

BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | PLC Library: Tc2_ProfinetDiag

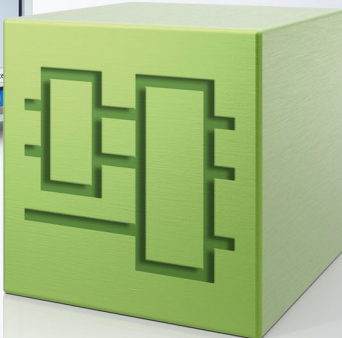
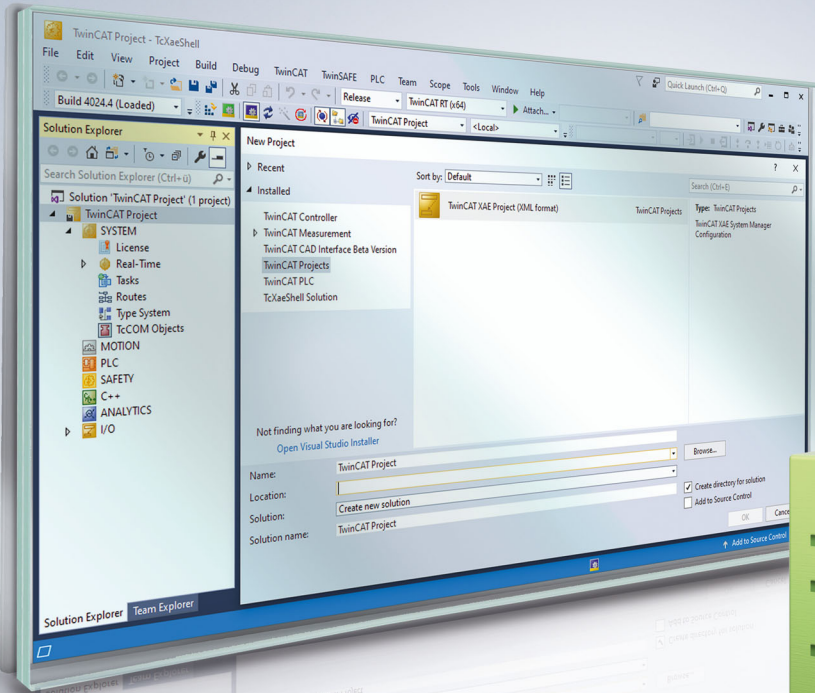


Table of contents

1 Foreword	5
1.1 Notes on the documentation	5
1.2 For your safety	5
1.3 Notes on information security.....	7
2 Overview	8
3 Function blocks	9
3.1 Controller.....	10
3.1.1 AlarmDiag	10
3.1.2 Identification and Maintenance	12
3.1.3 PROFINET RT Controller.....	20
3.1.4 FB_SET_PN_NAME	22
3.1.5 FB_RESET_PN_TO_FACTORY_SETTINGS	23
3.1.6 FB_PN_SCAN.....	24
3.1.7 FB_PN_SCAN_UpTo255.....	25
3.2 Device	26
3.2.1 EL6631-0010.....	26
3.2.2 Device via CCAT (CX-B930,TF6270)	29
3.2.3 FB_PN_SEND_ALARM	34
3.3 FB_PN_GET_PORT_STATISTIC.....	35
3.4 FB_PN_READ_PORT_DIAG.....	35
4 Data structures	37
4.1 Controller.....	37
4.1.1 IM	37
4.1.2 str_PN_Scan	38
4.1.3 AlarmDiag	39
4.1.4 ST_PN_DeviceInfo.....	40
4.2 Device	41
4.2.1 str_Diag_PN_Settings.....	41
4.2.2 str_luM_Data.....	41
4.3 EL6631-0010.....	41
4.3.1 E_PN_ALARM_TYP	41
4.4 IM	42
4.4.1 RecStruct	42
4.5 Port diagnosis	42
4.5.1 str_GetPortStatistic	42
4.5.2 str_PortDiag	42
5 Appendix	43
5.1 ADS Return Codes.....	43
5.2 Support and Service.....	48

1 Foreword

1.1 Notes on the documentation

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

For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.

The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all 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 notice.

No claims to modify 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 and licensed trademarks of Beckhoff Automation GmbH.

If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The distribution and reproduction of this document as well as the use and communication of its contents without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

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

The PLC library contains ready-made function blocks for use on the PROFINET controller and the device, which support handling, diagnosis and more. The names of the various function blocks may be similar, so the user must note that a distinction is made between "Controller" and "Device" before use.

System requirements

Target System	IPC or CX, (x86, x64, ARM)
Min. TwinCAT version	3.1.4018
Min. TwinCAT level	TC1200 TC3 PLC

3 Function blocks

Controller

Controller

Function blocks Controller	Meaning	Description
FB_SET_PN_NAME [▶ 22]	Assigns a name to the respective PROFINET device	TF6271 and EL663x
FB_RESET_PN_TO_FACTORY_SETTINGS [▶ 23]	Sets the respective PROFINET device to vendor settings	TF6271 and EL663x
FB_PN_SCAN [▶ 24]	Scans the PROFINET network and returns the number and a list of the PROFINET devices found	TF6271 and EL663x

AlarmDiag

Function blocks AlarmDiag	Meaning	Description
FB_PN_ALARM_DIAG [▶ 10]	Reading the diagnosis alarms	TF6271 and EL663x

I&M (Identification & Maintenance)

Function blocks I&M functions	Meaning	Description
FB_PN_IM0_READ [▶ 12]	Reading the I&M function 0	TF6271 and EL663x
FB_PN_IM1_READ [▶ 13]	Reading the I&M function 1	TF6271 and EL663x
FB_PN_IM2_READ [▶ 15]	Reading the I&M function 2	TF6271 and EL663x
FB_PN_IM3_READ [▶ 17]	Reading the I&M function 3	TF6271 and EL663x
FB_PN_IM4_READ [▶ 19]	Reading the I&M function 4	TF6271 and EL663x
FB_PN_IM1_WRITE [▶ 14]	Writing the I&M function 1	TF6271 and EL663x
FB_PN_IM2_WRITE [▶ 16]	Writing the I&M function 2	TF6271 and EL663x
FB_PN_IM3_WRITE [▶ 18]	Writing the I&M function 3	TF6271 and EL663x
FB_PN_IM4_WRITE [▶ 19]	Writing the I&M function 4	TF6271 and EL663x

PROFINET RT Controller

Function blocks PROFINET RT Controller	Meaning	Description
FB_PN_ReadStateOfDevices [▶ 20]	Reading the number of configured, missing devices and those with diagnosis information	TF6271, CCAT-M930 and EL663x
FB_PN_ReadCompleteInfoOfDevices [▶ 21]	Creates a list with the complete information of the configured devices	TF6271, CCAT-M930 and EL663x

Device

Device

Function blocks Device	Meaning	Description
FB_PN_SEND_ALARM [▶ 34]	Sends an alarm to the controller	TF6270

Device via CCAT (CX-B930, TF6270)

Function blocks Device via CCAT	Meaning	Description
FB_PROFINET_READ_IM [▶ 29]	Returns the I&M data of the respective device	TF6270 and CCAT-B930
FB_PROFINET_READ_NAME [▶ 30]	Returns the PROFIENT name of the respective PROFINET device	TF6270 and CCAT-B930
FB_PROFINET_READ_PRM [▶ 31]	Scans the PROFINET network and returns the number and a list of the PROFINET devices found	TF6270 and CCAT-B930
FB_PROFINET_WRITE_IM [▶ 32]	Enables the I&M data of the respective PROFINET device to be written	TF6270 and CCAT-B930
FB_PROFINET_SET_NAME [▶ 33]	Enables the PROFINET name of the respective PROFINET device to be changed	TF6270 and CCAT-B930

EL6631-0010

Function blocks EL6631-0010	Meaning	Description
FB_READ_PROFINET_NAME [▶ 26]	Reads the PROFINET name of an EL6631-0010 and, if configured, of a virtual EL6631-0010	EL6631-0010
FB_Write_IuM_EL6631_0010 [▶ 27]	Writes I&M1, I&M2, I&M3 and I&M4 data to the PROFINET device.	EL6631-0010
FB_Read_IuM_EL6631_0010 [▶ 28]	Reads I&M1, I&M2, I&M3 and I&M4 data from the PROFINET device.	EL6631-0010

Port diagnosis

Statistics and diagnosis information

Function blocks Statistics and diagnosis information	Meaning	Description
FB_PN_GET_PORT_STATISTIC [▶ 35]	Reading the port statistics	TF6271 and EL663x
FB_PN_READ_PORT_DIAG [▶ 35]	Reading the port diagnosis	TF6271 and EL663x

3.1 Controller

3.1.1 AlarmDiag

3.1.1.1 FB_PN_ALARM_DIAG



The function block FB_PN_ALARM_DIAG can be used to read out diagnosis alarms. Each instance of this function block provides a PLC input ("PnIoBoxDiag"). This input must be linked to the "PnIoBoxDiag" input of the device to be evaluated. Once the diagnosis alarms/warnings have been successfully read, the alarm state of the device is reset. The function block must be called once for each PROFINET device. A run index (iNrAlarms) indicates how many diagnosis alarms have been read from the buffer.

