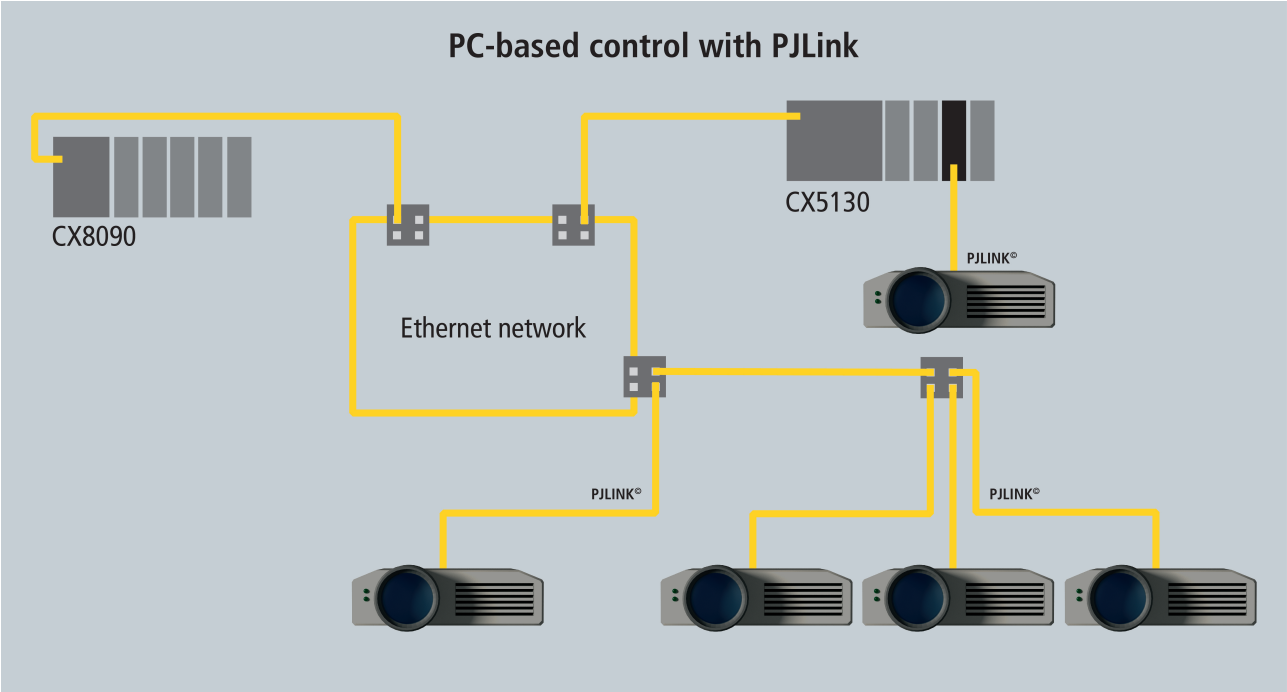
- Power control,
- input changeover,
- audio mute and blanking control.
- freeze control,*
- speaker volume control,*
- microphone volume control.*

Inquiries:

- Power state,
- available media input types,
- input changeover state,
- audio mute and blanking state,
- error state
- lamp operation hours,
- projector name, model name, vendor name, PJLink class,
- serial number, software version, name media input,*
- current resolution, recommended resolution,*
- filter operation hours, serial number filter, serial number lamp(s),*
- freeze state.*

**Commands and inquiries are only supported by PJLink class 2 projectors.*

Every PJLink-message will be responded by the addressed projector.



You can find further information on our website following:
PC-based Control for Stage- and Show Technology

The PJLink-specifications and information on PJLink are located at the website:
http://pjlink.jbmia.or.jp/english/.

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

Requirements:

Technical data	Request
TwinCAT version	TwinCAT 3.1 Build 4022.20 or higher
Visual Studio version	Visual Studio 2013 or higher
Required TwinCAT licenses	TF6310 license

3 Programming

3.1 FB_PJLink

This function block enables communication to a projector via TCP/IP and can be used for control and monitoring. Communication to a projector is established automatically by the function block. Any information received from the controller is evaluated and displayed in the structure `ST_PJLink_Projector_Status` (output parameter `stProjStatus`). In addition, there is an automatic status update function that is executed periodically. Information, warnings and errors are displayed using the TC3 Event Logger.

Requirements for `FB_PJLink`:

- TF6310 „TwinCAT TCP/IP“,
- Tc2_TcpIp (Communication) library to be linked in,
- TwinCAT v3.1.0 engineering environment or higher.

