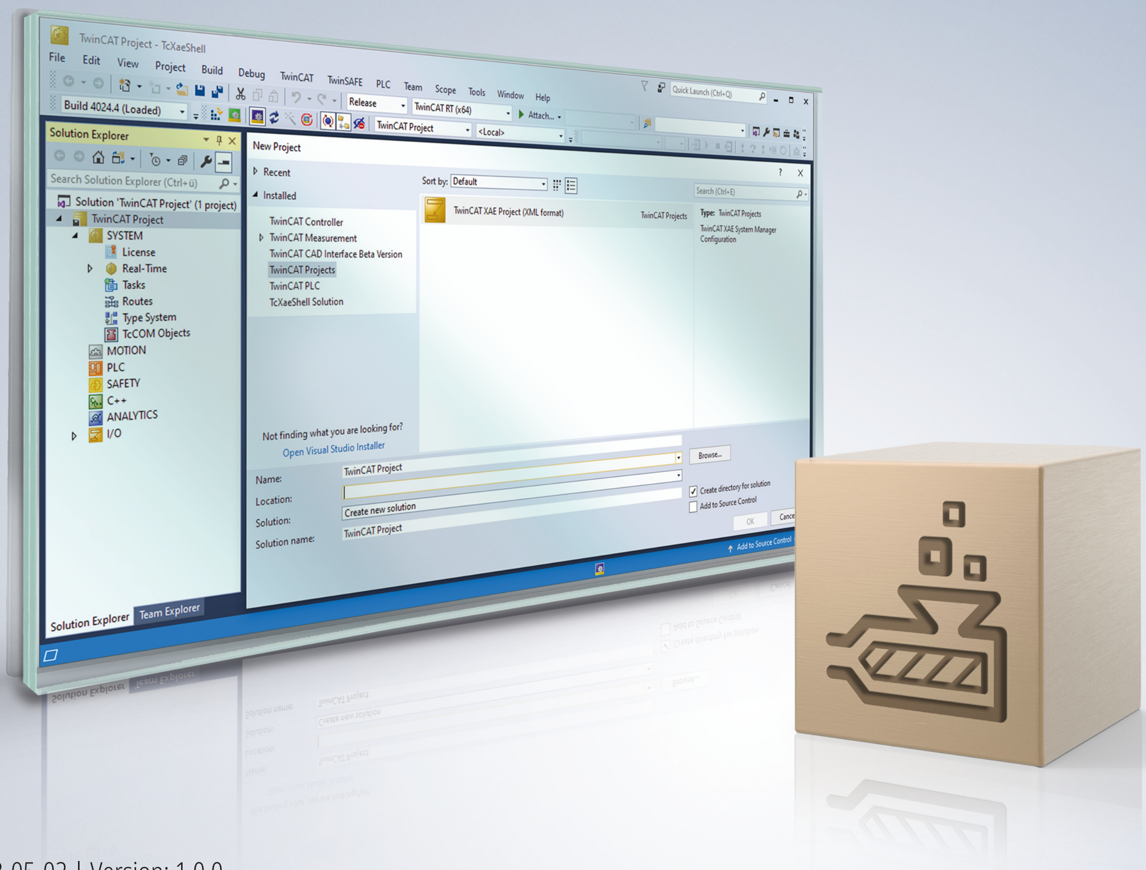
BECKHOFF New Automation Technology

Manual | EN

TF8540

TwinCAT 3 | Plastic Processing Framework



2023-05-02 | Version: 1.0.0

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 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

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 Introduction

The Plastics Processing Framework facilitates the setup of PLC projects for plastics machines. The goal of the Plastics Processing Framework is to combine the basic elements that are present in every plastics machine into one package.

Specifically, the following topics are covered:

- Temperature control
  Autotune
  Loading/ saving of parameters
- Message output
- Filter blocks
- Timers

This documentation is divided into three different modules:

- The first section describes the controller in more detail.
  Controller function
  Load and save functions
  Filter functions
- In the second section a weekly timer is presented to activate the controller also timer-controlled.
- The last section describes functions to realize message handling
  Add messages to a list or remove them from a list,
  Enabling or disabling messages

Each module is organized as follows:

- Overview: Gives an overview of this module.
- Function blocks: The individual function blocks for this module are explained.
- Structures: The individual structures for this function block are explained.
- Knowledge Base: Frequently asked questions and commissioning instructions can be found here.

Software version

The documentation is based on software version 1.0.9 of PfwLib_Processing.lib. In older versions of PfwLib_Processing.lib it cannot be guaranteed that all functions explained here are implemented.

Knowledge Base

All the functions, function blocks and data types present in this framework part are listed here. You will find answers to frequently asked questions and notes on the use of the framework, commissioning, problem analysis and sample projects in the Knowledge Base. Observe the notes on documentation.

Some of the components listed here are not intended to be used by an application. Their presence, interface and behavior is therefore not guaranteed. Because, however, a TwinCAT PLC Framework is strictly open, it is not possible to hide these internal components. It is, nevertheless, essential to avoid calling these components, identified with (internal use only) or (not recommended), directly from an application. If one of these components would, in practice, be useful for you, please make contact with our Support Department. We will then examine the possibility of making the function block available to you, independently of the library, and for you to then take the responsibility for using it.

If the library contains function blocks, types or constants that are not listed in the documentation, then these are elements that have not yet been approved, and are the subject of current software maintenance and development work. These elements must never be directly used in an application, because they are, as a general rule, not yet tested.

The framework makes a number of hardware and software requirements.

3 PLC temperature controller

3.1 Introduction

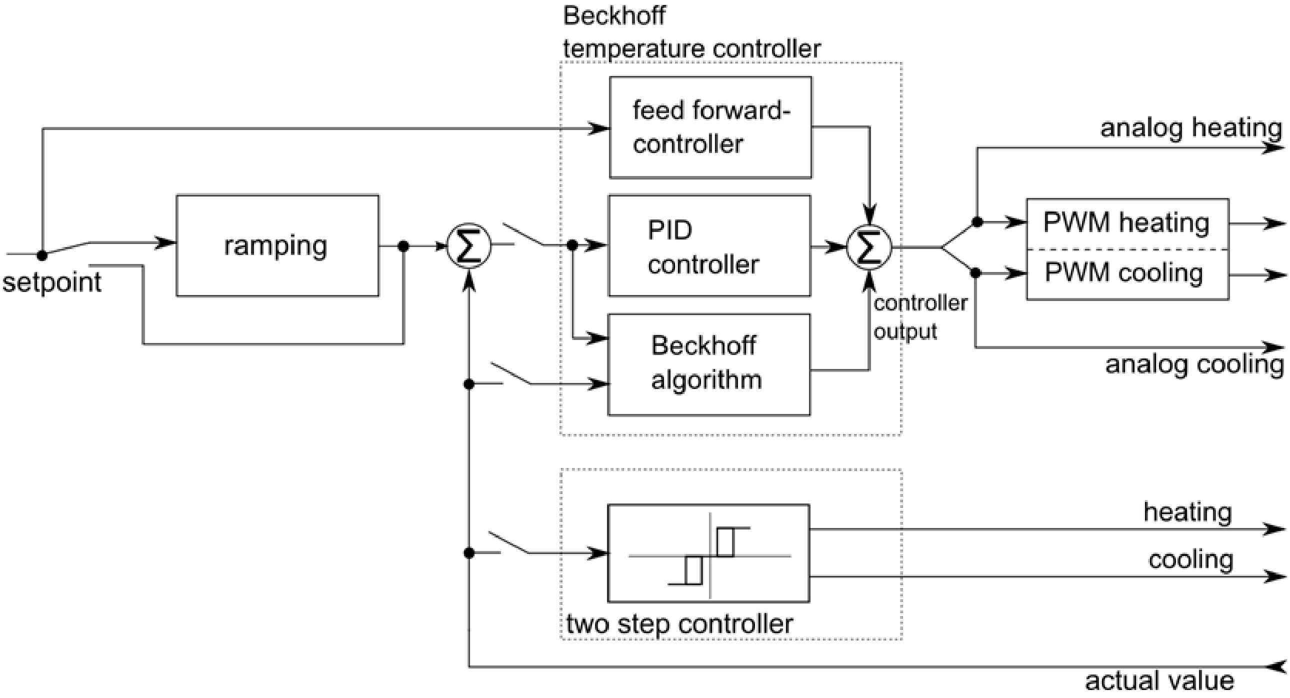
Temperature control of the Plastics Processing Framework

The temperature controller of the Plastics Processing Framework has been optimized for

- a quick start-up,
- overshot-free response to setpoint changes,
- fast correction of faults,
- optimal heating and cooling ratio,
- independent parameter determination and
- exact response to setpoint changes during production

. These requirements cannot be realized with a normal PID controller, which is why further algorithms were implemented in addition to the PID controller.

The following figure shows schematically the structure of the controller.



Determination of the path parameters:

- Takes place via the inflectional tangent method.
- Separate control parameter determination for heating and cooling.
- Determination of the idle load:

  Reduces overshoot on setpoint change.

  Parameter determination must be carried out in the controlled state.

  Takes only a few seconds and does not affect the actual temperature.

The control behavior:

- Large setpoint step-changes: Controlled by the Beckhoff algorithm. After reaching the set temperature, the PID controller is added. The optimally matched controller combinations enable fast and overshot-free control.
- Small setpoint step-changes: Are controlled by a modified PID controller, which has been optimized to handle large dead times.

- Exact response to setpoint changes: To guarantee exact response to setpoint changes during production, other stabilization measures have been taken in addition to the PID controller, such as taking transport and friction energy into account.

Other special features of the controller:

- Support of common thermocouples and Pt sensors.
- Error heating: Enables production even if the temperature sensor is defective.
- Extruder compensation: Calculates the shear rate and material transport into the process value.
- Zoning: Distribution of switch-on times within a PWM cycle avoids unnecessary power peaks.
- Power control heating tapes: Measures and checks the heating power of the heating tapes.
- I/O swap: If an input or output channel is defective, the operator can swap it from one channel to the other during operation.
- The temperature controller handles all terminal communication, so the controller only needs to be parameterized.
- Load and save routines are available for separate storage of product and machine data.
- Various filter functions (e.g. HMI filter) are available.
- Alarm handling functions are available.
- The library monitors:

  the operating state of the terminals,

  the autotuning,

  whether the current temperature exceeds certain limits,

  whether there is an appropriate temperature change for the heating power output.

Useful hints:

- Sample program: see Knowledge Base at Commissioning [▸ 88].
- Commissioning instructions: see Knowledge Base [▸ 61].
- How do I include other functions (power measurement, scope functions, etc.): see Knowledge Base in the FAQs [▸ 78].

3.2 Overview

Function blocks in Temp.-Ctrl. Folder

Name	Description
FB_TempCtrlMainBody_TcPfw [▸ 12]	Framework function block to be called cyclically.
FB_TempCtrlEnableZone_TcPfw [▸ 15]	Zones of temperature control are switched to active or passive state.
FB_TempCtrlStandByZone_TcPfw [▸ 15]	Zones of the temperature control are switched to standby or active state.
FB_TempCtrlState_TcPfw [▸ 16]	This function block determines a set of state information of a zone.
FB_TempCtrlCallback_TcPfw [▸ 17]	This function block checks the type of the linked terminal once and parameterizes it according to the specified sensor type.

Utilities

Name	Description
FB_TempCtrlAdaptFm33xx_TcPfw [▸ 26]	The I/O data of FM3312 or FM3332 fieldbus modules are adapted to the I/O structures of the library.
FB_TempCtrlClearSupply_TcPfw [▸ 27]	The data of the supply groups of the temperature control are initialized.
FB_TempCtrlClearZones_TcPfw [▸ 27]	The data of the controller zones of the temperature control are initialized.
FB_TermCoeRead_TcPfw [▸ 28]	Function block is used for read access to EL terminals.
FB_TermCoeWrite_TcPfw [▸ 29]	Function block is used for write access to EL terminals.
FB_TermRegRead_TcPfw [▸ 31]	Function block is used for read access to KL terminals.
FB_TermRegWrite_TcPfw [▸ 32]	Function block is used for write access to KL terminals.
FUN_TempCtrlSensorTypeCode_TcPfw	The function returns the numerical translation for a textual sensor type.
FUN_TempCtrlSensorTypeName_TcPfw	The function returns the textual translation for a numeric sensor type.

Framework function blocks in the StandAlone folder

Name	Description
FB_TempParamLoad_TcPfw [▸ 24]	This function block reads the parameters of a zone from a file.
FB_TempParamSave_TcPfw [▸ 21]	This function block writes the parameters of a zone into a file.
FB_TempParamSaveP_TcPfw [▸ 22]	This function block writes the parameters of a zone into a file.

Data types: Enumerations

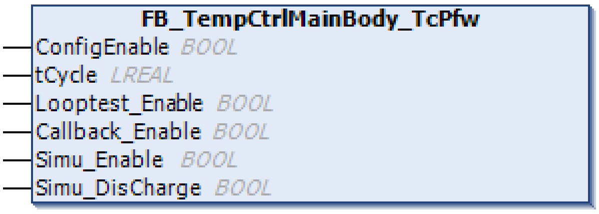
Name	Description
E_TcPfw_TctrlOutSelect [▸ 34]	Identifiers for selecting the output signal of a zone of temperature control.
E_TcPfw_TempSensType [▸ 35]	Identifiers for the supported types of temperature sensors.
E_TcPfw_TerminalType [▸ 36]	Framework basics: Identifiers for the supported types of I/O terminals.

Data types: Structure types

Name	Description
ST_TcPfw_FM3332_Input [▸ 42]	Such a structure contains the input data of a FM3312 or FM3332 fieldbus module.
ST_TcPfw_SupplyParam [▸ 54]	Such a structure contains the parameters and runtime data of a supply group.
ST_TcPfw_TempCtrl_Itf [▸ 56]	State and activity of a zone of temperature control.
ST_TcPfw_TempCtrlInput [▸ 38]	Input data of a zone of temperature control.
ST_TcPfw_TempCtrlOutput [▸ 37]	Output data of a zone of temperature control.
ST_TcPfw_TempMparamFromHmi_Itf [▸ 43]	Machine data of a zone of temperature control.
ST_TcPfw_TempPparamFromHmi_Itf [▸ 52]	Product data of a zone of temperature control.
ST_TcPfw_TempToHmi_Itf [▸ 53]	Visualization data of a zone of temperature control.

3.3 Function blocks

3.3.1 FB_TempCtrlMainBody_TcPfw()



```
FB_TempCtrlMainBody_TcPfw
ConfigEnable BOOL
tCycle LREAL
Looptest_Enable BOOL
Callback_Enable BOOL
Simu_Enable BOOL
Simu_DisCharge BOOL
```

This function block must be called in the application. It organizes internally the complete temperature control.

Syntax

```
VAR_INPUT
ConfigEnable    : BOOL;
tCycle          : LREAL;
Looptest_Enable: BOOL;
Callback_Enable: BOOL;
Simu_Enable     : BOOL:=FALSE;
Simu_DisCharge : BOOL:=FALSE;
END_VAR
```

Inputs

Name	Type	Description
ConfigEnable	BOOL	If TRUE, the configuration is valid.
tCycle	LREAL	Cycle time
Looptest_Enable	BOOL	A current measurement is performed.
Callback_Enable	BOOL	Checks once (by calling FB_TempCtrlCallback_TcPfw()) the type of the linked terminal and parameterizes it according to the specified sensor type.
Simu_Enable	BOOL	Used for internal simulation purposes.
Simu_DisCharge	LREAL	Used for internal simulation purposes.