 **Inputs**

```
VAR_INPUT
  bEnable : BOOL;
  NETID   : T_AmsNetId;
  PORT    : T_AmsPort;
END_VAR
```

Name	Type	Description
bEnable	BOOL	Enabling the function block.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

 **Outputs**

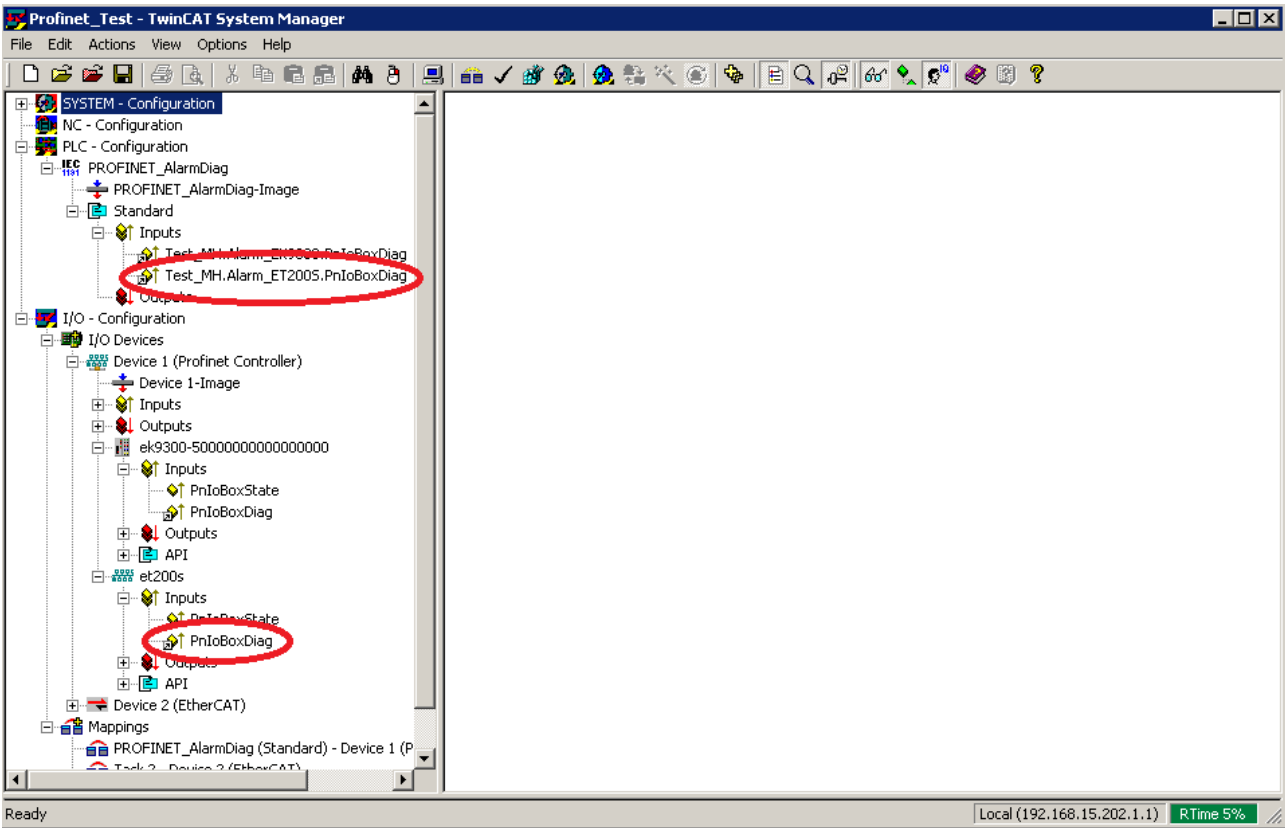
```
VAR_OUTPUT
  bBusy           : BOOL;
  stAlarmDiagData : ST_PN_AlarmDiagData;
  bError          : BOOL;
  iErrorID        : UDINT;
  iNrAlarms       : INT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
stAlarmDiagData	ST_PN_AlarmDiagData [▶ 40]	Diagnostic messages are output via this structure. An alarm is output via the structure in each cycle as long as the status bit [0x0010 = at least one AlarmCR got a diagnosis alarm] is present at the PLC input.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an ADS error number [▶ 43] if <i>bError</i> output is set.
iNrAlarms	INT	Number of alarms last read.

VAR

```
VAR
  PnIoBoxDiag AT %I* : WORD; (*Hardware Input*)
END_VAR
```

Name	Type	Description
PnIoBoxDiag	WORD	Hardware input. This variable must be linked to the PROFINET device. A change in the state of this variable informs the PLC program that new alarm diagnoses are pending in the linked PROFINET device.



Requirements

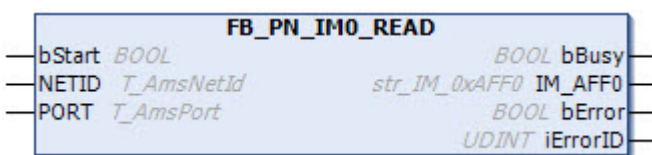
Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2 Identification and Maintenance

I&M (Identification & Maintenance)

Function blocks I&M functions	Meaning	Description
FB_PN_IM0_READ [► 12]	Reading the I&M function 0	TF6271 and EL663x
FB_PN_IM1_READ [► 13]	Reading the I&M function 1	TF6271 and EL663x
FB_PN_IM2_READ [► 15]	Reading the I&M function 2	TF6271 and EL663x
FB_PN_IM3_READ [► 17]	Reading the I&M function 3	TF6271 and EL663x
FB_PN_IM4_READ [► 19]	Reading the I&M function 4	TF6271 and EL663x
FB_PN_IM1_WRITE [► 14]	Writing the I&M function 1	TF6271 and EL663x
FB_PN_IM2_WRITE [► 16]	Writing the I&M function 2	TF6271 and EL663x
FB_PN_IM3_WRITE [► 18]	Writing the I&M function 3	TF6271 and EL663x
FB_PN_IM4_WRITE [► 19]	Writing the I&M function 4	TF6271 and EL663x

3.1.2.1 FB_PN_IM0_READ



The PROFINET controller uses the function block FB_PN_Im0_READ to read all I&M 0 (Identification & Maintenance) data from a device referenced via the input *Port*.
The frame structure of the I&M0 function corresponds to the index 0xAFF0 according to the PROFINET standard.

Inputs

```
VAR_INPUT
  bStart : BOOL;
  NETID : T_AmsNetId;
  PORT : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

Outputs

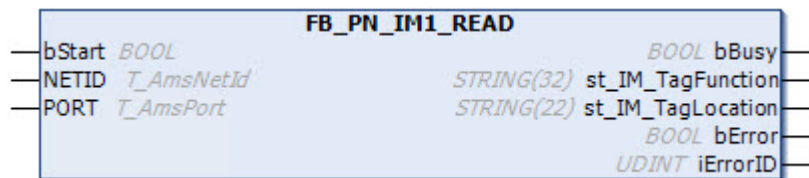
```
VAR_OUTPUT
  bBusy : BOOL;
  IM_AFF0 : str_IM_0xAFF0;
  bError : BOOL;
  iErrorID : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
IM_AFF0	str_IM_0xAFF0 [▶ 37]	IM_AFF0: Output of the I&M0 frame supplied by the device in a structure.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an ADS error number [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2.2 FB_PN_IM1_READ



The PROFINET controller uses the function block FB_PN_IM1_READ to read all I&M1 (Information & Maintenance) data from a device referenced via the *Port* input.
The frame structure of the I&M1 function corresponds to the index 0xAFF1 according to the PROFINET standard.

Inputs

```
VAR_INPUT
  bStart   : BOOL;
  NETID    : T_AmsNetId;
  PORT     : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

Outputs

```
VAR_OUTPUT
  bBusy           : BOOL;
  st_IM_TagFunction : STRING(32);
  st_IM_TagLocation : STRING(22);
  bError          : BOOL;
  iErrorID        : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
st_IM_TagFunction	STRING	Label read for function of the device.
st_IM_TagLocation	STRING	Label read for the installation location of the device.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [► 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2.3 FB_PN_IM1_WRITE



The PROFINET controller uses the function block FB_PN_IM1_WRITE to write all I&M1 (Identification & Maintenance) data to a device referenced via the input *Port*.

The frame structure of the I&M1 function corresponds to the index 0xAFF1 according to the PROFINET standard.

Inputs

```
VAR_INPUT
  bStart           : BOOL;
  NETID            : T_AmsNetId;
  PORT             : T_AmsPort;
  st_IM_TagFunction : STRING(32);
  st_IM_TagLocation : STRING(22);
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
st_IM_TagFunction	STRING	With this string the functional description is saved to the device.
st_IM_TagLocation	STRING	With this string the installation location is saved to the device.