Important: Obey Projector Settings

Following settings for the projector need to be set up:

- Activate the PJLink function,
- deactivate the PJLink-Security function (optional),
- activate network functionality,
- deactivate the ECO function (disabling of network functionality in standby mode).

Sending of commands and inquiries to the projector is controlled by calling methods and setting corresponding input parameters:

- `Power` (change the power status of the projector (Switch on/off)),
- `Mute` (change mute status of the projector (Switch video, audio, video/audio mute on/off)),
- `Input` (change media input channel of the projector (RGB, video, digital, storage, network, internal)),
- `Update` (send specific status inquiries),
- `Freeze` (change the freeze status of the projector (activate/deactivate freeze)),
- `Volume` (change the volume of the projectors speakers),
- `MicVolume` (change the volume of the projectors microphone).

Licence

For using `FB_PJLink` a TC1200 “TC3 PLC” licence and a TF6310 “TwinCAT TCP/IP” licence are required.

VAR_INPUT

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

bEnable: Setting this parameter `TRUE` activates the function module. Communication to a projector is established. If `FALSE` is set, communication is terminated and the function block is deactivated.

sProjector: The local IP address (IPv4) of the TCP/IP client/server socket as a string (e.g. `'192.168.0.5'`). An empty string can also be specified for the default network adapter.

sServerNetID: String containing the AMS network address of the TwinCAT TCP/IP Connection Server. An empty string can also be specified for the local computer (default).

sSecurityPW: Password for the Security Authorization function (optional). The identical (case sensitive) password must be entered in the projector settings.

bKeepAliveSocket: When `TRUE`, activates a Keep Alive mode for communication with the projector. Only one socket is opened, which is kept permanently open. In the event of an error, the function block remains in an error mode. `FALSE` (recommended) opens a new socket before sending a command and then closes it again. In the event of an error, a new connection process is automatically triggered.

tStatusUpdateTime: Update interval of the status information request from the projector. (Default=5s; higher update time `tStatusUpdateTime` reduces data traffic.)

eTraceLevel: Determines the event types that are sent via the Event Logger. Events of greater or equal importance than the value at the input parameter are sent. (Example: Value = Warning → Sending warnings, errors and criticals; value=Warning → Among other things, PJLink messages are logged.)

VAR_OUTPUT

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

bBusy: This output is set as long as the connection to the socket is established.

eStep: This output parameter specifies the current step in the `FB_PJLink` State machine.

bError: If an error occurs, this output is set. (e.g.: connection to the socket is not possible).

ipResultMessage: This interface is used to send events via the TC3 Event Logger.

nErrID: This parameter returns the TwinCAT TCP/IP Connection Server error number when the `bError` output is set.

sErrBlock: Provides further information on the error that occurred. (Causing function block)

stProjStatus: Structure with the projector status information. (After sending a command, the response of the projector is evaluated and output in `stProjStatus`.)

Example: Command = `"%1POWR 1$R"` → Projector response = `"%1POWR=ERR3"` → `ProjStatus.POWR.Err_UnavailableTime` = `TRUE`.

3.1.1 Methods

Power

Use this method to send power commands to a projector.

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

bOn: If the method is called and this parameter is `TRUE`, a rising edge at `bExecute` will evoke sending a power-on command. If the method is called and this parameter is `FALSE`, a rising edge at `bExecute` will evoke sending a power-off command.

bExecute: Rising edge evokes sending a command at method call.

The projector sends a confirmation if the command can be successfully executed. The confirmation sets the output parameter `stProjStatus.stPOWR.bSuccessful` to `TRUE` for the duration of one cycle.

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: Select the required mute type (audio, video or audio/video).

bMuteOn: Setting this parameter to `TRUE` causes a mute switch-on command. If the parameter is `FALSE`, the command becomes a mute switch-off command.

bExecute: Rising edge evokes sending a command at method call.

The projector sends a confirmation if the command can be successfully executed. The confirmation sets the output parameter `stProjStatus.stAVMT.bSuccessful` to `TRUE` for the duration of one cycle.

Input

Use this method to send a media input switching command to a projector.

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

eInputType: Select the required input type (RGB, video, digital, storage, network, internal).

sInputChannel: Selection of the respective channel. (Value: 1-9, A-Z. See device manual.)

bExecute: Rising edge evokes sending a command at method call.

The projector sends a confirmation if the command can be successfully executed. The confirmation sets the output parameter `stProjStatus.stINPT.bSuccessful` to `TRUE` for the duration of one cycle..

Update

Use this method to send status requests to a projector.

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

eUpdateType: Select the required status request.

bExecute: Rising edge evokes sending of a command at method call.

Responses of the projector will be displayed in the output parameter `stProjStatus.`

Freeze

Use this method to send a projector the command to switch the freezing status.