Behavior of the function block:

With each call the function block checks the global variable bPfw_UseTempControl. If this variable is TRUE and ConfigEnable indicates a valid configuration, the function block becomes active:

- In the first cycle the function block calls an internal function block of type FB_internal_tmpCtrlInitlinks_TcPfw() to initialize the structures used by the temperature control.
- If Looptest_Enable is set, a current measurement is performed.
- If Callback_Enable is set, a function block of type FB_TempCtrlCallback_TcPfw() is called. This function block checks the type of the linked terminal once and parameterizes it according to the specified sensor type.
- The following activities are performed for each zone of the control:

  If in aaaPfwTempMparamFromHmi the Signal Update is set, the entered values are limited to the permissible value ranges if required and taken over into the control. Update is deleted.

  If in aaaPfwTempPparamFromHmi the Signal Update is set, the entered values are limited to the permissible value ranges if required and taken over into the control. Update is deleted.

  If InUse is set to TRUE in aaaPfwTempToHmi, the following steps are performed:

  - A FB_CTRL_TempController() function block from the TcTempCtrl.LIB library is called.

  - In out_PfwTempCtrlOutput YPWMPos, YPWMNeg, YDigPos, YDigNeg and Yanalog are updated.

- Various signals (aaaTempFault_Reset, Autotune in aaaPfwTempMparamFromHmi etc.) control the resetting of fault conditions or activate autotuning.

- If a problem is reported by the controller function block or from the I/O interface, the corresponding events are activated.

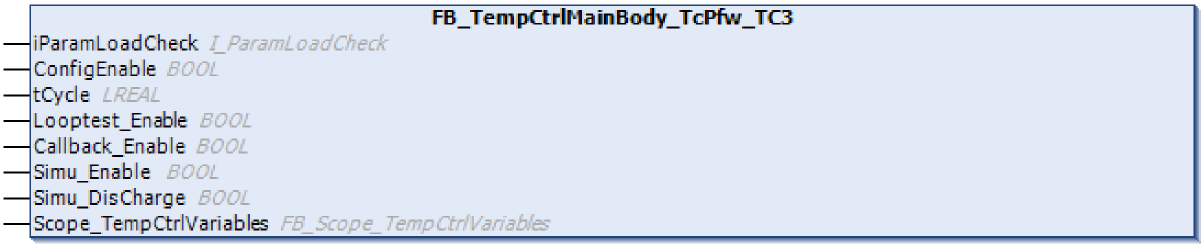
- Various data in aaaPfwTempToHmi are updated.

- If the actual temperature of at least one zone is below fAbsoluteLow in aaaPfwTempMparamFromHmi, aaaTempAlarm_AbsoluteLow is signaled.
- If the actual temperature of at least one zone is above fAbsoluteHigh in aaaPfwTempPparamFromHmi, aaaTempAlarm_AbsoluteHigh is signaled.

At the end of the cycle aaaTempFault_Reset is automatically deleted.

If Callback_Enable is not set, no function block of type FB_TempCtrlCallback_TcPfw() is called. In this case, the application must ensure that the I/O electronics match the sensor type.

3.3.2 FB_TempCtrlMainBody_TcPfw_TC3()


FB_TempCtrlMainBody_TcPfw_TC3
iParamLoadCheck I_ParamLoadCheck
ConfigEnable BOOL
tCycle LREAL
Looptest_Enable BOOL
Callback_Enable BOOL
Simu_Enable BOOL
Simu_DisCharge BOOL
Scope_TempCtrlVariables FB_Scope_TempCtrlVariables

This function block must be called cyclically in the application. It organizes internally the complete temperature control.

Syntax

```
VAR_INPUT
    iParamLoadCheck:I_ParamLoadCheck;
    ConfigEnable            : BOOL;
    tCycle                  : LREAL;
    Looptest_Enable         : BOOL;
    Callback_Enable         : BOOL;
    Simu_Enable             : BOOL:=FALSE;
    Simu_DisCharge          : BOOL:=FALSE;
    Scope_TempCtrlVariables: FB_Scope_TempCtrlVariables;
END_VAR
```

Inputs

Name	Type	Description
iParamLoadCheck	I_ParamLoadCheck	Optional interface for (external) storage of temperature zones.
ConfigEnable	BOOL	If TRUE, the configuration is valid
tCycle	LREAL	Cycle time
Looptest_Enable	BOOL	A current measurement is performed.
Callback_Enable	BOOL	Checks once (by calling FB_TempCtrlCallback_TcPfw()) the type of the linked terminal and parameterizes it according to the specified sensor type.
Simu_Enable	BOOL	Used for internal simulation purposes.
Simu_DisCharge	BOOL	Used for internal simulation purposes.
Scope_TempCtrlVariables	FB_Scope_TempCtrlVariables	Display of the individual temperature zones and their internal variables.

Behavior of the function block:

With each call the function block checks the global variable bPfw_UseTempControl. If this variable is TRUE and ConfigEnable indicates a valid configuration, the function block becomes active:

- In the first cycle the function block calls an internal function block of type FB_internal_tmpCtrlInitlinks_TcPfw() to initialize the structures used by the temperature control.
- If Looptest_Enable is set, a current measurement is performed.
- If Callback_Enable is set, a function block of type FB_TempCtrlCallback_TcPfw() is called. This function block checks the type of the linked terminal once and parameterizes it according to the specified sensor type.
- The following activities are performed for each zone of the control:
    - If in aaaPfwTempMparamFromHmi the Signal Update is set, the entered values are limited to the permissible value ranges if required and taken over into the control. Update is deleted.
    - If in aaaPfwTempPparamFromHmi the Signal Update is set, the entered values are limited to the permissible value ranges if required and taken over into the control. Update is deleted.
    - If InUse is set to TRUE in aaaPfwTempToHmi, the following steps are performed:

        - A FB_CTRL_TempController() function block from the TcTempCtrl.LIB library is called.

        - In out_PfwTempCtrlOutput YPWMPos, YPWMNeg, YDigPos, YDigNeg and Yanalog are updated.

        - Various signals (aaaTempFault_Reset, Autotune in aaaPfwTempMparamFromHmi etc.) control the resetting of fault conditions or activate autotuning.

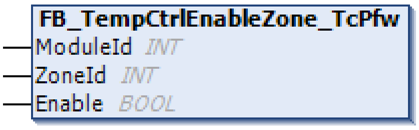
        - If a problem is reported by the controller function block or from the I/O interface, the corresponding events are activated.

        - Various data in aaaPfwTempToHmi are updated.
- If the actual temperature of at least one zone is below fAbsoluteLow in aaaPfwTempMparamFromHmi, aaaTempAlarm_AbsoluteLow is signaled.
- If the actual temperature of at least one zone is above fAbsoluteHigh in aaaPfwTempPparamFromHmi, aaaTempAlarm_AbsoluteHigh is signaled.

At the end of the cycle aaaTempFault_Reset is automatically deleted.

If Callback_Enable is not set, no function block of type FB_TempCtrlCallback_TcPfw() is called. In this case, the application must ensure that the I/O electronics match the sensor type.

3.3.3 FB_TempCtrlEnableZone_TcPfw()



One or more function blocks of this type are called from a function block of the application to switch individual zones, zone groups or all zones of the temperature control to the active or passive state.

Syntax

```
VAR_INPUT
ModuleId: INT:=-1;
ZoneId   : INT:=-1;
Enable   : BOOL:=TRUE;
END_VAR
```

Inputs

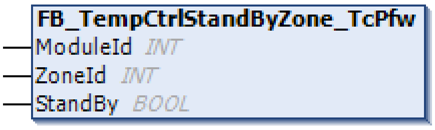
Name	Type	Description
ModuleId	INT	ModuleId of the temperature zones to be controlled.
ZoneId	INT	ZoneId of the temperature zones to be controlled.
Enable	BOOL	New state of the temperature zones.

Behavior of the function block:

- To control a zone in a group, both the ModuleId and the ZoneId must be specified.
- To control all zones in a group, the ModuleId must be specified. The value 0 is used here as ZoneId.
- To control all zones in all groups, use the value 0 for both ModuleId and ZoneId.

In the zone(s) selected in this way, ST_TcPfw_TempCtrl_Itf.Enable is updated with Enable.

3.3.4 FB_TempCtrlStandByZone_TcPfw()



One or more function blocks of this type are called from a function block of the application to switch individual zones, zone groups or all zones of the temperature control into the standby or active state.

Syntax

```
VAR_INPUT
ModuleId   : INT:=-1;
ZoneId     : INT:=-1;
StandBy    : BOOL:=TRUE;
END_VAR
```

Inputs

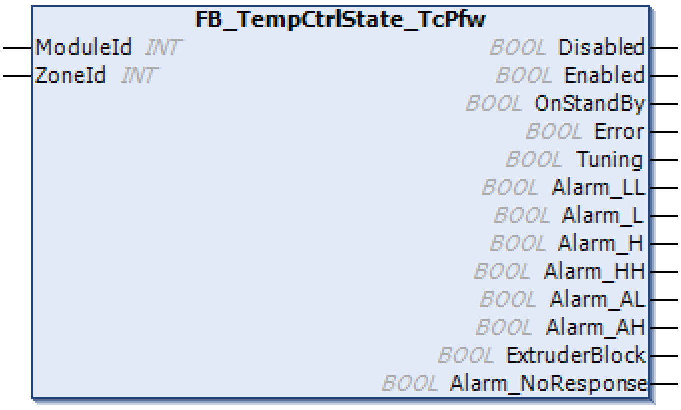
Name	Type	Description
ModuleId	INT	ModuleId of the temperature zones to be controlled.
ZoneId	INT	ZoneId of the temperature zones to be controlled.
StandBy	BOOL	New state of the temperature zones.

Behavior of the function block:

- To control a zone in a group, both the ModuleId and the ZoneId must be specified.
- To control all zones in a group, the ModuleId must be specified. The value 0 is used here as ZoneId.
- To control all zones in all groups, use the value 0 for both ModuleId and ZoneId.

In the zone(s) selected in this way ST_TcPfw_TempCtrl_Itf.SelSetpoint is updated with StandBy.

3.3.5 FB_TempCtrlState_TcPfw()



This function block determines a set of state information of a zone.

Syntax

```
VAR_INPUT
    ModuleId:INT:=-1;
    ZoneId:INT:=-1;
END_VAR
VAR_OUTPUT
    Disabled          : BOOL;
    Enabled           : BOOL;
    OnStandBy         : BOOL;
    Error             : BOOL;
    Tuning            : BOOL;
    Alarm_LL          : BOOL;
    Alarm_L           : BOOL;
    Alarm_H           : BOOL;
    Alarm_HH          : BOOL;
    Alarm_AL          : BOOL;
    Alarm_AH          : BOOL;
    ExtruderBlock     : BOOL;
    Alarm_NoResponse : BOOL;
END_VAR
```

Inputs

Name	Type	Description
ModuleId	INT	ModuleId of the temperature zones to be controlled.
ZoneId	INT	ZoneId of the temperature zones to be controlled.

Outputs

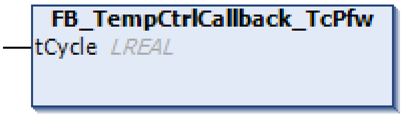
Name	Type	Description
Disabled	BOOL	TRUE if the zone is not enabled.
Enabled	BOOL	TRUE if the zone is enabled.
OnStandBy	BOOL	TRUE if the zone is enabled and switched to lowering.
Error	BOOL	TRUE if the zone is in an error state.
Tuning	BOOL	TRUE if the automatic tuning function of the zone is active.
Alarm_LL	BOOL	TRUE if the actual temperature of the zone falls below the outer negative tolerance threshold.
Alarm_L	BOOL	TRUE if the actual temperature of the zone falls below the inner negative tolerance threshold.
Alarm_H	BOOL	TRUE if the actual temperature of the zone falls below the inner positive tolerance threshold.
Alarm_HH	BOOL	TRUE if the actual temperature of the zone exceeds the outer positive tolerance threshold.
Alarm_AL	BOOL	TRUE if the actual temperature of the zone falls below the negative absolute alarm threshold.
Alarm_AH	BOOL	TRUE if the actual temperature of the zone exceeds the positive absolute alarm threshold.
ExtruderBlock	BOOL	TRUE if the actual temperature of the zone falls below the negative absolute alarm threshold and this causes the extruder to shut down.
Alarm_NoResponse	BOOL	TRUE if the actual temperature of the zone does not show an appropriate response to the heating power. Possible cause is, for example, an incorrectly mounted sensor or a defect in the heater that cannot be detected by other means.

Behavior of the function block:

In each cycle, the function block updates the state of one or more zones. Here, the behavior is determined in detail by the call:

- If the function block is called with unknown ModuleId>0 and/or an unknown ZoneId>0 or if aaaPfwTempToHmi[..].InUse=FALSE all outputs are FALSE.
- If ModuleId<>0 and ZoneId=0, the states of all zones of the module are ORed.
- If ModuleId<>0 and ZoneId<>0, the state of the selected zone is reported.

3.3.6 FB_TempCtrlCallback_TcPfw()



This function block checks the type of the linked terminal once and parameterizes it according to the specified sensor type.

Syntax

```
VAR_INPUT
    tCycle: LREAL;
END_VAR
```

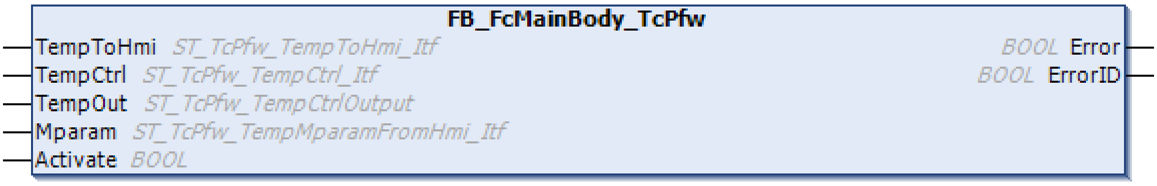
Inputs

Name	Type	Description
tCycle	LREAL	Cycle time

Behavior of the function block:

By means of aaaPfwTempMparamFromHmi[..].TempSensTerm the function block detects whether the I/O electronics used support register communication or acyclic CoE communication. If this is the case, the terminal type is read out from the module and checked. If the hardware type matches the software setting, the module is set to the sensor type specified in aaaPfwTempMparamFromHmi[..].SensorType.

3.3.7 FB_FcMainBody_TcPfw()


FB_FcMainBody_TcPfw
TempToHmi ST_TcPfw_TempToHmi_Itf
TempCtrl ST_TcPfw_TempCtrl_Itf
TempOut ST_TcPfw_TempCtrlOutput
Mparam ST_TcPfw_TempMparamFromHmi_Itf
Activate BOOL
BOOL Error
BOOL ErrorID

This function block must be called by the application after the TempCtrl_FB_TempCtrlMainBody_TcPfw. The function block organizes the intermittent switching on of the cooling independently of the controller output.