🔌 Outputs

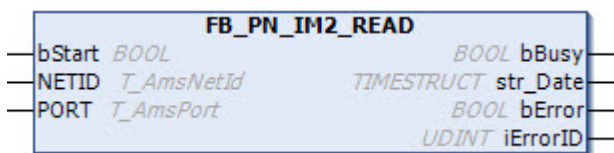
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorID   : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2.4 FB_PN_IM2_READ



The PROFINET controller uses the function block FB_PN_IM2_READ to read all I&M 2 (Identification & Maintenance) data from a device referenced via the input *Port*. The frame structure of the I&M2 function corresponds to the index 0xAFF2 according to the PROFINET standard.

🔌 Inputs

```
VAR_INPUT
  bStart : BOOL;
  NETID  : T_AmsNetId;
  PORT   : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

🔌 Outputs

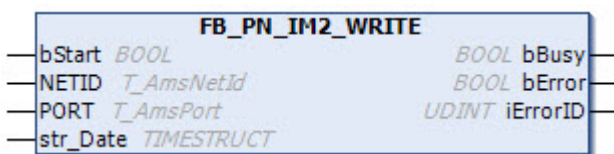
```
VAR_OUTPUT
  bBusy          : BOOL;
  str_Date       : TIMESTRUCT; (*YYYY-MM-DD HH:MM*);
  bError         : BOOL;
  iErrorID       : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
str_Date	TIMESTRUCT	Returns the device installation date in the format < YYYY-MM-DD HH:MM >.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2.5 FB_PN_IM2_WRITE



The PROFINET controller uses the function block FB_PN_IM2_WRITE to write all I&M 2 (Identification & Maintenance) data to a device referenced via the input *Port*.
 The frame structure of the I&M2 function corresponds to the index 0xAFF2 according to the PROFINET standard.

🔌 Inputs

```
VAR_INPUT
  bStart : BOOL;
  NETID  : T_AmsNetId;
  PORT   : T_AmsPort;
  str_Date : TIMESTRUCT; (*YYYY-MM-DD HH:MM*)
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
str_Date	TIMESTRUCT	Writes a date (e.g. device installation date) in the format < YYYY-MM-DD HH:MM > on the device.

🔌 Outputs

```
VAR_OUTPUT
  bBusy          : BOOL;
  bError         : BOOL;
  iErrorID       : UDINT;
END_VAR
```


Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [► 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2.6 FB_PN_IM3_READ



The PROFINET controller uses the function block FB_PN_IM3_READ to read all I&M3 (Identification & Maintenance) data from a device referenced via the input *Port*. The frame structure of the I&M3 function corresponds to the index 0xAFF3 according to the PROFINET standard.

Inputs

```
VAR_INPUT
  bStart : BOOL;
  NETID  : T_AmsNetId;
  PORT   : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

Outputs

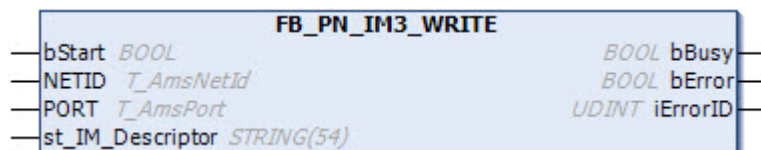
```
VAR_OUTPUT
  bBusy           : BOOL;
  st_IM_Descriptor : STRING(54);
  bError          : BOOL;
  iErrorID        : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
st_IM_Descriptor	STRING	Returns the vendor description stored for the device.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [► 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2.7 FB_PN_IM3_WRITE



The PROFINET controller uses the function block FB_PN_IM3_WRITE to write all I&M3 (Identification & Maintenance) data to a device referenced via the input *Port*.

The frame structure of the I&M3 function corresponds to the index 0xAFF3 according to the PROFINET standard.

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  NETID       : T_AmsNetId;
  PORT        : T_AmsPort;
  st_IM_Descriptor : STRING(54);
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
str_Date	TIMESTRUCT	Writes a date (e.g. device installation date) in the format < YYYY-MM-DD HH:MM > on the device.

 Outputs

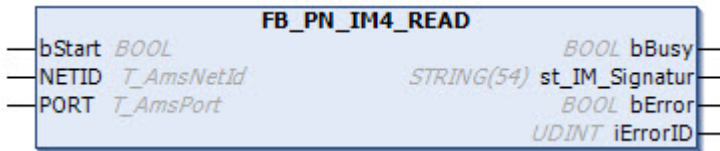
```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  iErrorID    : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2.8 FB_PN_IM4_Read



The PROFINET controller uses the function block FB_PN_IM4_READ to read all I&M4 (Identification & Maintenance) data from a device referenced via the input *Port*. The frame structure of the I&M4 function corresponds to the index 0xAFF4 according to the PROFINET standard.

Inputs

```
VAR_INPUT
  bStart : BOOL;
  NETID  : T_AmsNetId;
  PORT   : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

Outputs

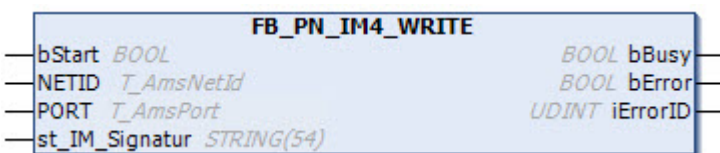
```
VAR_OUTPUT
  bBusy      : BOOL;
  st_IM_Signatur : STRING(54);
  bError     : BOOL;
  iErrorID   : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
st_IM_Signatur	STRING	Returns the vendor signature stored for the device.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.2.9 FB_PN_IM4_WRITE



The PROFINET controller uses the function block FB_PN_IM4_WRITE to write all I&M4 (Identification & Maintenance) data to a device referenced via *Port*.

The frame structure of the I&M4 function corresponds to the index 0xAFF4 according to the PROFINET standard.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  NETID      : T_AmsNetId;
  PORT       : T_AmsPort;
  st_IM_Signatur : STRING(54);
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
st_IM_Signatur	STRING	Signature of the vendor that is written to the device.

Outputs

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorID   : UDINT;
END_VAR
```

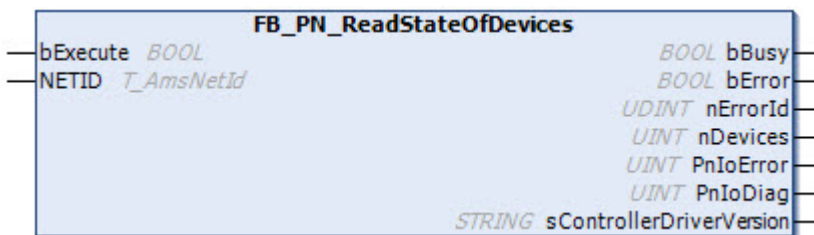
Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.3 PROFINET RT Controller

3.1.3.1 FB_PN_ReadStateOfDevices



When called, the function block FB_PN_ReadDtateOfDevices returns the number of configured and missing devices and those with diagnosis information.

PROFINET RT Controller driver version



The function block is only available for PROFINET RT Controller driver version v03(v0.21) or higher.
In TwinCAT XAE, the driver version can also be found in the PROFINET RT Controller I/O device.

Inputs

```
VAR_INPUT
  bExecute          : BOOL;
  NETID             : T_AmsNetIdArr;
END_VAR
```

Name	Type	Description
bExecute	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the PROFINET RT controller

Outputs

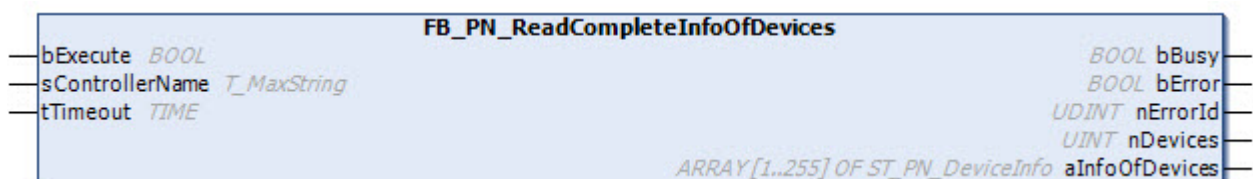
```
VAR_OUTPUT
  bBusy            : BOOL;
  bError           : BOOL;
  nErrorID        : UDINT;
  nDevices         : UINT;
  PnIoError       : UINT;
  PnIoDiag        : UINT;
  sControllerDriverVersion : STRING;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
nErrorId	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.
nDevices	UINT	Number of configured devices.(max.255)
PnIoError	UINT	Number of devices with error state or diagnosis.
PnIoDiag	UINT	Number of devices with diagnosis.
sControllerDriverVersion	UINT	PROFINET controller version 03 (V00.21) or higher required.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.54	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag (>= v1.4.1.0)

3.1.3.2 FB_PN_ReadCompleteInfoOfDevices



The function block FB_PN_ReadCompleteInfoOfDevices creates a list with the complete information of the configured devices (in TwinCAT). For PROFINET RT controllers such as TF6271 or CCAT M930 interface or EL6631 v11(v.024).