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

bFreeze: If this parameter is `TRUE` when calling the method, the freeze function is activated on the projector. Sending the command with the parameter value `FALSE` deactivates the freeze function.

bExecute: Rising edge evokes sending a command at method call.

The projector sends a confirmation if the command can be successfully executed. The confirmation sets the output parameter `stProjStatus.stFREZ.bSuccessful` to `TRUE` for the duration of one cycle.

Volume

Use this method to change the volume of the projector's speakers.

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

bIncrease: If this parameter is `TRUE` when calling the method, a command to increase the volume will be sent. Sending the command with the parameter value `FALSE` will decrease the volume.

bExecute: Rising edge evokes sending a command at method call.

The projector sends a confirmation if the command can be successfully executed. The confirmation sets the output parameter `stProjStatus.stSVOL.bSuccessful` to `TRUE` for the duration of one cycle. This also happens if the maximum/minimum volume value has already been reached and another command is sent.

MicVolume

Use this method to change the volume of the projector's microphone.

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

bIncrease: If this parameter is `TRUE` when calling the method, a command to increase the volume will be sent. Sending the command with the parameter value `FALSE` will decrease the volume.

bExecute: Rising edge evokes sending a command at method call.

The projector sends a confirmation if the command can be successfully executed. The confirmation sets the output parameter `stProjStatus.stMVOL.bSuccessful` to `TRUE` for the duration of one cycle. This also happens if the maximum/minimum volume value has already been reached and another command is sent.

3.2 Structs, Enumerations

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
```

This structure is used to bundle all the information that are received from the projector (output parameter: `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 PJLinkErrorCodes

Table 1: 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 Example

The https://infosys.beckhoff.com/content/1033/TF6310_pjlink/Resources/9947745163/.zip contains a PLC program to illustrate the use of `FB_PJLink`. Different commands can be sent to a projector once communication has been successfully established.

Commissioning procedure:

- The input parameter `sProjector` must be adjusted to the IP address of the projector that is to be controlled.
- The input parameter `sSecurityPW` must match the PJLink password in the projector settings. (Optional: only when PJLink security authorization is enabled.)
- Set the input parameter `bEnable` to `TRUE` to establish communication with the projector.

Comments regarding the program code:

- Calling `FB_PJLink` in the `pjlink` method in the `MAIN` program:

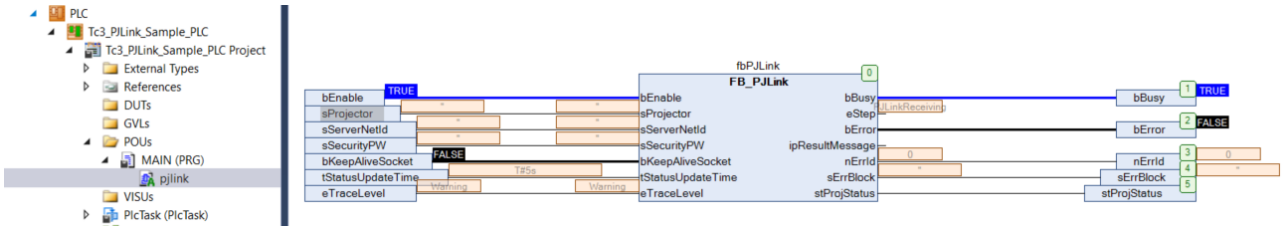


Fig. 1:

- Once the connection to a projector has been successfully established, methods can be called in the `MAIN` program to send commands to the projector:

```
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
```

Fig. 2: Example of the power-on command. The command is triggered by setting `bPowerExecute`.

- If the following error message is logged, the parameter `sSecurityPW` does not match the security password set for the projector.

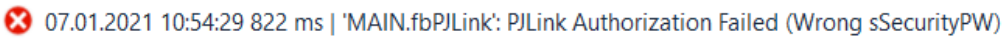


Fig. 3:

Additional notes:

- The example can also be tested in conjunction with the PJLink test software ("TEST4CNT"), which is used to simulate a projector. The software is available for download from http://pjlink.jbmia.or.jp/english/.
- The TwinCAT Function TF6310 "TwinCAT TCP/IP" is required for this example.

The download contains a TwinCAT project and the compiled library "Tc3_PJLink". This library must first be installed via the library repository. (See https://infosys.beckhoff.com/content/1033/te1000_xae_overview/html/te1000_xae_intro.htm(PLC > Libraries).)

The integrated libraries "Tc2_TcpIp", "Tc3_Module" and "Tc2_Utilities" are required for using the PJLink implementation.

More Information:
www.beckhoff.com/entertainment-industrie

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