Syntax

```
VAR_INPUT
    Activate:     BOOL;
END_VAR
VAR_IN_OUT
    TempToHmi  : ST_TcPfw_TempToHmi_Itf;
    TempCtrl   : ST_TcPfw_TempCtrl_Itf;
    TempOut    : ST_TcPfw_TempCtrlOutput;
    Mparam     : ST_TcPfw_TempMparamFromHmi_Itf;
END_VAR
VAR_OUTPUT
    Error    : BOOL;
    ErrorID : BOOL; (* not used yet *)
END_VAR
```

Inputs

Name	Type	Description
Activate	BOOL	Only with a TRUE the forced cooling is really output.

/ Inputs/outputs

Name	Type	Description
TempToHmi	ST_TcPfw_TempToHmi_Itf	A reference to the data on the HMI of the zone must be provided here.
TempCtrl	ST_TcPfw_TempCtrl_Itf	A reference to the runtime data of the zone must be provided here.
TempOut	ST_TcPfw_TempCtrlOutput	A reference to the output interface of the zone must be provided here.
Mparam	ST_TcPfw_TempMparamFromHmi_Itf	A reference to the machine parameters of the zone must be provided here.

Outputs

Name	Type	Description
Error	BOOL	Indicates when something is misconfigured.
ErrorID	BOOL	Not used.

Behavior of the function block:

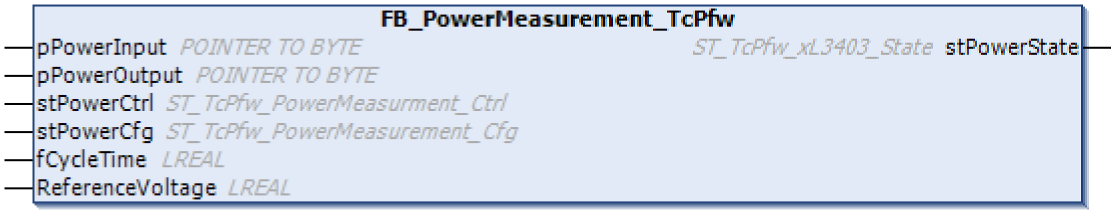
A prerequisite for proper functioning is that the zone is "InUse" and has cooling. Furthermore, fc_Enable must be enabled in the machine parameters of this zone and realistic times must have been set for the cooling time fc_OnTime and the pause time fpwmOffTime.

If the fpwmOffTime has expired, cooling becomes active for the time fc_OnTime minus the cooling power already output.

3.4 Utilities

3.4.1 Stand Alone

3.4.1.1 FB_PowerMeasurement_TcPfw()



FB_PowerMeasurement_TcPfw

pPowerInput *POINTER TO BYTE*

pPowerOutput *POINTER TO BYTE*

stPowerCtrl *ST_TcPfw_PowerMeasurment_Ctrl*

stPowerCfg *ST_TcPfw_PowerMeasurement_Cfg*

fCycleTime *LREAL*

ReferenceVoltage *LREAL*

ST_TcPfw_xL3403_State stPowerState

This function block must be called in the application. It organizes internally the complete temperature control.

Syntax

```
VAR_INPUT
    pPowerInput  : POINTER TO BYTE;
    pPowerOutput : POINTER TO BYTE;
    stPowerCtrl  : ST_TcPfw_PowerMeasurment_Ctrl;
    stPowerCfg   : ST_TcPfw_PowerMeasurement_Cfg;
    fCycleTime   : LREAL:=0.025;
END_VAR
VAR_OUTPUT
    stPowerState : ST_TcPfw_xL3403_State;
END_VAR
```

Inputs

Name	Type	Description
pPowerInput	POINTER TO BYTE	Pointer to the input structure of the power measurement terminal.
pPowerOutput	POINTER TO BYTE	Pointer to the output structure of the power measurement terminal.
stPowerCtrl	ST_TcPfw_PowerMeasurment_Ctrl	Allows the current voltage to be read out separately.
stPowerCfg	ST_TcPfw_PowerMeasurement_Cfg	Configuration of the power measurement terminal.
fCycleTime	LREAL	Transfer of the cycle time for this function block.

Outputs

Name	Type	Description
stPowerState	ST_TcPfw_xL3403_State	Here the current power, current and error states are reported back.

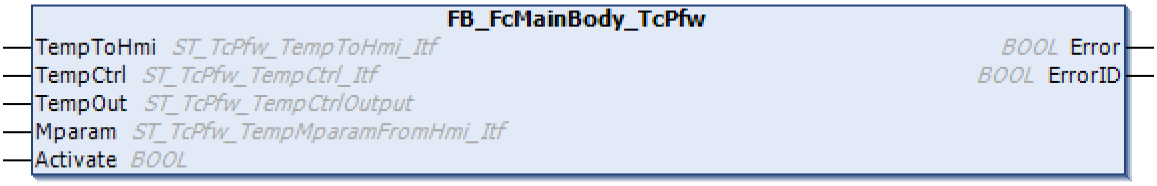
Behavior of the function block:

This function block must be called cyclically by the application. The function block receives the mapping interface from the application as a pointer via pPowerInput and pPowerOutput. Depending on the selected power measurement terminal in stPowerCfg the pointer addresses are interpreted. Depending on the supply line, the function block distributes the measured services to the individual SupplyLines.

The pointer address and the stored terminal type must match at all times. Otherwise there will be wrong memory accesses.

When measuring power with the EL3403, the increased filter time must also be taken into account.

3.4.1.2 FB_FcMainBody_TcPfw()


FB_FcMainBody_TcPfw
TempToHmi ST_TcPfw_TempToHmi_Itf
TempCtrl ST_TcPfw_TempCtrl_Itf
TempOut ST_TcPfw_TempCtrlOutput
Mparam ST_TcPfw_TempMparamFromHmi_Itf
Activate BOOL
BOOL Error
BOOL ErrorID

This function block must be called by the application after the TempCtrl_FB_TempCtrlMainBody_TcPfw. The function block organizes the intermittent switching on of the cooling independently of the controller output.

Syntax

```
VAR_INPUT
    Activate:   BOOL;
END_VAR
VAR_IN_OUT
    TempToHmi: ST_TcPfw_TempToHmi_Itf;
    TempCtrl : ST_TcPfw_TempCtrl_Itf;
    TempOut  : ST_TcPfw_TempCtrlOutput;
    Mparam   : ST_TcPfw_TempMparamFromHmi_Itf;
END_VAR
VAR_OUTPUT
    Error    : BOOL;
    ErrorID  : BOOL; (* not used yet *)
END_VAR
```

Inputs

Name	Type	Description
Activate	BOOL	Only with a TRUE the forced cooling is really output.

Inputs/outputs

Name	Type	Description
TempToHmi	ST_TcPfw_TempToHmi_Itf	A reference to the data on the HMI of the zone must be provided here.
TempCtrl	ST_TcPfw_TempCtrl_Itf	A reference to the runtime data of the zone must be provided here.
TempOut	ST_TcPfw_TempCtrlOutput	A reference to the output interface of the zone must be provided here.
Mparam	ST_TcPfw_TempMparamFromHmi_Itf	A reference to the machine parameters of the zone must be provided here.

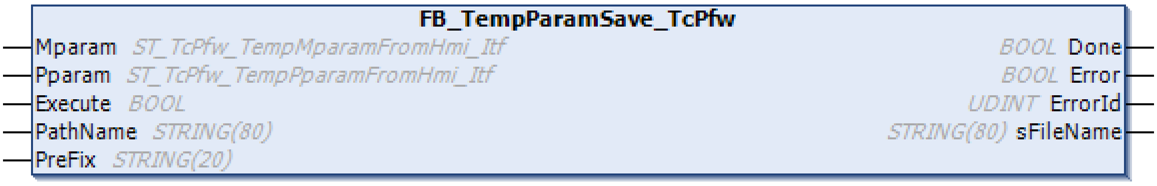
Outputs

Name	Type	Description
Error	BOOL	Indicates when something is misconfigured.
ErrorID	BOOL	Not used.

Behavior of the function block:

A prerequisite for proper functioning is that the zone is "InUse" and has cooling. Furthermore, fc_Enable must be enabled in the machine parameters of this zone and realistic times must have been set for the cooling time fc_OnTime and the pause time fpwmOffTime. If the fpwmOffTime has expired, cooling becomes active for the time fc_OnTime minus the cooling power already output.

3.4.1.3 FB_TempParamSave_TcPfw()



This function block writes the parameters of a zone into a file. A FB_TempParamLoad_TcPfw function block must be used to read the file.

Instead of this function block, you can alternatively call a save via the SaveDelay in the machine parameters, the product parameters or the SupplyLines.

Syntax

```
VAR_INPUT
    Execute   : BOOL;
    PathName  : STRING(80);
    PreFix    : STRING(20);
END_VAR
VAR_IN_OUT
    Mparam    : ST_TcPfw_TempMparamFromHmi_Itf;
    Pparam    : ST_TcPfw_TempPparamFromHmi_Itf;
END_VAR
VAR_OUT
    Done      : BOOL;
    Error     : BOOL;
    ErrorId   : DINT;
    sFileName : STRING(80);
END_VAR
```

Inputs

Name	Type	Description
Execute	BOOL	A rising edge at this input starts the process. With a FALSE at Execute all outputs are deleted. This ensures that they are present for at least one cycle.
PathName	STRING	The path name to be used must be provided here.
PreFix	STRING	Prefix to be used before the filename.

Inputs/outputs

Name	Type	Description
Mparam	ST_TcPfw_TempMparamFromHmi_Itf	A reference to the machine parameters of the zone must be provided here.
Pparam	ST_TcPfw_TempPparamFromHmi_Itf	A reference to the product parameters of the zone must be provided here.

Outputs

Name	Type	Description
Done	BOOL	A TRUE indicates here the successful processing of the command.
Error	BOOL	A TRUE indicates here the occurrence of a problem during the processing of the command.
ErrorId	DINT	If an error has occurred, coded information about the nature of the problem is provided here.
sFileName	STRING	Name of the saved file.

Behavior of the function block:

On a rising edge at Execute, the function block forms a filename from PathName, the textual name of the zone and STRING constants.

Example:

'C:\Parameter\Tctrl_Zone1.par' is formed from PathName:='C:\Parameter\' and Mparam.ZoneName:='Zone1'.

The parameters are written in an encoded binary format that cannot be edited with a text editor. The coding makes the format largely insensitive to version differences. As a rule, files are readable even if they were written by older or younger versions of the library.

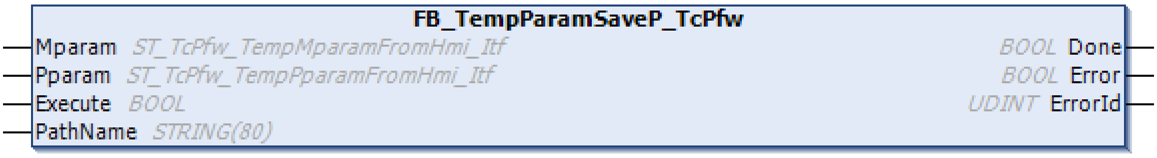
NOTE

If the product parameters are to be saved independently of the machine data of the zone, one FB_TempParamSave_TcPfw() function block and one FB_TempParamSaveP_TcPfw() function block must be used. To avoid mutual overwriting of files with the same name, the path names must be chosen differently. At system startup first the machine data and then the product parameters are to be loaded with FB_TempParamLoad_TcPfw() function blocks.

NOTE

New parameters may be added when the version of the library is changed. These are filled with default values whose effect does not always produce the desired behavior.

3.4.1.4 FB_TempParamSaveP_TcPfw()



This function block writes the product parameters of a zone into a file. A FB_TempParamLoad_TcPfw function block must be used to read the file.

Syntax

```
VAR_INPUT
    Execute : BOOL;
    PathName: STRING(80);
END_VAR
```

```
VAR_IN_OUT
    Mparam   : ST_TcPfw_TempMparamFromHmi_Itf;
    Pparam   : ST_TcPfw_TempPparamFromHmi_Itf;
END_VAR
VAR_OUT
    Done     : BOOL;
    Error    : BOOL;
    ErrorId  : DINT;
END_VAR
```

Inputs

Name	Type	Description
Execute	BOOL	The process is initiated by a rising edge at this input. With a FALSE at Execute all outputs are deleted. This ensures that they are present for at least one cycle.
PathName	STRING	The path name to be used must be provided here.

Inputs/outputs

Name	Type	Description
Mparam	ST_TcPfw_TempMparamFromHmi_Itf	A reference to the machine parameters of the zone must be provided here.
Pparam	ST_TcPfw_TempPparamFromHmi_Itf	A reference to the product parameters of the zone must be provided here.

Outputs

Name	Type	Description
Done	BOOL	A TRUE indicates here the successful processing of the command.
Error	BOOL	A TRUE indicates here the occurrence of a problem during the processing of the command.
ErrorId	BOOL	If an error has occurred, coded information about the nature of the problem is provided here.

Behavior of the function block:

On a rising edge at Execute, the function block forms a filename from PathName, the textual name of the zone and STRING constants.

Example:

'C:\Parameter\Tctrl_Zone1.par' is formed from PathName:='C:\Parameter\' and Mparam.ZoneName:='Zone1'.

The parameters are written in an encoded binary format that cannot be edited with a text editor. The coding makes the format largely insensitive to version differences. As a rule, files are readable even if they were written by older or younger versions of the library.

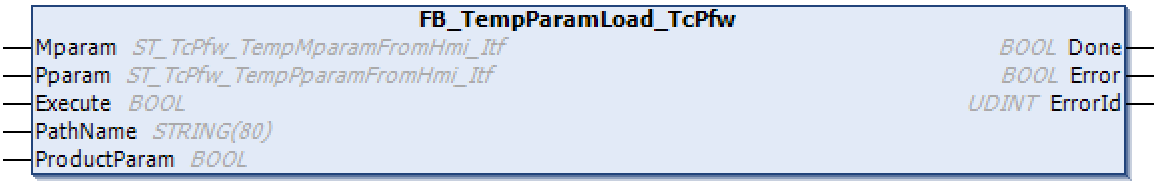
NOTE

If the product parameters are to be saved independently of the machine data of the zone, one FB_TempParamSave_TcPfw() function block and one FB_TempParamSaveP_TcPfw() function block must be used. To avoid mutual overwriting of files with the same name, the path names must be chosen differently. At system startup first the machine data and then the product parameters are to be loaded with FB_TempParamLoad_TcPfw() function blocks.

NOTE

New parameters may be added when the version of the library is changed. These are filled with default values whose effect does not always produce the desired behavior.

3.4.1.5 FB_TempParamLoad_TcPfw()



FB_TempParamLoad_TcPfw

Mparam ST_TcPfw_TempMparamFromHmi_Itf — BOOL Done

Pparam ST_TcPfw_TempPparamFromHmi_Itf — BOOL Error

Execute BOOL — UDINT ErrorId

PathName STRING(80)

ProductParam BOOL

This function block reads the parameters of a zone from a file. For writing the file a FB_TempParamSave_TcPfw function block must be used.

Syntax

```
VAR_INPUT
    Execute      : BOOL;
    PathName     : STRING(80);
    ProductParam :BOOL:=FALSE;
END_VAR
VAR_IN_OUT
    Mparam       : ST_TcPfw_TempMparamFromHmi_Itf;
    Pparam       : ST_TcPfw_TempPparamFromHmi_Itf;
END_VAR
VAR_OUT
    Done         : BOOL;
    Error        : BOOL;
    ErrorId      : DINT;
END_VAR
```

Inputs

Name	Type	Description
Execute	BOOL	The process is initiated by a rising edge at this input. With a FALSE at Execute all outputs are deleted. This ensures that they are present for at least one cycle.
PathName	STRING	The path name to be used must be provided here.
ProductParam	BOOL	If TRUE, the product data will be loaded.