Inputs

```
VAR_INPUT
  bExecute           : BOOL;
  sControllerName   : T_Maxstring
  tTimeout          : TIME;
END_VAR
```

Name	Type	Description
bExecute	BOOL	The function block is enabled by a positive edge at this input.
sControllerName	T_MaxString	Name of the PROFINET RT controller in the TwinCAT tree.
tTimeout	TIME	Timeout for the complete read process

Outputs

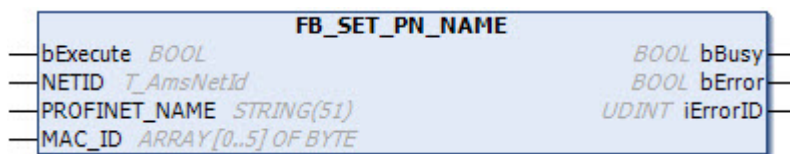
```
VAR_OUTPUT
  bBusy             : BOOL;
  bError            : BOOL;
  nErrorID          : UDINT;
  nDevices          : UINT;
  aInfoOfDevices    : ARRAY [1..255] OF st_PN_DeviceInfo;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
nErrorId	UDINT	Returns an <u>ADS error number</u> [► 43] if <i>bError</i> output is set.
nDevices	UINT	Number of PROFINET devices in the configuration.
aInfoOfDevices	ST_PN_DeviceInfo	Settings information of the configured PROFINET devices.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.54	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag (>= v1.4.1.0)

3.1.4 FB_SET_PN_NAME



The function block FB_SET_PN_NAME assigns a name to the respective PROFINET device. When assigning, ensure that only PROFINET-compliant characters are used.

Inputs

```
VAR_INPUT
  bExecute           : BOOL;
  NETID             : T_AmsNetId;
  PROFINET_NAME     : STRING(51);
  MAC_ID            : ARRAY [0..5] OF BYTE;
END_VAR
```

Name	Type	Description
bExecute	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PROFINET_NAME	STRING	Name to be assigned to the PROFINET device. Max. 240 characters and the following characters permitted "a..z", "0..9", ".", "-"
MAC_ID	BYTE	MAC ID of the controller

🔌 Outputs

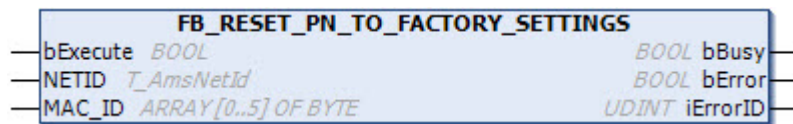
```
VAR_OUTPUT
  bBusy          : BOOL;
  bError         : BOOL;
  iErrorID      : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> ▶ 43 if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.5 FB_RESET_PN_TO_FACTORY_SETTINGS



The function block FB_RESET_PN_TO_FACTORY resets the respective PROFINET device to its vendor settings.

🔌 Inputs

```
VAR_INPUT
  bExecute      : BOOL;
  NETID        : T_AmsNetId;
  MAC_ID       : ARRAY [0..5] OF BYTE;
END_VAR
```

Name	Type	Description
bExecute	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
MAC_ID	BYTE	MAC ID of the controller

🔌 Outputs

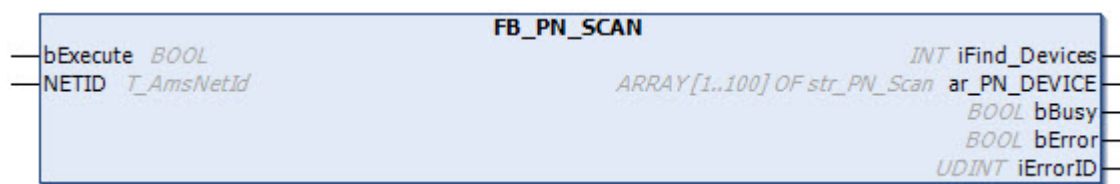
```
VAR_OUTPUT
  bBusy          : BOOL;
  bError         : BOOL;
  iErrorID      : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [► 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.6 FB_PN_SCAN



The function block FB_PN_SCAN scans the PROFINET network and returns the number and a list with information on the PROFINET devices found.

Inputs

```
VAR_INPUT
    bExecute      : BOOL;
    NETID        : T_AmsNetId;
END_VAR
```

Name	Type	Description
bExecute	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller

Outputs

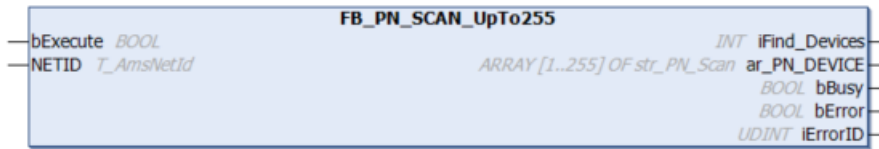
```
VAR_OUTPUT
    iFind_Devices : INT;
    ar_PN_DEVICE  : ARRAY [1..100] OF str_PN_SCAN;
    bBusy         : BOOL;
    bError        : BOOL;
    iErrorID      : UDINT;
END_VAR
```

Name	Type	Description
iFind_Devices	int	Number of PROFINET devices in the configuration.
ar_PN_DEVICE	str_PN_SCAN [► 38]	PROFINET/IP settings of the PROFINET devices.
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [► 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.1.7 FB_PN_SCAN_UpTo255



The FB_PN_SCAN_UpTo255 function block scans the PROFINET network and returns the number of and a list with information on the PROFINET devices found.

Inputs

```
VAR_INPUT
  bExecute      : BOOL;
  NETID         : T_AmsNetId;
END_VAR
```

Name	Type	Description
bExecute	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller

Outputs

```
VAR_OUTPUT
  iFind_Devices : INT;
  ar_PN_DEVICE  : ARRAY [1..255] OF str_PN_SCAN;
  bBusy         : BOOL;
  bError        : BOOL;
  iErrorID      : UDINT;
END_VAR
```

Name	Type	Description
iFind_Devices	int	Number of PROFINET devices in the configuration.
ar_PN_DEVICE	str_PN_SCAN [▶ 38]	PROFINET/IP settings of the PROFINET devices.
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the bBusy output is reset.
iErrorID	UDINT	Returns an ADS error number [▶ 43] if bError output is set.

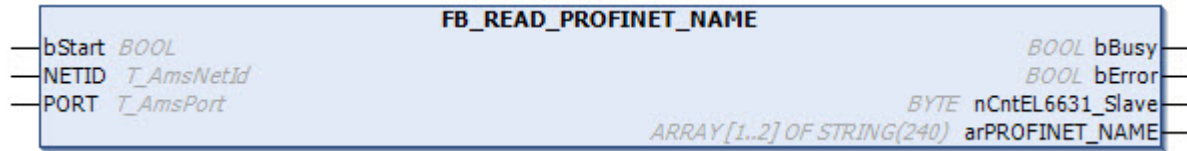
Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.57	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag (>= v1.5.2.0)

3.2 Device

3.2.1 EL6631-0010

3.2.1.1 FB_READ_PROFINET_NAME



The function block FB_READ_PROFINET_NAME reads the PROFINET name of an EL6631-0010 and, if configured, of a virtual EL6631-0010.

Inputs

```
VAR_INPUT
  bStart   : BOOL;
  NETID    : T_AmsNetId;
  PORT     : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

Outputs

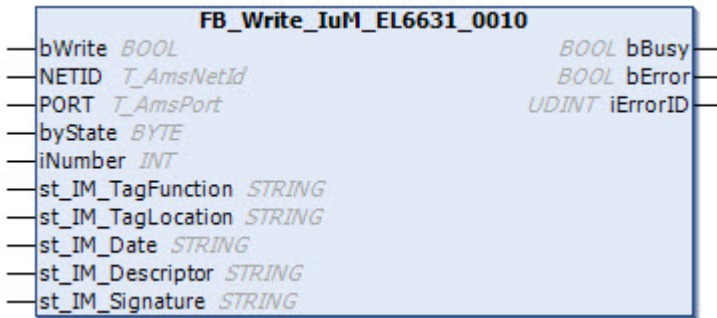
```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nCntEL6631_Slave : BYTE;
  arPROFINET_NAME : ARRAY [1..2] OF STRING(240);
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
nCntEL6631_Slave	BYTE	Provides information on how the EL6631-0010 is displayed. "0" = EL6631-0010 and "1" = virtual EL6631-0010.
arPROFINET_NAME	STRING	Contains the PROFINET names of the EL6631-0010 and, if configured, the virtual EL6631-0010.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.2.1.2 FB_Write_IuM_EL6631_0010



The function block FB_Write_IuM_EL6631_0010 writes I&M1, I&M2, I&M3 and I&M4 (Identification & Maintenance) data to the PROFINET device as a string via EtherCAT in accordance with the PROFINET specification.

Inputs

```

VAR_INPUT
  bWrite      : BOOL;
  NETID      : T_AmsNetId;
  PORT       : T_AmsPort;
  byState    : BYTE;
  iNumber    : INT:=0;
  st_IM_TagFunction : STRING;
  st_IM_TagLocation : STRING;
  st_IM_Date : STRING;
  st_IM_Descriptor : STRING;
  st_IM_Signature : STRING;
END_VAR
    
```

Name	Type	Description
bWrite	BOOL	A positive edge at this input enables the function block and the I&M data is written to the selected Profinet device.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T-AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
byState	BYTE	This byte can be used to select which I&M data is to be written. Bit 0 -> I&M1 Bit 1 -> I&M2 Bit 2 -> I&M3 Bit 3 -> I&M4
iNumber	INT	Two PROFINET devices can be mapped with one terminal. iNumber ("0" or "1") is used to select the device for which the I&M data is to be written.
st_IM_TagFunction	STRING	Label for the function of the device is written to the device. I&M1 byState.0=TRUE
st_IM_Taglocation	STRING	Label for the installation location of the device is written to the device. I&M1 byState.0=TRUE
st_IM_Date	STRING	Date of device installation is written to the device. I&M2 byState.1=TRUE
st_IM_Descriptor	STRING	Description of the vendor is written to the device. I&M3 byState.2=TRUE
st_IM_Signature	STRING	The vendor signature is written to the device. I&M4 byState.3=TRUE

🔌 Outputs

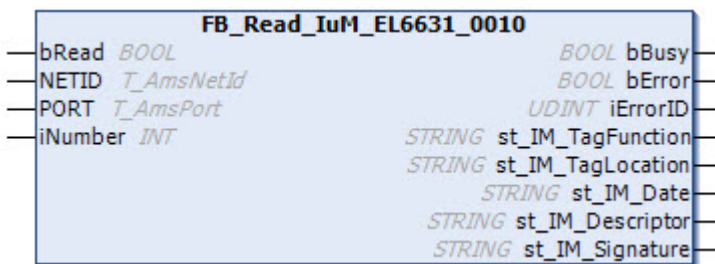
```
VAR_OUTPUT
  bBusy          : BOOL;
  bError         : BOOL;
  iErrorID       : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an ADS error number [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.2.1.3 FB_Read_IuM_EL6631_0010



The function block **FB_Read_IuM_EL6631_0010** reads I&M1, I&M2, I&M3 and I&M4 (Identification & Maintenance) data from a PROFINET device via EtherCAT as a string. The I&M0 data is read for a PROFINET device with EtherCAT via CoE (CAN over EtherCAT).

🔌 Inputs

```
VAR_INPUT
  bRead          : BOOL;;
  NETID          : T_AmsNetId;
  PORT           : T_AmsPort;
  iNumber        : INT:=0;
END_VAR
```

Name	Type	Description
bRead	BOOL	A positive edge at this input enables the function block and the I&M data is read from the PROFINET device.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T-AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
iNumber	INT	Two PROFINET devices can be mapped with one terminal. iNumber ("0" or "1") is used to select the device for which the I&M data is to be read.

🔌 Outputs

```
VAR_OUTPUT
  bBusy          : BOOL;
  bError         : BOOL;
  iErrorID       : UDINT;
  st_IM_TagFunction : STRING; (* I&M1 *)
```

```

st_IM_TagLocation      : STRING; (* I&M1 *)
st_IM_Date            : STRING; (* I&M2 *)
st_IM_Descriptor      : STRING; (* I&M3 *)
st_IM_Signature       : STRING; (* I&M4 *)
END_VAR

```

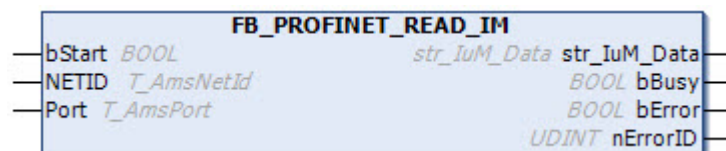
Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an ADS error number [► 43] if <i>bError</i> output is set.
st_IM_TagFunction	STRING	Label read for function of the device. I&M1
st_IM_Taglocation	STRING	Label read for the installation location of the device. I&M1
st_IM_Date	STRING	Returns the device installation date in the format. I&M2
st_IM_Descriptor	STRING	Returns the vendor description stored for the device. I&M3
st_IM_Signature	STRING	Returns the vendor signature stored for the device. I&M4

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.2.2 Device via CCAT (CX-B930,TF6270)

3.2.2.1 FB_PROFINET_READ_IM



The function block FB_PROFINET_READ_IM returns the I&M (Identification & Maintenance) data of the respective PROFINET device.

Inputs

```

VAR_INPUT
  bStart : BOOL;
  NETID  : T_AmsNetId;
  PORT   : T_AmsPort;
END_VAR

```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

🔌 Outputs

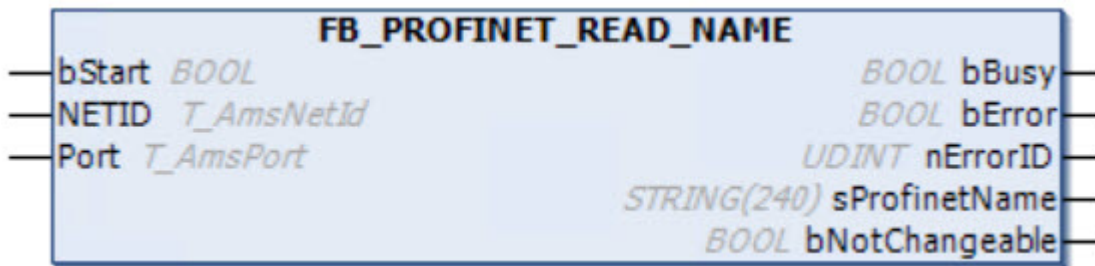
```
VAR_OUTPUT
  str_IuM_Data      : str_IuM_Data
  bBusy            : BOOL;
  bError           : BOOL;
  nErrorID        : UDINT;
END_VAR
```

Name	Type	Description
str_IuM_Data	str_IuM_Data [▶ 41]	Structural element with I&M data
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
nErrorId	UDINT	Returns an ADS error number [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.2.2.2 FB_PROFINET_READ_NAME



The function block FB_PROFINET_READ_NAME returns the PROFINET name of the respective PROFINET device and the information whether the PROFINET name can be changed by the PROFINET controller.

🔌 Inputs

```
VAR_INPUT
  bStart : BOOL;
  NETID  : T_AmsNetId;
  PORT   : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS NET ID of the PROFINET device
PORT	T_AmsPort	ADS port number of the PROFINET device; default = 0xFFFF

🔌 Outputs

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorID   : UDINT;
  sProfinetName : STRING(240);
  bNotChangeable : BOOL;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
nErrorId	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.
sProfinetName	STRING(240)	PROFINET device name. Max. 240 characters and the following characters permitted "a..z", "0..9", ".", "-"
bNotChangeable	BOOL	If "TRUE", the PROFINET controller cannot change the PROFINET name of the device.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag (>= v1.5.1.0)

3.2.2.3 FB_PROFINET_READ_PRM



The function block FB_PROFINET_READ_PRM scans the PROFINET network and returns the number and a list with information on the PROFINET devices found.

Inputs

```

VAR_INPUT
  NETID      : T_AmsNetId;
  PORT      : T_AmsPort;
  bStart    : BOOL;
END_VAR
    
```

Name	Type	Description
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T-AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
bStart	BOOL	The function block is enabled by a positive edge at this input.

Outputs

```

VAR_OUTPUT
  str_Diag_PN_Settings : str_Diag_PN_Settings;
  bBusy                : BOOL;
  bError               : BOOL;
  nErrorID             : UDINT;
END_VAR
    