Inputs/outputs

Name	Type	Description
Mparam	ST_TcPfw_TempMparamFromHmi_Itf	A reference to the machine parameters of the zone must be provided here.
Pparam	ST_TcPfw_TempPparamFromHmi_Itf	A reference to the product parameters of the zone must be provided here.

Outputs

Name	Type	Description
Done	BOOL	A TRUE indicates here the successful processing of the command.
Error	BOOL	A TRUE indicates here the occurrence of a problem during the processing of the command.
ErrorId	DINT	If an error has occurred, coded information about the nature of the problem is provided here.

Behavior of the function block:

On a rising edge at Execute, the function block forms a filename from PathName, the textual name of the zone and STRING constants.

Example:

'C:\Parameter\Tctrl_Zone1.par' is formed from PathName:='C:\Parameter\' and Mparam.ZoneName:='Zone1'.

The parameters are read in an encoded binary format that cannot be edited with a text editor. The coding makes the format largely insensitive to version differences. As a rule, files are readable even if they were written by older or younger versions of the library.

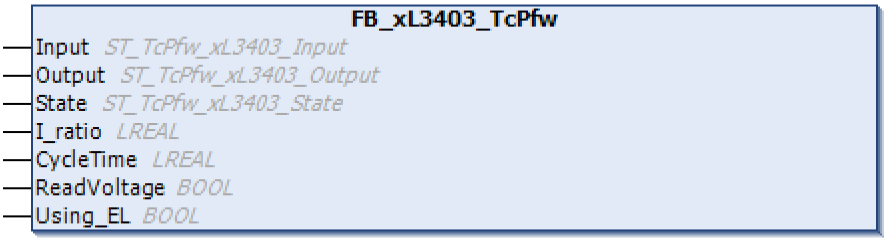
NOTE

If the product parameters are to be saved independently of the machine data of the zone, one FB_TempParamSave_TcPfw() function block and one FB_TempParamSaveP_TcPfw() function block must be used. To avoid mutual overwriting of files with the same name, the path names must be chosen differently. At system startup first the machine data and then the product parameters are to be loaded with FB_TempParamLoad_TcPfw() function blocks.

NOTE

New parameters may be added when the version of the library is changed. These are filled with default values whose effect does not always produce the desired behavior.

3.4.2 FB_xL3403_TcPfw()


FB_xL3403_TcPfw
Input ST_TcPfw_xL3403_Input
Output ST_TcPfw_xL3403_Output
State ST_TcPfw_xL3403_State
I_ratio LREAL
CycleTime LREAL
ReadVoltage BOOL
Using_EL BOOL

This function block processes the data (voltage, line) determined by an xL3403 and makes it available to the application.

The FB_PowerMeasurement_TcPfw() function block should be called by the application. This function block calls FB_xL3403_TcPfw() internally.

Syntax

```
VAR_INPUT
    I_ratio     : LREAL:=1.0;
    CycleTime   : LREAL:=0.025;
    ReadVoltage : BOOL:=FALSE;
    Using_EL    : BOOL:=FALSE;
END_VAR
VAR_IN_OUT
    Input       : ST_TcPfw_xL3403_Input;
    Output      : ST_TcPfw_xL3403_Output;
    State       : ST_TcPfw_xL3403_State;
END_VAR
```

Inputs

Name	Type	Description
I_ratio	LREAL	The ratio of the current transformers must be entered here.
CycleTime	LREAL	Cycle time with which this function block is called.
ReadVoltage	BOOL	A TRUE causes the currently measured voltage to be read out instead of the power.
Using_EL	BOOL	A TRUE tells the function block that it is an EL terminal.

Inputs/outputs

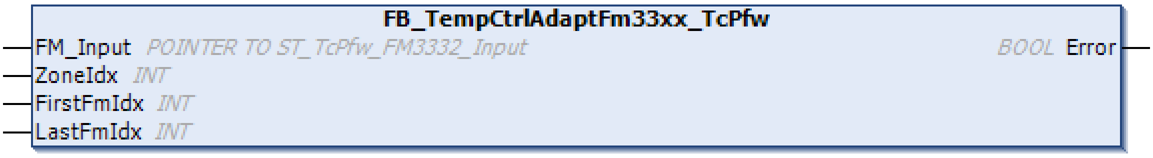
Name	Type	Description
Input	ST_TcPfw_xL3403_Input	Provides the input data of the terminal.
Output	ST_TcPfw_xL3403_Output	Provides the output data of the terminal.
State	ST_TcPfw_xL3403_State	Returns the state of the terminal and the processed data to the application.

Behavior of the function block:

If the terminal does not report an error, the prepared input data are provided in each cycle.

When measuring power with the EL3403, the increased filter time must also be taken into account.

3.4.3 FB_TempCtrlAdaptFm33xx_TcPfw()


FB_TempCtrlAdaptFm33xx_TcPfw
FM_Input POINTER TO ST_TcPfw_FM3332_Input
ZoneIdx INT
FirstFmIdx INT
LastFmIdx INT
BOOL Error

The I/O data of FM3312 or FM3332 fieldbus modules are adapted to the I/O structures of the library.

This function block must be called in the application. It organizes internally the complete temperature control.

Syntax

```
VAR_INPUT
FM_Input   : POINTER TO ST_TcPfw_FM3332_Input;
ZoneIdx    : INT;
FirstFmIdx: INT;
LastFmIdx : INT;
END_VAR
VAR_OUTPUT
Error      : BOOL;
END_VAR
```

Inputs

Name	Type	Description
FM_Input	POINTER TO ST_TcPfw_FM3332_Input	The address of a single variable or an array of type ST_TcPfw_FM3332_Input.
ZoneIdx	INT	The index of the zone to which the process image is to be assigned.
FirstFmIdx	INT	The first index of the process image provided as FM_Input.
LastFmIdx	INT	The last index of the process image provided as FM_Input.

Outputs

Name	Type	Description
Error	BOOL	Any problems with call parameters or zone parameters are signaled here.

Behavior of the function block:

If one of the call parameters is outside the permissible range, this is reported with Error. Furthermore, Error is reported if the call parameters are correct but the addressed zone has an invalid setting in aaaPfwTempMparamFromHmi[ZoneIdx].TermChannel in its parameters.

Two modes can be used here:

- If in FM_Input the address of the only ST_TcPfw_FM3332_Input process image of the application or within an array the address of the process image responsible for this zone is provided, the index (1..32) of the input in the process image must be specified as TermChannel.
- If in FM_Input the address of the first ST_TcPfw_FM3332_Input process image of an array is provided, the index of the input in the process image array is to be specified as TermChannel. This index is 1..32 for the inputs of the first module, 33..64 for the inputs of the second module and so on.

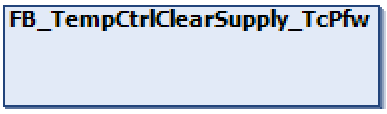
NOTE

The same image is used for modules with less than 32 inputs. The channels not implemented in the module's hardware then remain unused, but are counted when determining the input index as described above.

Otherwise the data of the ST_TcPfw_FM3332_Input process image is converted to the ST_TcPfw_TempCtrlInput process image of the zone:

- The process value for the actual temperature is compatible and is copied.
- An EL_SnsWcState for connection monitoring is derived from the DpState.
- The bit in OpenCircuit[..] belonging to the measuring channel is mapped as SNS_Overrange bit in KL_SnsState.
- The bit in Backvoltage[..] belonging to the measuring channel is mapped as SNS_GeneralError bit in KL_SnsState.

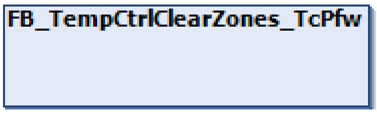
3.4.4 FB_TempCtrlClearSupply_TcPfw()



A function block of this type must be called once in the initialization phase of the application before the parameters are provided.

The function block initializes the data of the supply groups of the temperature control.

3.4.5 FB_TempCtrlClearZones_TcPfw()

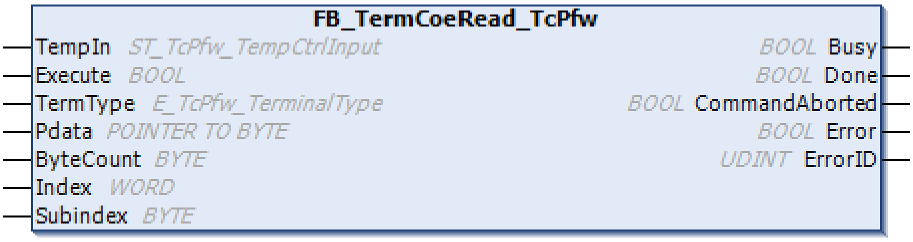


A function block of this type must be called once in the initialization phase of the application before the parameters are provided.

The function block initializes the data of all zones of the temperature control:

- aaaPfwTempPparamFromHmi
- aaaPfwTempMparamFromHmi
- aaaPfwTempToHmi
- aaaTempCtrl
- out_PfwTempCtrlOutput

3.4.6 FB_TermCoeRead_TcPfw()



A function block of this type is used by FB_TempCtrlCallback_TcPfw for read access to EL terminals.

Syntax

```
VAR_INPUT
     Execute      : BOOL;
    TermType      : E_TcPfw_TerminalType:=eTcPfwTermT_NoTerminal;
    Pdata         : POINTER TO BYTE:=0;
    ByteCount     : BYTE:=0;
    Index         : WORD:=0;
    Subindex      : BYTE:=0;
END_VAR
VAR_IN_OUT
    TempIn        : ST_TcPfw_TempCtrlInput;
END_VAR
VAR_OUTPUT
    Busy          : BOOL:=FALSE;
    Done          : BOOL:=FALSE;
    CommandAborted: BOOL:=FALSE;
    Error         : BOOL:=FALSE;
    ErrorID       : UDINT:=0;
END_VAR
```

Inputs

Name	Type	Description
Execute	BOOL	A rising edge starts the process. A falling edge clears all outputs.
TermType	E_TcPfw_TerminalType	The coded type of the addressed terminal.
Pdata	POINTER TO BYTE	The destination address for the read data.
ByteCount	BYTE	The size of the data to be read in bytes.
Index	WORD	The CoE address according to the object directory of the terminal.
Subindex	BYTE	The CoE address according to the object directory of the terminal.

Inputs/outputs

Name	Type	Description
TempIn	ST_TcPfw_TempCtrlInput	The input process image of the terminal.

Outputs

Name	Type	Description
Busy	BOOL	The ongoing activity of the function block is signaled here.
Done	BOOL	The successful completion of the operation is reported here.
CommandAborted	BOOL	A TRUE indicates here that the process was aborted.
Error	BOOL	A TRUE indicates here the occurrence of a problem.
ErrorID	UDINT	In the event of an error, coded information is provided here.

Behavior of the function block:

A rising edge at Execute causes the function block to perform a series of checks.

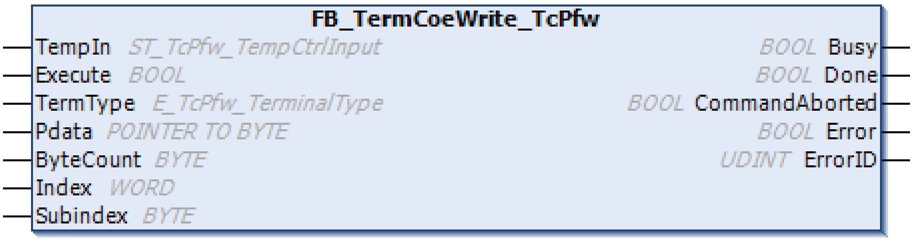
- The subindex must be in the range 0 to 127 (inclusive).
- ByteCount must be greater than 0.
- Pdata must not be 0.
- TermType must identify an EL terminal. Only these support the CoE communication mechanism used here.

If one of the above conditions is not met, an error is reported. Otherwise, the access is transmitted to the terminal. The result of the transmission is provided at the outputs.

In addition to the above-mentioned error possibilities, problems can occur during transmission. Furthermore, the terminal can report a problem (addressing, values, access type).

A CoE access requires that ST_TcPfw_TempCtrlInput.EL_AdsAddr is linked.

3.4.7 FB_TermCoeWrite_TcPfw()



A function block of this type is used by FB_TempCtrlCallback_TcPfw for write access to EL terminals.

Syntax

```
VAR_INPUT
Execute       : BOOL;
TermType      : TcPfw_TerminalType:=eTcPfwTermT_NoTerminal;
Pdata         : POINTER TO BYTE:=0;
ByteCount     : BYTE:=0;
Index         : WORD:=0;
Subindex      : BYTE:=0;
END_VAR
VAR_IN_OUT
TempIn        : ST_TcPfw_TempCtrlInput;
END_VAR
VAR_OUTPUT
Busy          : BOOL:=FALSE;
Done          : BOOL:=FALSE;
CommandAborted: BOOL:=FALSE;
Error         : BOOL:=FALSE;
ErrorID       : UDINT:=0;
END_VAR
```

Inputs

Name	Type	Description
Execute	BOOL	A rising edge starts the process. A falling edge clears all outputs.
TermType	_TcPfw_TerminalType	The coded type of the addressed terminal.
Pdata	POINTER TO BYTE	The destination address for the read data.
ByteCount	BYTE	The size of the data to be read in bytes.
Index	WORD	The CoE address according to the object directory of the terminal.
Subindex	BYTE	The CoE address according to the object directory of the terminal.

Inputs/outputs

Name	Type	Description
TempIn	ST_TcPfw_TempCtrlInput	The input process image of the terminal.

Outputs

Name	Type	Description
Busy	BOOL	The ongoing activity of the function block is signaled here.
Done	BOOL	The successful completion of the operation is reported here.
CommandAborted	BOOL	A TRUE indicates here that the process was aborted.
Error	BOOL	A TRUE indicates here the occurrence of a problem.
ErrorID	UDINT	In the event of an error, coded information is provided here.

Behavior of the function block:

A rising edge at Execute causes the function block to perform a series of checks:

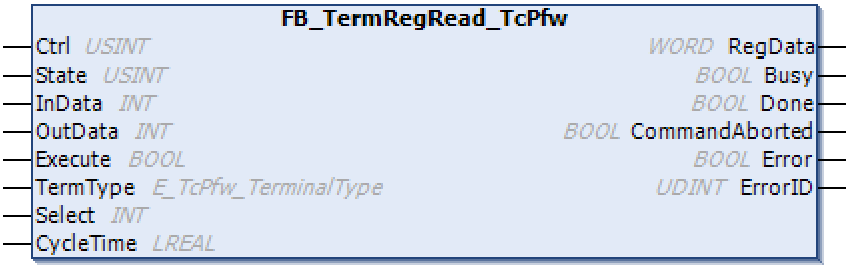
- The subindex must be in the range 0 to 127 (inclusive).
- ByteCount must be greater than 0.
- Pdata must not be 0.
- TermType must identify an EL terminal. Only these support the CoE communication mechanism used here.

If one of the above conditions is not met, an error is reported. Otherwise, the access is transmitted to the terminal. The result of the transmission is provided at the outputs.

In addition to the above-mentioned error possibilities, problems can occur during transmission. Furthermore, the terminal can report a problem (addressing, values, access type).

A CoE access requires that ST_TcPfw_TempCtrlInput.EL_AdsAddr is linked.