```

Name	Type	Description
str_Diag_PN_Settings	str_Diag_PN_Settings [▶ 41]	PROFINET/IP settings
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
nErrorId	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.2.2.4 FB_PROFINET_WRITE_IM



The function block **FB_PROFINET_WRITE_IM** can be used to write the I&M (Identification & Maintenance) data of the respective PROFINET device.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  NETID       : T_AmsNetId;
  PORT        : T_AmsPort;
  str_IuM_Data : str_IuM_Data
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
str_IuM_Data	str_IuM_Data [▶ 41]	Structural element with I&M data

Outputs

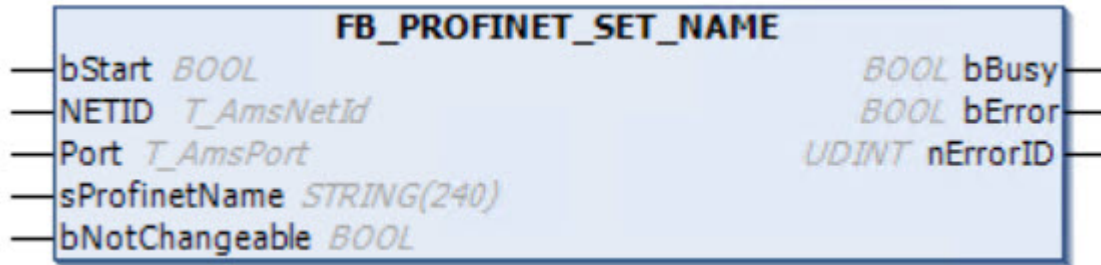
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorID   : UDINT;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
nErrorId	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.2.2.5 FB_PROFINET_SET_NAME



The function block FB_PROFINET_SET_NAME can be used to change the PROFINET name of the respective PROFINET device. PROFINET driver version 06 (V00.34) or higher, TF6270, CCAT PN Interface(B930) required

Inputs

```

VAR_INPUT
  bStart      : BOOL;
  NETID       : T_AmsNetId;
  PORT        : T_AmsPort;
  sProfinetName : STRING(240);
  bNotChangeable : BOOL;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS NET ID of the PROFINET device
PORT	T_AmsPort	ADS port number of the PROFINET device; default = 0xFFFF
sProfinetName	STRING(240)	PROFINET device name. Max. 240 characters and the following characters permitted "a..z", "0..9", ".", "-"
bNotChangeable	BOOL	If "TRUE", the PROFINET controller cannot change the PROFINET name.

Outputs

```

VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorID   : UDINT;
END_VAR
    
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the bBusy output is reset.
nErrorId	UDINT	Returns an <u>ADS error number</u> [▶ 43] if bError output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4024.55	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag (>= v1.5.1.0)

3.2.3 FB_PN_SEND_ALARM



The function block FB_PN_SEND_ALARM sends an alarm to the controller.

Inputs

```

VAR_INPUT
  bStart      : BOOL;
  NETID      : T_AmsNetId;
  PORT       : T_AmsPort;
  PN_ALARM_Typ : E_PN_ALARM_TYP;
  PN_slotNumber : WORD;
  PN_SubSlotNumber : WORD;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).
PN_ALARM_Typ	E_PN_ALARM_TYP [▶ 41]	Contains predefined alarm types
PN_SlotNumber	WORD	Slot number
PN_SubSlotNumber	WORD	SubSlot number

Outputs

```

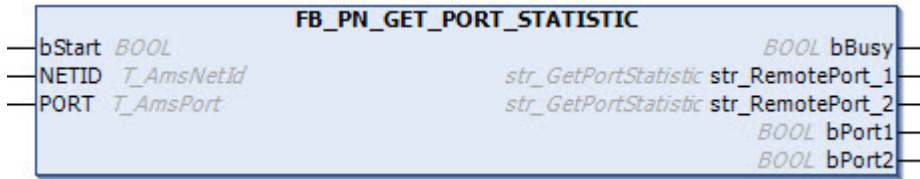
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorID   : UDINT;
END_VAR
    
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	If an error occurs during the transmission of the command, this output is set after the <i>bBusy</i> output is reset.
iErrorID	UDINT	Returns an <u>ADS error number</u> [▶ 43] if <i>bError</i> output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.3 FB_PN_GET_PORT_STATISTIC



When called, the function block FB_PN_GET_PORT_STATISTIC provides the statistical data on the ports of a PROFINET device.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  NETID       : T_AmsNetId;
  PORT        : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

Outputs

```
VAR_OUTPUT
  bBusy        : BOOL;
  str_RemotePort_1 : str_GetPortStatistic;
  str_RemotePort_2 : str_GetPortStatistic;
  bPort1       : BOOL;
  bPort2       : BOOL;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
str_RemotePort_1	str_GetPortStatistic [▶ 42]	This structure contains the statistical data for Port 1.
str_RemotePort_2	str_GetPortStatistic [▶ 42]	This structure contains the statistical data for Port 2.
bPort1	BOOL	Is TRUE if the port has a link.
bPort2	BOOL	Is TRUE if the port has a link.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

3.4 FB_PN_READ_PORT_DIAG



The function block FB_PN_READ_PORT_DIAG retrieves the diagnosis information of the ports of a PROFINET device.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  NETID      : T_AmsNetId;
  PORT       : T_AmsPort;
END_VAR
```

Name	Type	Description
bStart	BOOL	The function block is enabled by a positive edge at this input.
NETID	T_AmsNetId	AMS Net ID of the controller
PORT	T_AmsPort	Port via which the controller communicates with the device (port = Device ID + 1000hex).

Outputs

```
VAR_OUTPUT
  bBusy      : BOOL;
  str_RemotePort_1 : str_GetPortStatistic;
  str_RemotePort_2 : str_GetPortStatistic;
END_VAR
```

Name	Type	Description
bBusy	BOOL	When the function block is enabled, this output is set and remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.
str_RemotePort_1	str_GetPortStatistic [► 42]	This structure contains the statistical data for Port 1.
str_RemotePort_2	str_GetPortStatistic [► 42]	This structure contains the statistical data for Port 2.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4018	PC or CX (x64, x86, ARM)	Tc2_ProfinetDiag

4 Data structures

4.1 Controller

4.1.1 IM

4.1.1.1 str_SW_Rec

The data structure **str_SW_REC** contains the software version of the PROFINET device.

```

TYPE str_SW_Rec :
STRUCT
  cSWRevPrefix      : STRING(2);
  nSWRevFuncEnhance : BYTE;
  nSWRevBugFix      : BYTE;
  nSWRevIntCha      : BYTE;
END_STRUCT
END_TYPE

```

Name	Description
cSWRevPrefix	Revision Prefix (V = officially released Version, R = Revision, P = Prototype, U = Under Test/Field Test, T = Test Device)
nSWRevFunctionEnhance	Function extension
nSWRevBugFix	Bug Fix
nSWRevIntCha	Internal Change

4.1.1.2 str_IM_0xAFF0

The data structure **str_IM_0xAFF0** maps the structure of the I&M0 frame in the PLC. This structure is used to read from a PROFINET device and contains vendor information.

```

TYPE str_IM_0xAFF0 :
STRUCT
  nBlockTyp      : WORD;
  nBlockLen      : WORD;
  nBlockVersion  : WORD;
  nVendorID      : WORD;
  cOrderID       : STRING(21);
  cSerialNumber  : STRING(17);
  nHW_Rev        : WORD;
  strSW_Rev      : str_SW_Rec;
  nRevCount      : WORD;
  nProfileID     : WORD;
  nProfileSpecType : WORD;
  arIM_Version   : ARRAY [0..1] OF BYTE;
  nSupport       : WORD;
END_STRUCT
END_TYPE

```

4.1.1.3 str_IM_0xAFF1

The data structure **str_IM_0xAFF1** maps the structure of the I&M1 frame in the PLC. This structure is used both for reading from, and for the writing to a PROFINET device.

```

TYPE str_IM_0xAFF1 :
STRUCT
  nBlockTyp      : WORD;
  nBlockLen      : WORD;
  nBlockVersion  : WORD;
  st_IM_TagFunction : STRING(32);
  st_IM_TagLocation : STRING(22);
END_STRUCT
END_TYPE

```

4.1.1.4 str_IM_0xAFF2

The data structure **str_IM_0xAFF2** maps the structure of the I&M2 frame in the PLC. This structure is used both for reading from, and for the writing to a PROFINET device.

```

TYPE str_IM_0xAFF1 :
STRUCT
  nBlockTyp      : WORD;
  nBlockLen      : WORD;
  nBlockVersion  : WORD;
  st_IM_Date     : STRING(16);
END_STRUCT
END_TYPE

```

4.1.1.5 str_IM_0xAFF3

The data structure **str_IM_0xAFF3** maps the structure of the I&M3 frame in the PLC. This structure is used for writing to and reading from a PROFINET device.