3.4.8 FB_TermRegRead_TcPfw()



A function block of this type is used by FB_TempCtrlCallback_TcPfw for read access to EL terminals.

Syntax

```
VAR_INPUT
    Execute        : BOOL;
    TermType       : E_TcPfw_TerminalType:=eTcPfwTermT_NoTerminal;
    Select         : INT:=-1;
    CycleTime      : LREAL:=0.025;
END_VAR
VAR_IN_OUT
    Ctrl           : USINT;
    State          : USINT;
    InData         : INT;
    OutData        : INT;
END_VAR
VAR_OUTPUT
    RegData        : WORD:=0;
    Busy           : BOOL:=FALSE;
    Done           : BOOL:=FALSE;
    CommandAborted: BOOL:=FALSE;
    Error          : BOOL:=FALSE;
    ErrorID        : UDINT:=0;
END_VAR
```

Inputs

Name	Type	Description
Execute	BOOL	A rising edge starts the process. A falling edge clears all outputs.
TermType	E_TcPfw_TerminalType	The coded type of the addressed terminal.
Select	INT	The register address of the terminal.
CycleTime	LREAL	The cycle time of the calling task.

Inputs/outputs

Name	Type	Description
Ctrl	USINT	A reference to ST_TcPfw_TempCtrlOutput.KL_SnsCtrl of the terminal.
State	USINT	A reference to ST_TcPfw_TempCtrlInput.KL_SnsState of the terminal.
InData	INT	A reference to ST_TcPfw_TempCtrlInput.KL_SnsData of the terminal.
OutData	INT	A reference to ST_TcPfw_TempCtrlOutput.KL_SnsData of the terminal.

Outputs

Name	Type	Description
RegData	WORD	If successfully executed, the read register content is provided here.
Busy	BOOL	The ongoing activity of the function block is signaled here.
Done	BOOL	The successful completion of the operation is reported here.
CommandAborted	BOOL	A TRUE indicates here that the process was aborted.
Error	BOOL	A TRUE indicates here the occurrence of a problem.
ErrorID	UDINT	In the event of an error, coded information is provided here.

Behavior of the function block:

A rising edge at Execute causes the function block to perform a series of checks:

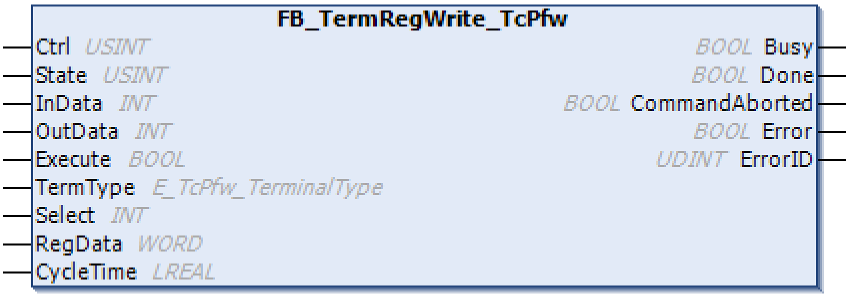
- Select must be in the range 0 to 63 (inclusive).
- No other register communication must be active with this terminal.
- TermType must identify a KL terminal. Only these support the communication mechanism used here.

If one of the above conditions is not met, an error is reported. Otherwise, the access is transmitted to the terminal. The result of the transmission is provided at the outputs.

In addition to the above-mentioned error possibilities, problems can occur during transmission. Furthermore, the terminal can report a problem (addressing, values, access type).

A register access requires that all elements with name beginning with "KL_" in ST_TcPfw_TempCtrlInput are linked.

3.4.9 FB_TermRegWrite_TcPfw()



A function block of this type is used by FB_TempCtrlCallback_TcPfw for write access to KL terminals.

Syntax

```
VAR_INPUT
    Execute         : BOOL;
    TermType        : E_TcPfw_TerminalType:=eTcPfwTermT_NoTerminal;
    Select          : INT:=-1;
    RegData         : WORD:=0;
    CycleTime       : LREAL:=0.01;
END_VAR
VAR_IN_OUT
    Ctrl            : USINT;
    State           : USINT;
    InData          : INT;
    OutData         : INT;
END_VAR
VAR_OUTPUT
    Busy            : BOOL:=FALSE;
    Done            : BOOL:=FALSE;
    CommandAborted  :BOOL:=FALSE;
    Error: BOOL     :=FALSE;
    ErrorID         : UDINT:=0;
END_VAR
```

Inputs

Name	Type	Description
Execute	BOOL	A rising edge starts the process. A falling edge clears all outputs.
TermType	E_TcPfw_TerminalType	The coded type of the addressed terminal.
Select	INT	The register address of the terminal.
RegData	WORD	The register content to be written is to be provided here.
CycleTime	LREAL	The cycle time of the calling task.

Inputs/outputs

Name	Type	Description
Ctrl	USINT	A reference to ST_TcPfw_TempCtrlOutput.KL_SnsCtrl of the terminal.
State	USINT	A reference to ST_TcPfw_TempCtrlInput.KL_SnsState of the terminal.
InData	INT	A reference to ST_TcPfw_TempCtrlInput.KL_SnsData of the terminal.
OutData	INT	A reference to ST_TcPfw_TempCtrlOutput.KL_SnsData of the terminal.

Outputs

Name	Type	Description
Busy	BOOL	The ongoing activity of the function block is signaled here.
Done	BOOL	The successful completion of the operation is reported here.
CommandAborted	BOOL	A TRUE indicates here that the process was aborted.
Error	BOOL	A TRUE indicates here the occurrence of a problem.
ErrorID	UDINT	In the event of an error, coded information is provided here.

Behavior of the function block:

A rising edge at Execute causes the function block to perform a series of checks:

- Select must be in the range 0 to 63 (inclusive).
- No other register communication must be active with this terminal.
- TermType must identify a KL terminal. Only these support the communication mechanism used here.

If one of the above conditions is not met, an error is reported. Otherwise, the access is transmitted to the terminal. The result of the transmission is provided at the outputs.

In addition to the above-mentioned error possibilities, problems can occur during transmission. Furthermore, the terminal can report a problem (addressing, values, access type).

A register access requires that all elements with name beginning with "KL_" in ST_TcPfw_TempCtrlInput are linked.

3.5 Enumerations

3.5.1 E_TcPfw_TctrlPowerTerminal

This enumeration defines identifiers for the supported types of I/O terminals. These identifiers are used in ST_TcPfw_PowerMeasurement_Cfg.

Syntax

```
TYPE E_TcPfw_TctrlPowerTerminal:
(* last modification: 16.03.2009 *)
(
NoTerminal,
KL3403,
EL3403,
EL3773,
EL3x64,
EL3413,
EL3443,
EL3446,
EL3453,
Simulation:=1000,
Customized:=10000
);
END_TYPE
```

Values

Name	Description
NoTerminal	No power measurement terminal connected.
KL3403	Bus Terminal, 3-channel analog input, power measurement, 500 V AC, 1 A, 16 bit
EL3403	EtherCAT Terminal, 3-channel analog input, power measurement, 500 V AC, 1 A, 16 bit
EL3773	EtherCAT Terminal, 3-channel analog input, multi-function, 500 V AC/DC, 1 A, 16 bit, 10 ksps, oversampling
EL3x64	EL3064 - EtherCAT Terminal, 4-channel analog input, voltage, 0...10 V, 12 bit, single-ended EL3164 - EtherCAT Terminal, 4-channel analog input, voltage, 0...10 V, 16 bit, single-ended
EL3413	EtherCAT Terminal, 3-channel analog input, power measurement, 690 V AC, 1/5 A, 16 bit, electrically isolated
EL3443	EtherCAT Terminal, 3-channel analog input, power measurement, 480 V AC/DC, 1 A, 24 bit
EL3446	EtherCAT Terminal, 6-channel analog input, current, 1 A, 24 bit, distributed power measurement
EL3453	EtherCAT Terminal, 3-channel analog input, power measurement, 690 V AC, 0.1/1/5 A, 24 bit, electrically isolated
Simulation	Power measurement via simulation.
Customized	Lower limit of the customer-specific range.

3.5.2 E_TcPfw_TctrlOutSelect

This enumeration defines identifiers for selecting the output signal of a zone of temperature control. These identifiers are used in ST_TcPfw_TempMparamFromHmi_Itf.

Syntax

```
TYPE E_TcPfw_TctrlOutSelect:
(* last modification: xx.xx.200x *)
(
eTcPfwTcOut_NoSignal,
eTcPfwTcOut_PWM,
eTcPfwTcOut_Sign,
eTcPfwTcOut_2step
);
END_TYPE
```

Values

Name	Description
eTcPfwTcOut_NoSignal	None of the output signals is selected.
eTcPfwTcOut_PWM	The PWM signal derived from the PID controller response is selected.
eTcPfwTcOut_Sign	A switching signal is selected which is derived from the sign of the controller output.
eTcPfwTcOut_2step	A switching signal is selected that is formed by an on-off controller (Schmitt trigger) from the control deviation.

3.5.3 E_TcPfw_TempSensType

This enumeration defines identifiers for the supported types of temperature sensors. These identifiers are used in ST_TcPfw_TempMparamFromHmi_Itf.

Syntax

```
TYPE E_TcPfw_TempSensType:
(* last modification: 17.02.2011 *)
(
eTcPfwTempSensT_NoSensor,
eTcPfwTempSensT_TC_B,
eTcPfwTempSensT_TC_E,
eTcPfwTempSensT_TC_J,
eTcPfwTempSensT_TC_K,
eTcPfwTempSensT_TC_L,
eTcPfwTempSensT_TC_N,
eTcPfwTempSensT_TC_R,
eTcPfwTempSensT_TC_S,
eTcPfwTempSensT_TC_T,
eTcPfwTempSensT_TC_U,

eTcPfwTempSensT_PT_100:=100,
eTcPfwTempSensT_NI_100,
eTcPfwTempSensT_NI_120,
eTcPfwTempSensT_PT_200,
eTcPfwTempSensT_PT_500,
eTcPfwTempSensT_PT_1000,
eTcPfwTempSensT_NI_1000,

eTcPfwTempsT_Customized:=10000
);
END_TYPE
```

Values

Name	Description
eTcPfwTempSensT_NoSensor	This identifier appears in the configuration of unused temperature controller zones.
eTcPfwTempSensT_TC_B	Type B sensor: 600 °C to 1800 °C
eTcPfwTempSensT_TC_E	Type B sensor: -100 °C to 1000 °C
eTcPfwTempSensT_TC_J	Type B sensor: -100 °C to 1200 °C
eTcPfwTempSensT_TC_K	Type B sensor: -100 °C to 1370 °C
eTcPfwTempSensT_TC_L	Type B sensor: -25 °C to 900 °C
eTcPfwTempSensT_TC_N	Type B sensor: -100 °C to 1300 °C
eTcPfwTempSensT_TC_R	Type B sensor: 0 °C to 1700 °C
eTcPfwTempSensT_TC_S	Type B sensor: 0 °C to 1700 °C
eTcPfwTempSensT_TC_T	Type B sensor: -100 °C to 400 °C
eTcPfwTempSensT_TC_U	Type B sensor: -25 °C to 600 °C
eTcPfwTempSensT_PT_100	PT100 sensor: -200 °C to 850 °C
eTcPfwTempSensT_NI_100	NI100 sensor: -60 °C to 250 °C.
eTcPfwTempSensT_NI_120	NI120 sensor: -60 °C to 320 °C.
eTcPfwTempSensT_PT_200	PT200 sensor: -200 °C to 850 °C
eTcPfwTempSensT_PT_500	PT500 sensor: -200 °C to 850 °C
eTcPfwTempSensT_PT_1000	PT1000 sensor: -200 °C to 850 °C
eTcPfwTempSensT_NI_1000	NI1000 sensor: -60 °C to 250 °C
eTcPfwTempsT_Customized	These and all higher numerical values mark application-specific temperature sensors.

3.5.4 E_TcPfw_TerminalType

This enumeration defines identifiers for the supported types of I/O terminals. These identifiers are used in ST_TcPfw_TempMparamFromHmi_Itf.

Syntax

```
TYPE E_TcPfw_TerminalType:
(* last modification: 16.03.2009 *)
(
eTcPfwTermT_NoSensor,

eTcPfwTermT_KL_RangeLow:=1000,
eTcPfwTermT_KL300x, (* +/-10V *)
eTcPfwTermT_KL301x, (* 0..20mA *)
eTcPfwTermT_KL302x, (* 4..20mA *)
eTcPfwTermT_KL304x, (* 0..20mA *)
eTcPfwTermT_KL305x, (* 4..20mA *)
eTcPfwTermT_KL306x, (* 0..10V *)
eTcPfwTermT_KL310x, (* +/-10V *)
eTcPfwTermT_KL311x, (* 0..20mA *)
eTcPfwTermT_KL312x, (* 4..20mA *)
eTcPfwTermT_KL313x, (* +/-10V *)
eTcPfwTermT_KL314x, (* 0..20mA *)
eTcPfwTermT_KL315x, (* 4..20mA *)
eTcPfwTermT_KL317x, (* 0..2V *)
eTcPfwTermT_KL318x, (* +/-2V *)
eTcPfwTermT_KL340x, (* +/-10V *)
eTcPfwTermT_KL344x, (* 0..20mA *)
eTcPfwTermT_KL346x, (* 0..10V *)
eTcPfwTermT_KL331x, (* thermo couple *)
eTcPfwTermT_KL_RangeHigh,

eTcPfwTermT_EL_RangeLow:=2000,
eTcPfwTermT_EL331x, (* thermo couple *)
eTcPfwTermT_EL320x, (* PT100/PT1000 *)
eTcPfwTermT_EL316x, (* 0..10V *)
eTcPfwTermT_EL_RangeHigh,
eTcPfwTermT_XX_RangeLow:=3000,
eTcPfwTermT_FM33xx, (* thermo couple *)
eTcPfwTermT_EM8908, (* multi signal backplane *)
eTcPfwTermT_XX_RangeHigh,

eTcPfwTermT_Customized:=10000
);
END_TYPE
```

Values

Name	Description
eTcPfwTermT_NoSensor	No I/O electronics available, the zone is operated in simulation.
eTcPfwTermT_KL_RangeLow	Lower limit of identifiers for K-bus terminals.
eTcPfwTermT_KL300x	K-bus terminals for connection of 1 or 2 ±10 V signals.
eTcPfwTermT_KL301x	K-bus terminals for connection of 1 or 2 0..20 mA signals.
eTcPfwTermT_KL302x	K-bus terminals for connection of 1 or 2 4..20 mA signals.
eTcPfwTermT_KL304x	K-bus terminals for connection of 1, 2 or 4 0..20 mA signals.
eTcPfwTermT_KL305x	K-bus terminals for connection of 1, 2 or 4 4..20 mA signals.
eTcPfwTermT_KL306x	K-bus terminals for the connection of 1, 2 or 4 0..10 V signals.
eTcPfwTermT_KL310x	K-bus terminals for connection of 2 ±10 V signals.
eTcPfwTermT_KL311x	K-bus terminals for connection of 2 0..20 mA signals.
eTcPfwTermT_KL312x	K-bus terminals for connection of 2 4..20 mA signals.
eTcPfwTermT_KL313x	K-bus terminals for connection of 2 ±10 V signals.
eTcPfwTermT_KL314x	K-bus terminals for connection of 2 0..20 mA signals.
eTcPfwTermT_KL315x	K-bus terminals for connection of 2 4..20 mA signals.
eTcPfwTermT_KL317x	K-bus terminals for the connection of 2 0..2 V signals.
eTcPfwTermT_KL318x	K-bus terminals for connection of 2 ±2 V signals.
eTcPfwTermT_KL340x	K-bus terminals for connection of 4 or 8 ±10 V signals.
eTcPfwTermT_KL344x	K-bus terminals for connection of 4 or 8 0..20 mA signals.
eTcPfwTermT_KL346x	K-bus terminals for the connection of 4 or 8 0..10 V signals.
eTcPfwTermT_KL331x	K-bus terminals for direct connection of 1, 2 or 4 thermocouples.
eTcPfwTermT_KL_RangeHigh	Upper limit of identifiers for K-bus terminals.
eTcPfwTermT_EL_RangeLow	Lower limit of identifiers for EtherCAT Terminals.
eTcPfwTermT_EL331x	EtherCAT Terminals for direct connection of 1, 2 or 4 thermocouples.
eTcPfwTermT_EL320x	EtherCAT Terminals for direct connection of PT100 or PT1000 sensors.
eTcPfwTermT_EL316x	EtherCAT Terminals for the connection of analog inputs.
eTcPfwTermT_EL_RangeHigh	Upper limit of identifiers for EtherCAT Terminals.
eTcPfwTermT_XX_RangeLow	Lower limit of identifiers for other modules.
eTcPfwTermT_FM33xx	Profibus module for direct connection of 12 or 32 thermocouples. Note: An adjustment function block must be called.
eTcPfwTermT_EM8908	I/O board for injection molding machines.
eTcPfwTermT_XX_RangeHigh	Upper limit of identifiers for other modules.
eTcPfwTermT_Customized	The temperature input is supplied by the application.

3.6 Structures

3.6.1 Mapping

3.6.1.1 ST_TcPfw_TempCtrlOutput

This structure contains the output data of a zone for the I/O link.

NOTE

The link established with the SystemManager to the I/O terminal variables must match the type set in aaaPfwTempMparamFromHmi[..].TempSensTerm.

Syntax

```
TYPE ST_TcPfw_TempCtrlOutput :
(* last modification: 15.07.2008 *)
STRUCT
(*
==========================================
temperature zone output data
see cnv_TempCtrlOutput_TcPfw for format definition
==========================================
*)
Htr_Analog : INT;
KL_SnsData : INT;
KL_SnsCtrl : USINT;
Htr_PwmPos : BOOL;
Htr_PwmNeg : BOOL;
Htr_DigPos : BOOL;
Htr_DigNeg : BOOL;
SelOutNeg  : BOOL;
SelOutPos  : BOOL;
NoHighAlarm: BOOL;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
Htr_Analog	INT	The power demand of the zone as an integer number in the range ±32767.
KL_SnsData	INT	When using a terminal from the KL331x type family, the "Data Off" process value must be linked here. This connection is used for register communication.
KL_SnsCtrl	USINT	When using a terminal from the KL331x type family, the "Control" process value must be linked here. This connection is used for register communication.
Htr_PwmPos	BOOL	The PWM heating signal. Should not be used anymore, please use SelOutNeg instead.
Htr_PwmNeg	BOOL	The PWM cooling signal. Should not be used anymore, please use SelOutPos instead.
Htr_DigPos	BOOL	The digital heating signal. Should not be used anymore, please use SelOutNeg instead.
Htr_DigNeg	BOOL	The digital cooling signal. Should not be used anymore, please use SelOutPos instead.
SelOutNeg	BOOL	The cooling signal selected by aaaPfwTempMparamFromHmi[..].OutputSel_C.
SelOutPos	BOOL	The heating signal selected by aaaPfwTempMparamFromHmi[..].OutputSel_H.
NoHighAlarm	BOOL	This output is TRUE as long as the temperature of the zone does not exceed the threshold aaaPfwTempMparamFromHmi[..].AbsoluteHigh.

3.6.1.2 ST_TcPfw_TempCtrlInput

This structure contains the input data of a zone for the I/O link.

NOTE

The link established with the SystemManager to the I/O terminal variables must match the type set in aaaPfwTempMparamFromHmi[..].TempSensTerm.

Syntax

```
TYPE ST_TcPfw_TempCtrlInput :
(* last modification: 11.01.2008 *)
STRUCT
(*
==========================================
temperature zone input data
see cnv_TempCtrlInput_TcPfw for format definition
==========================================
*)
KL_SnsData      : INT;
EL_SnsData      : INT;
EL_SnsState     : UINT;
KL_SnsState     : USINT;
EL_SnsUnderrun : BOOL;
EL_SnsOverrun  : BOOL;
EL_SnsError     : BOOL;
EL_SnsWcState  : BOOL;
EL_AdsAddr      : ST_TcPfw_AdsAddr;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
KL_SnsData	INT	When using a terminal from the KL331x type family, the "Data On" process value must be linked here. This connection is used to determine the actual temperature and for register communication.
EL_SnsData	INT	When using a terminal from the EL331x type family, the "State" process value must be linked here. This connection is used to determine the actual temperature.
EL_SnsState	UINT	When using a terminal from the EL331x type family, the "State" process value must be linked here. This connection is used for monitoring the terminal operating state.
KL_SnsState	USINT	When using a terminal from the KL331x type family, the "State" process value must be linked here. This connection is used for diagnostics and for register communication.
EL_SnsUnderrun	BOOL	When using a terminal from the EL331x type family, the "Underrange" signal must be linked here. This connection is used for diagnostics.
EL_SnsOverrun	BOOL	When using a terminal from the EL331x type family, the "Overrange" signal must be linked here. This connection is used for diagnostics.
EL_SnsError	BOOL	When using a terminal from the EL331x type family, the "Error" signal must be linked here. This connection is used for diagnostics.
EL_SnsWcState	BOOL	When using a terminal from the EL331x type family, the "WcState" process value must be linked here. This connection is used for connection monitoring.
EL_AdsAddr	ST_TcPfw_AdsAddr	When using a terminal from the EL331x type family, the "AdsAddr" process value must be linked here. This connection is used for CoE communication.

3.6.1.3 Power measurement

3.6.1.3.1 ST_TcPfw_EL3773_Input

Such a structure contains the input data for power measurement.

Syntax

```
TYPE ST_TcPfw_EL3773_Input:
(* location PfwLib_TempControl.PRO *)
(* last modification: 08.09.2010 *)
STRUCT
    uiStatusU1   : UINT;
    iVoltageU1   : ARRAY[1..cnOversampling] OF INT;
    uiStatusU2   : UINT;
    iVoltageU2   : ARRAY[1..cnOversampling] OF INT;
    uiStatusU3   : UINT;
    iVoltageU3   : ARRAY[1..cnOversampling] OF INT;
    uiStatusI1   : UINT;
    iCurrentI1   : ARRAY[1..cnOversampling] OF INT;
    uiStatusI2   : UINT;
    iCurrentI2   : ARRAY[1..cnOversampling] OF INT;
    uiStatusI3   : UINT;
    iCurrentI3   : ARRAY[1..cnOversampling] OF INT;

    SampleCount  : UINT;
    WcState      : BOOL;
    InputToggle  : BOOL;
    State        : UINT;

    DcOutputShift: UDINT;
    DcInputShift : UDINT;

    AdsAddr      : ST_TcPfw_AdsAddr;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
uiStatusU1	UINT	Status of the first voltage channel
iVoltageU1	ARRAY	Voltage of the first channel
uiStatusU2	UINT	Status of the second voltage channel
iVoltageU2	ARRAY	Voltage of the second channel
uiStatusU3	UINT	Status of the third voltage channel
iVoltageU3	ARRAY	Voltage of the third channel
uiStatusI1	UINT	Status of the first current channel
iCurrentI1	ARRAY	Current value of the first current channel
uiStatusI2	UINT	Status of the second current channel
iCurrentI2	ARRAY	Current value of the second current channel
uiStatusI3	UINT	Status of the third current channel
iCurrentI3	ARRAY	Current value of the third current channel
SampleCount	UINT	The SampleCounter is incremented by one unit with each process data cycle. The CycleCounter enables the higher-level controller to check whether a data record has possibly been omitted or transmitted twice. In that case the DC shift time of the terminal usually has to be adapted.
WcState	BOOL	Must be linked to the "WcState" variable of the EL terminal. This is used to detect whether the process data coming from the terminal are OK.
InputToggle	BOOL	The variable InputToggle indicates whether a new valid telegram was received. The value is incremented by one after each successful cycle.
State	UINT	Must be linked to the "State" variable of the EL terminal. This is used to return the current state of the terminal.
DcOutputShift	UDINT	DcOutputShift is the time for the output of the process data to the drive, i.e. for the time delay between the calculation and the effect of these data.
DcInputShift	UDINT	DcInputShift is the time required to transmit status information, such as the actual position of a drive, to the controller. In other words, it is the time between the acquisition and the evaluation of these data.
AdsAddr	ST_TcPfw_AdsAddr	When using the EL3773 terminal, the process value "AdsAddr" must be linked here. This connection is used for CoE communication.

3.6.1.3.2 ST_TcPfw_xL3403_Input

Such a structure contains the input data for power measurement.

Syntax

```
TYPE ST_TcPfw_xL3403_Input:
(* location PfwLib_TempControl.PRO *)
(* last modification: 08.09.2010 *)
STRUCT
    KL_DataIn      : ARRAY[1..3] OF INT;
    KL_State       : ARRAY[1..3] OF USINT;

    EL_Current     : ARRAY[1..3] OF DINT;
    EL_Voltage     : ARRAY[1..3] OF DINT;
    EL_Power       : ARRAY[1..3] OF DINT;
    EL_NoZeroCross : ARRAY[1..3] OF BOOL;
    EL_WcState     : BOOL;
    EL_State       : UINT;
    EL_AdsAddr     : ST_TcPfw_AdsAddr;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
KL_DataIn	ARRAY OF INT	Must be linked to "Data on". This is used to communicate with the terminals.
KL_State	ARRAY OF USINT	Must be linked to the "State" variable of the KL terminal. This is used to transfer the state of the terminal.
EL_Current	ARRAY OF DINT	Must be linked to the "Current" variable of the EL terminal.
EL_Voltage	ARRAY OF DINT	Must be linked to the "Voltage" variable of the EL terminal.
EL_Power	ARRAY OF DINT	Must be linked to the "Active Power" variable of the EL terminal. The power is stored in this variable.
EL_NoZeroCross	ARRAY OF BOOL	Must be linked to "Missing Zero Crossing" of the EL terminal.
EL_WcState	BOOL	Must be linked to the "WcState" variable of the EL terminal. This is used to detect whether the process data coming from the terminal are OK.
EL_State	UINT	Must be linked to the "State" variable of the EL terminal. This is used to return the current state of the terminal.
EL_AdsAddr	ST_TcPfw_AdsAddr	When using the EL3403 terminal, the process value "AdsAddr" must be linked here. This connection is used for CoE communication.

3.6.1.3.3 ST_TcPfw_xL3403_Output

Such a structure contains the output data for power measurement.

Syntax

```
TYPE ST_TcPfw_xL3403_Output :
(* location PfwLib_TempControl.PRO *)
(* last modification: 08.09.2010 *)
STRUCT
    KL_Ctrl    : ARRAY[1..3] OF USINT;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
KL_Ctrl	ARRAY OF USINT	Must be linked to "Data Off" in the terminal. This is used to define which process data the terminal is to be made available.

Example: 0 -> apparent power.

3.6.1.3.4 ST_TcPfw_FM3332_Input

Such a structure contains the input data of a FM3312 or FM3332 fieldbus module.

The link established with the SystemManager to the variables of the I/O module must match the type set in aaaPfwTempMparamFromHmi[..].TempSensTerm.

To be able to evaluate the process image of a FM33xx module, it must be distributed to the process images of the zones and converted. For this purpose a function block of type FB_TempCtrlAdaptFm33xx_TcPfw() must be used.

Syntax

```
TYPE ST_TcPfw_FM3332_Input:
(* last modification: 16.03.2009 *)
STRUCT
DpState         : USINT;
ExtDiagFlag     : BOOL;
(**)
Kanal_Daten     : ARRAY[1..32] OF UINT;
(**)
OpenCircuit     : ARRAY[0..3] OF SINT;
Backvoltage     : ARRAY[0..3] OF SINT;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
DpState	USINT	The ProfiBus DP State. This status is used to monitor communication and device status.
ExtDiagFlag	BOOL	reserved
Channel_Data	ARRAY OF UINT	The actual values of up to 32 channels.
OpenCircuit	ARRAY OF SINT	Alarm signals for up to 32 channels are combined in four bytes of 8 bits each. Reported problem: wire break in the measuring circuit.
Back voltage	ARRAY OF SINT	Alarm signals for up to 32 channels are combined in four bytes of 8 bits each. Reported problem: external voltage in the measuring circuit.

3.6.1.3.5 ST_TcPfw_xL3403_State

Such a structure contains the results of the power measurement and makes this available to the application.

Syntax

```
TYPE ST_TcPfw_xL3403_State :
(* location PfwLib_TempControl.PRO *)
(* last modification: 08.09.2010 *)
STRUCT
    Power: ARRAY[1..3] OF LREAL;
    Voltage: ARRAY[1..3] OF LREAL;
    Current: ARRAY[1..3] OF LREAL;
    LineError: ARRAY[1..3] OF BOOL;

    SubType       : INT;
    ErrorID       : INT;
    LatchedErrID  : INT;
    LatchedErr    : BOOL;
    Error         : BOOL;
    Ready         : BOOL;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
Power	ARRAY OF LREAL	Prepared power for further processing.
Voltage	ARRAY OF LREAL	Prepared tension for further processing.
Current	ARRAY OF LREAL	Prepared current for further processing.
LineError	ARRAY OF BOOL	A TRUE indicates that the terminal is in an error state.
SubType	INT	The terminal subtype, which influences the conversion factor.
ErrorID	INT	Detailed information about the error is provided via this ErrorID. The error codes can be consulted in the constants.
LatchedErrID	INT	Here the last active ErrorID is stored even after an error reset.
LatchedErr	BOOL	Latched Error = TRUE remains active even after an error reset. This must be actively reset, i.e. LatchedError = FALSE.
Error	BOOL	A TRUE indicates that the function block is in an error state. By a TRUE at the reset input of the function block FB_xL3403_TcPfw.htm it is possible to reset the function block.
Ready	BOOL	Indicates a completed conversion at a KL terminal.

3.6.2 ST_TcPfw_TempMparamFromHmi_Itf

Such a structure contains the machine data of a zone.

Syntax

```
TYPE ST_TcPfw_TempMparamFromHmi_Itf :
(* last modification: 20.12.2010 *)
STRUCT
(*
==========================================
temperature zone machine parameters see cnv_TempMparamFromHmi_TcPfw for format definition
==========================================
*)
ZoneName: STRING(79);