```

TYPE str_IM_0xAFF3 :
STRUCT
  nBlockTyp      : WORD;
  nBlockLen      : WORD;
  nBlockVersion  : WORD;
  st_IM_Descriptor : STRING(54);
END_STRUCT
END_TYPE

```

4.1.1.6 str_IM_0xAFF4

The data structure **str_IM_0xAFF4** maps the structure of the I&M4 frame in the PLC. This structure is used for writing to and reading from a PROFINET device.

```

TYPE str_IM_0xAFF3 :
STRUCT
  nBlockTyp      : WORD;
  nBlockLen      : WORD;
  nBlockVersion  : WORD;
  st_IM_Signatur : STRING(54);
END_STRUCT
END_TYPE

```

4.1.2 str_PN_Scan

The data structure **str_PN_Scan** contains information about the PROFINET device.

```

TYPE str_PN_Scan :
STRUCT
  IP_Addr      : ARRAY [0..3] OF BYTE;
  SubNetMask   : ARRAY [0..3] OF BYTE;
  DefaultGateway : ARRAY [0..3] OF BYTE;
  MacID        : ARRAY [0..5] OF BYTE;
  VendorID     : WORD;
  DeviceID     : WORD;
  PN_NAME      : STRING(51);
END_STRUCT
END_TYPE

```

Name	Description
IP_Addr	IP address
SubNetMask	Subnet mask
DefaultGateway	Default gateway
MacId	MacId
VendorID	Vendor ID
DeviceID	Device ID
PN_NAME	PROFINET Name of the PROFINET device

4.1.3 AlarmDiag

4.1.3.1 ST_PN_DiagMessage

The data structure **ST_PN_DiagMessage** contains the complete data stream of a diagnostic message that is sent by a PROFINET on request. This data stream is evaluated in the FB_PN_ALARM_DIAG function block and is copied to a readable structure.

```
TYPE ST_PN_DiagMessage :  
STRUCT  
  nFlags      : WORD;  
  nTextID     : WORD;  
  TimeStamp   : ARRAY[0..7] OF BYTE;  
  nData       : ARRAY[0..299] OF BYTE;  
END_STRUCT  
END_TYPE
```

4.1.3.2 ST_PN_Diag

The data structure **ST_PN_Diag** contains a diagnosis message of a terminal that is connected to a controller via a PN device.

```
TYPE str_PortDiag :  
STRUCT  
  strTimeStamp   : ARRAY[0..7] OF BYTE;  
  nAPI           : DWORD;  
  nSlot          : WORD;  
  nSubSlot       : WORD;  
  nAlarmType     : WORD;  
  nAlarmSpecifier : WORD;  
  nUserStructIdentifier : WORD;  
  nChannelNumber : WORD;  
  nChannelErrorTyp : WORD;  
  nChannelProperties : WORD;  
  nExtChannelErrorTyp : WORD;  
  arSpare        : ARRAY [1..9] OF WORD;  
  arUserSpecificData : ARRAY [0..19] OF BYTE;  
END_STRUCT  
END_TYPE
```

The information content of the structure corresponds to that of the Diag History, which is displayed in the System Manager.

The screenshot shows the 'Diagnosis' tab in a software interface. It features a table of diagnostic messages and a detailed view of a selected alarm.

Type	Timestamp	Message	AddInfo	MessageID
Warning	20/12/2023 10:48:32 640 ms	ek9300.test: AR received diagnosis alarm (Alarmtype 0x0003, Slot 3, Subslot 2).	Yes	6
Warning	20/12/2023 10:48:32 636 ms	ek9300.test: AR received diagnosis alarm (Alarmtype 0x0003, Slot 3, Subslot 1).	Yes	5
Warning	20/12/2023 10:48:32 580 ms	ek9300.test: AR received diagnosis alarm (Alarmtype 0x0001, Slot 0, Subslot 1).	Yes	4
Info	20/12/2023 10:48:12 300 ms	ek9300.test: AR is established (got ApplReady).	No	3
Info	20/12/2023 10:48:10 001 ms	ek9300.test: Controller send PmEnd.	No	2
Info	20/12/2023 10:48:09 980 ms	ek9300.test: Controller start the parameterization.	No	1
Info	20/12/2023 10:48:09 969 ms	ek9300.test: Controller send ConnectReq to device.	No	0

Below the table, a detailed view of a 'Pull alarm (0x0003)' is shown:

```

Pull alarm (0x0003)
The diagnosis alarm received from:
API Number 0x00000000, Slot Number 0x0003, Subslot Number 0x0002
  
```

4.1.3.3 ST_PN_AlarmDiagData

The data structure **ST_PN_AlarmDiagData** contains the alarm diagnosis data record read from a device, including a time stamp that indicates when the event occurred and a flag that indicates that 'user-specific' data are present.

```
TYPE ST_PN_AlarmDiagData :
```

```
STRUCT
  ST_TimeStamp      : TIMESTRUCT;
  sNameOfStation   : STRING(20);
  ST_Diag           : ST_PN_Diag;
  bUserSpecData    : BOOL;
END_STRUCT
```

```
END_TYPE
```

4.1.4 ST_PN_DeviceInfo

The data structure **ST_PN_DeviceInfo** contains information about the PROFINET controller.

```
TYPE str_PN_DeviceInfo :
```

```
STRUCT
  nBoxAddr          : UINT;
  sBoxName          : STRING(240);
  sIP_Addr          : STRING(15);
  sSubNetMask       : STRING(15);
  sDefaultGateway   : STRING(15);
  PnIoState         : WORD;
  PnIoDiag          : WORD;
  nNrOfInputCR      : UINT;
  nNrOfOutputCR     : UINT;
  nCycleTime        : UDINT;
END_STRUCT
```

```
END_TYPE
```


Name	Description
nBoxAddr	BOX ID from TwinCAT
sBoxName	PROFINET box/device name
sIP_Addr	IP address
sSubNetMask	Subnet mask
sDefaultGateway	Default gateway
PnIoState	PROFINET state
PnIoDiag	PROFINET diagnosis
nNoOfInputCR	Number of input CR (Communication Relation)
nNoOfOutputCR	Number of output CR (Communication Relation)
nCycleTime	PROFINET cycle time in μ s

4.2 Device

4.2.1 str_Diag_PN_Settings

The data structure **str_Diag_PN_Settings** contains IP information that is permanently stored in PROFINET devices.

```

TYPE str_Diag_PN_Settings :
STRUCT
  IP_Addr          : ARRAY [0..3] OF BYTE;
  Sub_Net_Mask    : ARRAY [0..3] OF BYTE;
  Default_Gateway : ARRAY [0..3] OF BYTE;
  st_PN_Name      : STRING(255);
END_STRUCT
END_TYPE
    
```

4.2.2 str_IuM_Data

The data structure **str_IuM_Data** contains information about the PROFINET device.

```

TYPE str_IuM_Data :
STRUCT
  st_IM_TagFunction      : STRING(32);
  st_IM_TagLocation     : STRING(22);
  st_IM_Date            : STRING(16);
  st_IM_Description     : STRING(54);
  st_IM_Signature       : STRING(54);
END_STRUCT
END_TYPE
    
```

4.3 EL6631-0010

4.3.1 E_PN_ALARM_TYP

E_PN_ALARM_TYP

The enumeration type **E_PN_ALARM_TYP** lists all PROFINET communication alarms.

```

TYPE E_PN_ALARM_TYP :
(
  PN_ALARM_RESERVE           :=0,
  PN_ALARM_DIAGNOSE_APPEARS :=1,
  PN_ALARM_PROCESS          :=2,
  PN_ALARM_PULL             :=3,
  PN_ALARM_PLUG             :=4,
  PN_ALARM_STATUS           :=5,
  PN_ALARM_UPDATE           :=6,
  PN_ALARM_REDUNDANCY       :=7,
  PN_ALARM_Controlled_by_Supervisor :=8,
  PN_ALARM_Released         :=9,
  PN_ALARM_Plug_Wrong_Submodule :=16#A,
)
    
```

```

    PN_ALARM_Diagnosis_Disappears           :=16#B,
    PN_ALARM_Multicast_Communication_Mismatch :=16#C,
    PN_ALARM_Multicast                       :=16#D,
    PN_ALARM_STATUS_                         :=16#E,
    PN_ALARM_Sync_                           :=16#F,
    PN_ALARM_Isochronous_Mode_Problem_Notification :=16#10
);
END_TYPE

```

4.4 IM

4.4.1 RecStruct

The data structure **RecStruct** maps the PROFINET record data, i.e. the acyclic parameter data, in the PLC.