AbsoluteHigh         : LREAL;
AbsoluteLow          : LREAL;
ExtruderComp         : LREAL;
KpCool               : LREAL;
KpHeat               : LREAL;
TdCool               : LREAL;
TdHeat               : LREAL;
TnCool               : LREAL;
TnHeat               : LREAL;
TvCool               : LREAL;
TvHeat               : LREAL;
Overshoot            : LREAL;
Tracking_Td          : LREAL;
Ramping_Rate         : LREAL;  (* starting with cnv_TempMparamFromHmi_TcPfw=9 *)
Ramping_RateC        : LREAL;  (* starting with cnv_TempMparamFromHmi_TcPfw=15 *)
Ramping_Tolerance    : LREAL;  (* starting with cnv_TempMparamFromHmi_TcPfw=9 *)
dTmax                : LREAL;
SensorOffset         : LREAL;
SettlingTime         : LREAL;
SupplyLoad_Cooler    : LREAL;
SupplyLoad_Heater    : LREAL;
SupplyLoad_Tolerance : LREAL;
TuneEnd              : LREAL;
TuneKp               : LREAL;
TuneTd               : LREAL;
TuneTn               : LREAL;
TuneTv               : LREAL;
TuneTrackingTd       : LREAL:=0.0;  (* starting with cnv_TempMparamFromHmi_TcPfw=15 *)
TuneY                : LREAL;

L_LoadIdle           : LREAL;
Weighting_C          : LREAL;

ErrorHeatingFactor   : LREAL:=0.0;  (* starting with V1.0.8: will define default heating in error sta
te *)
fPwmStdMaxOnTime     : LREAL;
```

```
fPwmMaxOnTime         : ARRAY[cnv_TempCtrl_SetpointFirst..cnv_TempCtrl_SetpointLast] OF LREAL:=0.0;
fPwmMinOnTime         : LREAL:=0.0;
fc_OnTime             : LREAL:=0.0; (* starting with cnv_TempMparamFromHmi_TcPfw=15 *)
fc_OffTime            : LREAL:=0.0; (* starting with cnv_TempMparamFromHmi_TcPfw=15 *
ActTempGain           : LREAL:=1.0; (* starting with cnv_TempMparamFromHmi_TcPfw=18 *)
ActTempOffset         : LREAL:=0.0; (* starting with cnv_TempMparamFromHmi_TcPfw=18 *)
SaveDelay             : DINT:=-1; (* starting with cnv_TempMparamFromHmi_TcPfw=17 *)

OutputSel_H           : E_TcPfw_TctrlOutSelect:=eTcPfwTcOut_PWM;
OutputSel_C           : E_TcPfw_TctrlOutSelect:=eTcPfwTcOut_PWM;
TempSensTerm          : E_TcPfw_TerminalType :=eTcPfwTermT_NoTerminal;
SensorType            :E_TcPfw_TempSensType:=eTcPfwTempSensT_NoSensor;
App_HmiType           : INT:=0;
TermChannel           : INT;
ExtruderId            : INT;
ModuleId              : INT;
ZoneId                : INT;
SupplyId              : INT;

CJ_CompMode           : INT; (* starting with cnv_TempMparamFromHmi_TcPfw=8 *)
CJ_CompZone           : INT; (* starting with cnv_TempMparamFromHmi_TcPfw=8 *)
TermIdx               : INT; (* used to connect the zone to a terminal *)
HeaterSwapIdx         : INT; (* used for I/O re-location of selected heater signal *)
CoolerSwapIdx         : INT; (* used for I/O re-location of selected cooler signal *)
nPwmFactorC           : INT;
eTuningMethod         : E_TcPfw_TctrlTuningMethod;

InUse                 : BOOL;
UseCooling            : BOOL;
ExtruderCompEna       : BOOL;
TuneCooling           : BOOL;
Autotune              : BOOL;
StartReTune           : BOOL;
Enable                : BOOL;
Update                : BOOL;
EnaExtruderBlock      : BOOL;
NoFanWhileTrackDown   : BOOL;
Ena_TuneIdleLoad      : BOOL;
LooptestUpdate        : BOOL:=FALSE;
EnableErrorHeating    : BOOL:=FALSE; (* starting with V1.0.8: will enable default heating in error st
ate *)
ReadBack              : BOOL:=FALSE;
TuneExtruderComp      : BOOL;
TuneHeaterLoad        : BOOL:=FALSE; (* tuning heater power monitoring *)
OpenloopHeating       : BOOL:=FALSE; (* starting with cnv_TempMparamFromHmi_TcPfw=15 *

fc_Enable             : BOOL:=FALSE; (* starting with cnv_TempMparamFromHmi_TcPfw=15 *)
HibernateI_Cool       : BOOL:=FALSE; (* starting with cnv_TempMparamFromHmi_TcPfw=15 *)
HibernateI_Heat       : BOOL:=FALSE; (* starting with cnv_TempMparamFromHmi_TcPfw=15 *)
bSavingParams         : BOOL:=FALSE; (* *)
bLoadParams           : BOOL:=FALSE; (* *)
bHighPrecision        : BOOL;
bDisableTerminalCom   : BOOL;
bReset                : BOOL;
END_STRUCT
END_TYPE
```

Variable	Monitoring	Action	Control	Confi-guration	Auto-tuning	Description
AbsoluteHigh	x					An alarm is triggered if the actual zone temperature exceeds this limit value.
AbsoluteLow	x					An alarm is triggered if the actual zone temperature falls below this limit value.
ActTempGain				x		Scaling for the current temperature (for subsequent calibration)
ActTempOffset				x		Offset for the current temperature (for a subsequent calibration)
Autotune		x				A TRUE here activates the autotuning (automatic parameter determination) of the zone. In order to perform the autotuning successfully, the zone must be stable, InUse, Enable and be able to pass through at least a 40 °C temperature lift.
App_HmiType						Represents a numbering used only by the application, but stored by the library. Group formations (hot runner, cylinder1, cylinder2, etc.) are to be made possible via this.
bDisableTerminalCom		x				Disables terminal communication for this zone.
bHighPrecision			x			The control of the actual temperature is highly accurate. As a result, heating up may take more time.
bReset		x				Performs a reset in this zone.
bSavingParams		x				This indicates that machine data are being stored.
bLoadParams		x				A TRUE triggers the loading of machine parameters.
CJ_CompMode						Activates external compensation for thermocouples.
CJ_CompZone						Zone that measures the temperature to be compensated.
CoolerSwapIdx				x		Defines the output of fan switching signals via the redirectable I/O level.
dTmax					x	This is where the maximum rate of rise is recorded during autotuning. The determined value is displayed in °C/s.
Enable				x		A TRUE gives the enable for an active heating or cooling.

Variable	Monitoring	Action	Control	Confi-gura-tion	Auto-tun-ing	Description
EnableError Heating		x				A TRUE here activates the output of heating power when the temperature sensor is disturbed. As soon as a sensor error occurs, the temperature necessary to maintain the current temperature is output. This function is only fully functional if the Ena_TuneIdleLoad was successfully executed before.
EnaExtruder Block						reserved
Ena_TuneIdl eLoad		x				A TRUE activates the parameter determination for the "IdleLoad". In this optimization, the system is not excited in any way.
ErrorHeating Factor				x		This parameter influences the output of heating power when the temperature sensor is disturbed. It can take values between 0% and 100%, where 100% is the maximum power to keep the zone at the current temperature.
eTuningMet hod						eTcPfwTcTun_StepRespons e: Is the default autotune operation. The parameters are determined via a step response.
eTuningMet hod						eTcPfwTcTun_OscillationTes t: In preparation
ExtruderCo mp						This parameter compensates the friction and transport energy in a zone.
ExtruderCo mpEna						With a TRUE the extruder compensation is activated. When the path is switched on, the appropriate energy is automatically provided in the respective zone, minimizing controller settling. Speed changes in the range of 20% (related to the adjustment speed) are compensated without any problems, whereas in case of large speed changes (e.g. product change) a new compensation is necessary.
ExtruderId						reserved

Variable	Monitoring	Action	Control	Confi-gura-tion	Auto-tun-ing	Description
fc_Enable			x			Activation of a fluid forced cooling. In order not to overheat the fluid under the heating tape, the fluid must circulate at certain intervals. Activation automatically causes the fault to be calculated in the controller.
fc_OnTime			x			For this time (in seconds) the forced cooling is active (the cooling output is switched). If cooling is performed via the controller output during the "fc_OffTime" phase, this is taken into account.
fc_OffTime			x			For this time duration (in seconds) the forced cooling is inactive, but can be activated at any time via the controller output.
fPwmMinOnTime			x			This factor can be used to define the minimum PWM switch-on time in relation to the cycle time. A value between 0.1 and 1.0 must be entered.
fPwmMaxOnTime			x			This factor can be used to define the maximum PWM switch-on time in relation to the cycle time. A value between 0.05 and 0.75*PwmMaxOn must be entered. A PWMMaxOn time is assigned to each setpoint in the array. If null the PWMMaxOn Time from the SupplyLine is used.
fPwmStdMaxOnTime			x			This factor can be used to define the maximum PWM switch-on time in relation to the cycle time. A value between 0.05 and 0.75*PwmMaxOn must be entered. This variable is active when controlling to the setpoint. If null the PWMMaxOn Time from the SupplyLine is used.
HeaterSwapIdx				x		Defines the output of heating switching signals via the redirectable I/O level.
HibernateI_Heat						A TRUE causes the I-part of the heating control to freeze.
HibernateI_Cool						A TRUE causes the I-part of the cooling controller to freeze.

Variable	Monitoring	Action	Control	Confi-gura-tion	Auto-tun-ing	Description
InUse				x		The zone becomes an active part of the temperature control by TRUE. If FALSE, the zone will not be active even if Enable is set and the group is switched on. It will not signal a fault at any time or for any reason and will not be considered in load balancing or optional current measurement.
KpCool			x			The parameter for the P part of the temperature controller. **This parameter should be determined by autotuning.**
KpHeat			x			The parameter for the P part of the temperature controller. **This parameter should be determined by autotuning.**
L_LoadIdle			x			This parameter represents the base load. A properly set IdleLoad enables the set temperature to be reached without overshoot, as well as good system behavior during "error heating". **This parameter should be determined by an Idle-Tune.**
LooptestUpdate						reserved.
ModuleId				x		This parameter assigns a temperature group to the zone. A number of functions (e.g. switching on, lowering, etc.) are organized and controlled within a temperature group.
NoFanWhileTrackDown				x		If this parameter is set, an existing cooling is not used to reach the set temperature when the preset is reduced.
nPwmFactorC			x			The PWM cycle time is multiplied by this factor to realize an appropriate cycle time during cooling.
OutputSel_C				x		These parameters determine which of the offered signals are selected for heating.
OutputSel_H				x		These parameters determine which of the offered signals are selected for (optional) cooling.

Variable	Monitoring	Action	Control	Confi-gura-tion	Auto-tun-ing	Description
Overshoot					x	This is where the amount of overshoot is recorded during autotuning. This allows conclusions to be drawn about the dynamics of the system.
OpenloopHeating		x				If a zone has no sensor, the zone can be heated in a controlled manner via a fixed control value.
Ramping_Rate			x		x	Specifies the slope with which the controller setpoint should reach the entered setpoint during heating. The input is to be made in °C/min **This parameter should be determined by autotuning.**
Ramping_RateC			x		x	Specifies the slope with which the controller setpoint should reach the entered setpoint during cooling. The input is to be made in °C/min **This parameter should be determined by autotuning.**
Ramping_Tolerance			x			Specifies from when a setpoint change is to be increased via a ramp. It is recommended to approach setpoint changes of 5 to 10 °C via ramping. If the slope of the automatically determined ramp is too small, it can be changed via the Ramping_Rate parameter.
ReadBack						reserved.
SaveDelay				x		Time in ns. The value is counted down continuously. If the value reaches 0, saving is activated. -1 means idle state.
SensorOffset						An offset to be used when determining the actual temperature can be specified here. The specification has to be made in °C.
SensorType				x		These parameters determine which of the supported sensor types are used to record the actual temperature.
SettlingTime	x		x		x	This parameter is used in various places to take into account the time behavior of the zone.

Variable	Monitoring	Action	Control	Confi-gura-tion	Auto-tun-ing	Description
StartReTune					x	If the controller does not regulate ideally during operation, there is the possibility of subsequent self-optimization. This can be done during production.
SupplyId				x		This parameter assigns the zone to a supply line. It will take some parameters for the PWM output from this group. A range of functions (e.g. load balancing, current measurement etc.) is organized and synchronized within a supply line. By default, there are 4 different supply groups, with 1 to 3 to be used for phases 1 to 3. Two or three-phase heating tapes can be entered in supply line 4.
SupplyLoad_Cooler						The cooling capacity of the zone in watts.
SupplyLoad_Heater	x		x		x	The heating power of the zone in watts. In the case of (optional) monitoring of the heating power, this is the setpoint.
SupplyLoad_Tolerance	x					If the deviation of the measured heating power exceeds this tolerance, an alarm is triggered. The tolerance is specified in % between 0.0 and 100.0. If 0.0 is set here or the FB_TempCtrlMainBody_TcPfw() function block is called with Looptest_Enable:=FALSE, no monitoring takes place.
TdCool			x			The parameter for the D-part (damping time) of the temperature controller. **This parameter should be determined by autotuning.**
TdHeat			x			The parameter for the D-part (damping time) of the temperature controller. **This parameter should be determined by autotuning.**
TempSensTerm				x		These parameters determine which of the supported I/O terminals is used to acquire the actual temperature.

Variable	Monitoring	Action	Control	Confi-gura-tion	Auto-tun-ing	Description
TermChannel				x		For multi-channel terminals, the channel within the terminal must be specified here.
TermIdx				x		Defines the redirection of the actual temperature acquisition.
TnCool			x			The parameter for the I-part of the temperature controller. **This parameter should be determined by autotuning.**
TnHeat			x			The parameter for the I-part of the temperature controller. **This parameter should be determined by autotuning.**
Tracking_Td			x			This parameter is used for controlling of large set value changes. **This parameter should be determined by autotuning.**
TuneCooling					x	Only if a TRUE is entered here, the cooling behavior is evaluated during autotuning.
TuneEnd					x	This percentage of the temperature setpoint is used in autotuning.
TuneExtruderComp			x			A TRUE here calculates the ExtruderComp.
TuneHeaterLoad			x			A TRUE here calculates the heating power of this zone.
Tune_IdleLoad			x			A TRUE here calculates the IdleLoad of this zone during autotuning.
TuneKp					x	The P part of the autotuning mechanism. This factor is used to weight the P part in autotuning.
TuneTd					x	The D-part of the autotuning mechanism. This factor is used to weight the delay of the D-part in autotuning.
TuneTn					x	The I-part of the autotuning mechanism. This factor is used to weight the I-part in autotuning.
TuneTv					x	The D-part of the autotuning mechanism. This factor is used to weight the D-part in autotuning.

Variable	Monitoring	Action	Control	Confi-gura-tion	Auto-tun-ing	Description
TuneTrackingTd					x	The D-part of the autotuning mechanism. This factor is used to set the D-part of the Beckhoff algorithm. A value between zero and one means aggressive control; a value greater than one means cautious control. Default is zero.
TuneY					x	This percentage of the available heating power is used in autotuning.
TvCool			x			The parameter for the D-part (rate time) of the temperature controller. **This parameter should be determined by autotuning.**
TvHeat			x			The parameter for the D-part (rate time) of the temperature controller. **This parameter should be determined by autotuning.**
Update		x				With a TRUE the user interface signals here that it has changed values in this structure. The framework will check these values, adjust them if necessary and adopt them.
UseCooling				x		Only if a TRUE is entered here, the controller outputs to the cooling.
Weighting_C			x		x	Weighting factor, which is used to determine the cooling parameters for the control.
ZoneId				x		This parameter numbers the zones inside the machine. The numerical value may only be used in a single zone.
ZoneName				x		The textual name of the zone. Example: 'Ext_1' or 'Head_5'.

3.6.3 ST_TcPfw_TempPparamFromHmi_Itf

Such a structure contains the product data of a zone of temperature control.

Syntax