```

TYPE RecStruct :
STRUCT
    nRw           : WORD; (* 0 = read / 1 = write *)
    nNrOfAR      : WORD; (* Number of Application Relationship*)
    nAPI         : DWORD; (* Application Process Identifier *)
    nSlot        : WORD; (* Coupler = 0 / Terminal > 0*)
    nSubSlot     : WORD; (* 1 *)
    nIndex       : WORD; (* Register e.g. 0xAFF0 = IM0 *)
    nHwLength    : WORD; (* Read = 0 *)
    nLenOfAlign  : WORD; (* 0 *)
END_STRUCT
END_TYPE

```

4.5 Port diagnosis

4.5.1 str_GetPortStatistic

All statistical information of a device is represented in the data structure **str_GetPortStatistic**.

```

TYPE str_GetPortStatistic :
STRUCT
    Speed           : DWORD;
    PhyMAC          : STRING(50);
    OperatingStatus : STRING(16);
    RxOctets        : DWORD;
    RxUniCastPackets : DWORD;
    RxBadPackets   : DWORD;
    RxDroppedFrames : DWORD;
    RxUnknownProtocol : DWORD;
    TxOctets        : DWORD;
    TxUniCastPackets : DWORD;
    TxBadPackets   : DWORD;
    TxDroppedPackets : DWORD;
END_STRUCT
END_TYPE

```

4.5.2 str_PortDiag

All port diagnostic information is represented in the data structure **str_PortDiag**.

```

TYPE str_PortDiag :
STRUCT
    PortId           : STRING(128);
    PortDescription  : STRING(128);
    SystemName       : STRING(128);
    SystemDescription : STRING(128);
    ChassisId        : STRING(128);
END_STRUCT
END_TYPE

```

5 Appendix

5.1 ADS Return Codes

Grouping of error codes:

Global error codes: [ADS Return Codes \[▶ 43\]](#)... (0x9811_0000 ...)

Router error codes: [ADS Return Codes \[▶ 43\]](#)... (0x9811_0500 ...)

General ADS errors: [ADS Return Codes \[▶ 44\]](#)... (0x9811_0700 ...)

RTime error codes: [ADS Return Codes \[▶ 46\]](#)... (0x9811_1000 ...)

Global error codes

Hex	Dec	HRESULT	Name	Description
0x0	0	0x98110000	ERR_NOERROR	No error.
0x1	1	0x98110001	ERR_INTERNAL	Internal error.
0x2	2	0x98110002	ERR_NORTIME	No real time.
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	Allocation locked – memory error.
0x4	4	0x98110004	ERR_INSERTMAILBOX	Mailbox full – the ADS message could not be sent. Reducing the number of ADS messages per cycle will help.
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	Wrong HMSG.
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	Target port not found – ADS server is not started or is not reachable.
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	Target computer not found – AMS route was not found.
0x8	8	0x98110008	ERR_UNKNOWNCMDID	Unknown command ID.
0x9	9	0x98110009	ERR_BADTASKID	Invalid task ID.
0xA	10	0x9811000A	ERR_NOIO	No IO.
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	Unknown AMS command.
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 error.
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	Port not connected.
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	Invalid AMS length.
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Invalid AMS Net ID.
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installation level is too low – TwinCAT 2 license error.
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	No debugging available.
0x12	18	0x98110012	ERR_PORTDISABLED	Port disabled – TwinCAT system service not started.
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port already connected.
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 error.
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync error.
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	No index map for AMS Sync available.
0x18	24	0x98110018	ERR_INVALIDAMSSPORT	Invalid AMS port.
0x19	25	0x98110019	ERR_NOMEMORY	No memory.
0x1A	26	0x9811001A	ERR_TCPSEND	TCP send error.
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	Host unreachable.
0x1C	28	0x9811001C	ERR_INVALIDAMSFAGMENT	Invalid AMS fragment.
0x1D	29	0x9811001D	ERR_TLSSSEND	TLS send error – secure ADS connection failed.
0x1E	30	0x9811001E	ERR_ACCESSDENIED	Access denied – secure ADS access denied.

Router error codes

Hex	Dec	HRESULT	Name	Description
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	Locked memory cannot be allocated.
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	The router memory size could not be changed.
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages.
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	The Debug mailbox has reached the maximum number of possible messages.
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	The port type is unknown.
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	The router is not initialized.
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	The port number is already assigned.
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	The port is not registered.
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	The maximum number of ports has been reached.
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	The port is invalid.
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	The router is not active.
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	The mailbox has reached the maximum number for fragmented messages.
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	A fragment timeout has occurred.
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	The port is removed.

General ADS error codes

Hex	Dec	HRESULT	Name	Description
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	General device error.
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	Service is not supported by the server.
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	Invalid index group.
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	Invalid index offset.
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	Reading or writing not permitted.
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	Parameter size not correct.
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	Invalid data values.
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	Device is not ready to operate.
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	Device is busy.
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	Invalid operating system context. This can result from use of ADS blocks in different tasks. It may be possible to resolve this through multitasking synchronization in the PLC.
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	Insufficient memory.
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	Invalid parameter values.
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	Not found (files, ...).
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	Syntax error in file or command.
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	Objects do not match.
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	Object already exists.
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	Symbol not found.
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	Invalid symbol version. This can occur due to an online change. Create a new handle.
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	Device (server) is in invalid state.
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode not supported.
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	Notification handle is invalid.
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	Notification client not registered.
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	No further handle available.
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	Notification size too large.
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	Device not initialized.
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	Device has a timeout.
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	Interface query failed.
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	Wrong interface requested.
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class ID is invalid.
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object ID is invalid.
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	Request pending.
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	Request is aborted.
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	Signal warning.
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	Invalid array index.
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	Symbol not active.
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	Access denied.
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	Missing license.
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	License expired.
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	License exceeded.
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	Invalid license.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	License problem: System ID is invalid.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	License not limited in time.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Licensing problem: time in the future.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSESETIMETOLONG	License period too long.
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception at system startup.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	License file read twice.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Invalid signature.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Invalid certificate.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public key not known from OEM.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	License not valid for this system ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo license prohibited.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Invalid function ID.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Outside the valid range.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Invalid alignment.
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Invalid platform level.

Hex	Dec	HRESULT	Name	Description
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Context – forward to passive level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Context – forward to dispatch level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Context – forward to real time.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Client error.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARAM	Service contains an invalid parameter.
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling list is empty.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var connection already in use.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	The called ID is already in use.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNC TIMEOUT	Timeout has occurred – the remote terminal is not responding in the specified ADS timeout. The route setting of the remote terminal may be configured incorrectly.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Error in Win32 subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Invalid client timeout value.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port not open.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	No AMS address.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Internal error in Ads sync.
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Hash table overflow.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Key not found in the table.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESVM	No symbols in the cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Invalid response received.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port is locked.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was cancelled.

RTime error codes

Hex	Dec	HRESULT	Name	Description
0x1000	4096	0x98111000	RTERR_INTERNAL	Internal error in the real-time system.
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	Timer value is not valid.
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	Task pointer has the invalid value 0 (zero).
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	Stack pointer has the invalid value 0 (zero).
0x1004	4100	0x98111004	RTERR_PrioEXISTS	The request task priority is already assigned.
0x1005	4101	0x98111005	RTERR_NOMORETCB	No free TCB (Task Control Block) available. The maximum number of TCBs is 64.
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	No free semaphores available. The maximum number of semaphores is 64.
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	No free space available in the queue. The maximum number of positions in the queue is 64.
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	An external synchronization interrupt is already applied.
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	No external sync interrupt applied.
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	Application of the external synchronization interrupt has failed.
0x1010	4112	0x98111010	RTERR_IRQNOTLESSOREQUAL	Call of a service function in the wrong context
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel VT-x extension is not supported.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel VT-x extension is not enabled in the BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Missing function in Intel VT-x extension.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Activation of Intel VT-x fails.

Specific positive HRESULT Return Codes:

HRESULT	Name	Description
0x0000_0000	S_OK	No error.
0x0000_0001	S_FALSE	No error. Example: successful processing, but with a negative or incomplete result.
0x0000_0203	S_PENDING	No error. Example: successful processing, but no result is available yet.
0x0000_0256	S_WATCHDOG_TIMEOUT	No error. Example: successful processing, but a timeout occurred.

TCP Winsock error codes

Hex	Dec	Name	Description
0x274C	10060	WSAETIMEDOUT	A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not respond.
0x274D	10061	WSAECONNREFUSED	Connection refused - no connection could be established because the target computer has explicitly rejected it. This error usually results from an attempt to connect to a service that is inactive on the external host, that is, a service for which no server application is running.
0x2751	10065	WSAEHOSTUNREACH	No route to host - a socket operation referred to an unavailable host.
More Winsock error codes: Win32 error codes			

5.2 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our [download finder](#) contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for [local support and service](#) on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline: +49 5246 963-157
e-mail: support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline: +49 5246 963-460
e-mail: service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963-0
e-mail: info@beckhoff.com
web: www.beckhoff.com

More Information:
www.beckhoff.com/te1000

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