```
TYPE ST_TcPfw_TempPparamFromHmi_Itf :
(* last modification: 11.11.2008 *)
STRUCT
(*
see cnv_TempPparamFromHmi_TcPfw for format definition
Attention: HMI access via address
*)
Setpoint: LREAL;
StandbySetpoint: LREAL;
```

```
Setpoints       : ARRAY[cnv_TempCtrl_SetpointFirst..cnv_TempCtrl_SetpointLast] OF LREAL; (* AST: supp
orting selectable setpoints *)
Threshold_PP    : LREAL;
Threshold_P     : LREAL;
Threshold_M     : LREAL;
Threshold_MM    : LREAL;

Openloop_Output: LREAL;

SaveDelay: DINT:=-1;

Update          : BOOL;
bLoadParams     : BOOL:=FALSE;  (* *)
bSavingParams   : BOOL:=FALSE;  (* *)
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
Setpoint	LREAL	The temperature setpoint of the zone.
StandbySetpoint	LREAL	This value is used as the temperature setpoint by the zone in standby.
Setpoints	ARRAY OF LREAL	Additional setpoints can be specified here, which can be easily switched over via an index. The number of the index is to be specified in TempCtrl.SelectSetpoint.
Threshold_PP	LREAL	The outer positive tolerance limit.
Threshold_P	LREAL	The inner positive tolerance limit.
Threshold_M	LREAL	The inner negative tolerance limit.
Threshold_MM	LREAL	The outer negative tolerance limit.
Openloop_Output	LREAL	Control value output if no actual temperature is available.
SaveDelay	DINT	Memory delay in µs. After a time written to this variable by the application, the saving of the actual values is triggered. (If the value is zero, the system is saved; if it is -1, the system is at rest; if the value is greater than zero, the system is saved after the time has elapsed)
Update	BOOL	With a TRUE the user interface signals here that it has changed values in this structure. The framework will check these values, adjust them if necessary and adopt them.
bLoadParams	BOOL	This flag triggers the saving of the parameters.
bSavingParams	BOOL	Signals to the application that data are being saved.

3.6.4 ST_TcPfw_TempToHmi_Itf

Such a structure contains the visualization data of a zone of temperature control.

Syntax

```
TYPE ST_TcPfw_TempToHmi_Itf :
(* last modification: 25.09.2008 *)
STRUCT
ActualTemp          : LREAL;
SupplyMatch         : LREAL;
ActCurrent          : LREAL;
FileErrId           : DINT;
ErrorId             : UINT;
ModuleId            : INT;
PowerLevel          : INT;
ZoneId              : INT;
Cooling             : BOOL;
Enable              : BOOL;
Error               : BOOL;
FileErr             : BOOL;
Heating             : BOOL;
InUse               : BOOL;
OnStandBy           : BOOL;
```

```
TuningActive      : BOOL;
TuningDone        : BOOL;
IdleLoadActive    : BOOL;
IdleLoadDone      : BOOL;
LooptestActive    : BOOL;
ExtruderCompActive: BOOL;
ExtruderCompDone  : BOOL;
(*
*)
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
ActualTemp	LREAL	The actual temperature of the zone.
SupplyMatch	LREAL	Reflects the relationship between measured current power and the specified target power.
ActCurrent	LREAL	Actual current.
FileErrId	DINT	In case of storage/loading of the zone, a coded information is provided here. However, only if saving and loading are performed via the machine and product parameters.
ErrorId	UINT	In the event of an error, coded information is provided here. The conversion of the error number into a plain text can be seen in the global variables. Error numbers that are not listed there are usually general Beckhoff error numbers of subordinate function blocks (mostly ADS errors).
ModuleId	INT	This Id specifies the group to which this zone is assigned.
PowerLevel	INT	This value reflects the power value specified by the controller in %.
ZoneId	INT	This Id reflects the classification of the zone within its group.
Cooling	BOOL	A TRUE here indicates that the zone is actively cooling.
Enable	BOOL	A TRUE here indicates that the controller is enabled for the zone.
Error	BOOL	A TRUE here indicates that a controller, autotuning or hardware error has occurred in the zone.
FileErr	BOOL	A TRUE here signals that an error has occurred in the storage/loading case of the zone.
Heating	BOOL	A TRUE here indicates that the zone is actively heating.
InUse	BOOL	A TRUE here indicates that the zone is an active part of the current configuration. Prerequisite is that in the machine parameters ModuleId<>0; ZoneId<>0; SupplyId<>0 and InUse:=TRUE
OnStandBy	BOOL	A TRUE indicates here that the zone has been switched to the standby setpoint.
TuningActive	BOOL	During autotuning of the zone this signal is TRUE.
TuningDone	BOOL	A successful autotuning of the zone is reported here.
IdleLoadActive	BOOL	This signal is TRUE during IdleLoad tuning of the zone.
IdleLoadDone	BOOL	A successful IdleLoad tuning of the zone is reported here.
LooptestActive	BOOL	A TRUE here indicates that power measurement is active.
ExtruderCompActive	BOOL	Signals that the automatic calculation of extruder compensation is active.
ExtruderCompDone	BOOL	Signals that the automatic calculation of the extruder compensation has been successful.

3.6.5 ST_TcPfw_SupplyParam

This structure describes the parameters and runtime data of a supply unit.

Syntax

```
TYPE ST_TcPfw_SupplyParam :
(* last modification: 28.05.2008 *)
STRUCT
(*
see cnv_SupplyParam_TcPfw for format definition
*)

(*
temperature controller pwm setup
*)
fPwmCycleTime   : LREAL; (* will be updated to all temperature zones in supply group *)
fPwmMinOnTime   : LREAL;  (* will be updated to all temperature zones in supply group *)
fPwmMaxOnTime   : LREAL; (* will be updated to all temperature zones in supply group *)
fPwmMaxOnC      : LREAL;
fPwmMaxRampLoad : LREAL;        (* in kW, will be updated to all temperature zones in supply group *
)

fActSupplyLoad  : LREAL;
fActSupplyCurrent: LREAL;
fSupplyLoad     : LREAL;
fSupplyMatch    : LREAL;

nPwmFactorC     : INT:=1; (* will be updated to all temperature zones in supply group *)

(*
internal
*)
fUsedLoad       : LREAL;
fUsedLoad_H     : LREAL;
fUsedLoad_C     : LREAL;
tpwmtimer       : LREAL:=0.0;

SaveDelay       : DINT:=-1;
FileErrId       : DINT;


refresh_H       : BOOL;

Unsaved         : BOOL;
bSavingParams   : BOOL:=FALSE; (* *)
bLoadParams     : BOOL:=FALSE; (* *)
FileErr         : BOOL:=FALSE; (* *)
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
fPwmCycleTime	LREAL	The cycle time (in seconds) of the PWM signal generator.
fPwmMinOnTime	LREAL	The minimum switch-on component of the PWM signal. Range 0.1 to 1.0
fPwmMaxOnTime	LREAL	The maximum switch-on time from the PWM signal. Range 0.1 to 1.0
fPwmMaxOnC	LREAL	The maximum switch-on time of the cooling output from the PWM signal. If zero, the fPwmMaxOnTime is used. Range 0.1 to 1.0
fPwmMaxRampLoad	LREAL	reserved
fActSupplyLoad	LREAL	If a power measurement is performed, the current power can be read here.
fActSupplyCurrent	LREAL	If a power measurement is performed, the current currently measured can be read here.
fSupplyLoad	LREAL	The predicted total power of the supply line.
fSupplyMatch	LREAL	The ratio of fActSupplyLoad to fSupplyLoad.
nPwmFactorC	INT	The multiplier for the cooling PWM cycle time. Tpwm_cool := Tpwm_heat * nPwmFactorC.
fUsedLoad	LREAL	reserved
fUsedLoad_H	LREAL	reserved
fUsedLoad_C	LREAL	reserved
tpwmtimer	LREAL	Counter for the PWM cycles of the heating control.
SaveDelay	DINT	Memory delay in µs. After a time written to this variable by the application, the saving is triggered. (If the value is zero, the system is saved; if it is -1, the system is at rest; if the value is greater than zero, the system is saved after the time has elapsed)
FileErrId	DINT	In case of storage/loading of the zone, a coded information is provided here. However, only if saving and loading are performed via the machine and product parameters.
refresh_H	BOOL	Trigger signals for heating and cooling PWM cycles.
Unsaved	BOOL	Signals the application that parameters have changed from the library.
bSavingParams	BOOL	Signals to the application that data are being saved.
bLoadParams	BOOL	This flag triggers the saving of the parameters.
FileErr	BOOL	In case of storage/charging of the zone, it is displayed here if the storage or charging process has failed.

The controller output is usually provided to the heating tape as a PWM signal. It is possible to provide a separate PWM configuration for each phase of a supply line.

3.6.6 ST_TcPfw_TempCtrl_Itf

Such a structure contains the visualization data of a zone of temperature control.

Syntax

```
TYPE ST_TcPfw_TempCtrl_Itf :
(* last modification: 01.10.2010 *)
STRUCT
(*
==========================================
temperature zone internal data
see cnv_TempCtrl_Itf_TcPfw for format definition
==========================================
*)
stRtData: ST_TcPfw_TempCtrl_RtData;

ZoneName: STRING(79);

Heater_SupplyLoad  : LREAL;
```

```
Setpoint           : LREAL;
Threshold_PP       : LREAL;
Threshold_P        : LREAL;
Threshold_M        : LREAL;
Threshold_MM       : LREAL;
StandbySetpoint    : LREAL;

tempEnergy         : LREAL;
Openloop_Output    : LREAL;

PrevSameSupply     : INT;
NextSameSupply     : INT;
EvtIdx_Autotune    : INT;
EvtIdx_Hardware    : INT;
TempTermInit       : INT;
SelectSetpoint     : INT:=0; (* AST: supporting selectable setpoints *)

SelSetpoint        : BOOL;
Cmd_TuneHeaterLoad : BOOL;
Sema_Update        : BOOL;
Sema_Used          : BOOL;
LoopTest           : BOOL;
LoopTest_Inv       : BOOL;
Enable             : BOOL;
Alarm_LowLow       : BOOL;
Alarm_Low          : BOOL;
Alarm_High         : BOOL;
Alarm_HighHigh     : BOOL;
Alarm_AbsoluteLow  : BOOL;
Alarm_AbsoluteHigh : BOOL;
Alarm_NoResponse   : BOOL;
Force_Heating      : BOOL:=FALSE;
Force_Cooling      : BOOL:=FALSE;

Fault              : BOOL:=FALSE; (* starting with V1.0.9 *)
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
stRtData	ST_TcPfw_TempCtrl_RtData	The runtime data of the zone.
ZoneName	STRING	Reserved, not guaranteed.
Heater_SupplyLoad	LREAL	The total heating power of all zones of the same supply line.
Setpoint	LREAL	The set temperature of the zone. (Copy of the parameter from ST_TcPfw_TempPparamFromHmi_Itf).
Threshold_PP	LREAL	The outer positive tolerance limit of the zone. (Copy of the parameter from ST_TcPfw_TempPparamFromHmi_Itf).
Threshold_P	LREAL	The inner positive tolerance limit of the zone. (Copy of the parameter from ST_TcPfw_TempPparamFromHmi_Itf).
Threshold_M	LREAL	The inner negative tolerance limit of the zone. (Copy of the parameter from ST_TcPfw_TempPparamFromHmi_Itf).
Threshold_MM	LREAL	The outer negative tolerance limit of the zone. (Copy of the parameter from ST_TcPfw_TempPparamFromHmi_Itf).
StandbySetpoint	LREAL	The set lowering temperature of the zone. (Copy of the parameter from ST_TcPfw_TempPparamFromHmi_Itf).
tempEnergy	LREAL	Intermediate variable for energy calculation.
Openloop_Output	LREAL	Copy of the specified control rate from the product parameters.
PrevSameSupply	INT	Reserved for internal use.
NextSameSupply	INT	Reserved for internal use.
EvtIdx_Autotune	INT	Reserved for Blow Molding Framework.
EvtIdx_Hardware	INT	Reserved for Blow Molding Framework.
TempTermInit	INT	Reserved for the FB_TempCtrlCallback_TcPfw() function block.
SelectSetpoint	INT	Selection of the setpoint from the Setpoints array. If the value is outside the value range, Setpoint from the product parameters is active.
SelSetpoint	BOOL	The switching of the effective setpoint of the zone. A TRUE selects the StandbySetpoint, a FALSE the Setpoint.
Cmd_TuneHeaterLoad	BOOL	Reserved for automatic measurement of heating power.
Sema_Update	BOOL	reserved
Sema_Used	BOOL	reserved
LoopTest	BOOL	The current measurement for this zone is active. The heating is switched on for a short time independently of the control.
LoopTest_Inv	BOOL	Current measurement for another zone of the same supply line is active. The heating is switched off for a short time independently of the control.
Enable	BOOL	A TRUE here indicates that the controller is enabled for the zone.
Alarm_LowLow	BOOL	The zone is ST_TcPfw_TempMparamFromHmi_Itf.InUse and the actual temperature of the zone is below the effective setpoint by more than the outer negative tolerance limit.
Alarm_Low	BOOL	The zone is ST_TcPfw_TempMparamFromHmi_Itf.InUse and the actual temperature of the zone is below the effective setpoint by more than the inner negative tolerance limit.

Name	Type	Description
Alarm_High	BOOL	The zone is ST_TcPfw_TempMparamFromHmi_Itf.InUse and the actual temperature of the zone is more than the inner positive tolerance limit above the effective setpoint.
Alarm_HighHigh	BOOL	The zone is ST_TcPfw_TempMparamFromHmi_Itf.InUse and the actual temperature of the zone is more than the outer positive tolerance limit above the effective setpoint.
Alarm_AbsoluteLow	BOOL	The zone is ST_TcPfw_TempMparamFromHmi_Itf.InUse and the actual temperature of the zone is below ST_TcPfw_TempMparamFromHmi_Itf.AbsoluteLow.
Alarm_AbsoluteHigh	BOOL	The zone is ST_TcPfw_TempMparamFromHmi_Itf.InUse and the actual temperature of the zone is above ST_TcPfw_TempMparamFromHmi_Itf.AbsoluteHigh. If this flag is set, no heating power is generated in this zone.
Alarm_NoResponse	BOOL	The actual temperature of the zone has not responded to the heating power within a reasonable time.
Force_Heating	BOOL	A TRUE produces a heating power output at 100% for 100 cycles.
Force_Cooling	BOOL	A TRUE will produce a cooling output at 100% for 100 cycles.
Fault	BOOL	Error.

3.6.7 ST_TcPfw_PowerMeasurement_Cfg

Such a structure contains the configuration data for power measurement. These data are not stored.

Syntax

```
TYPE ST_TcPfw_xL3403_Input:
(* location PfwLib_TempControl.PRO *)
(* last modification: 08.09.2010 *)
STRUCT
    CycleTime      : LREAL:=0.025;
    I_ratio        : LREAL:=1.0;

    ePowerTerminal: E_TcPfw_TctrlPowerTerminal;
    TerminalIdx    : INT;
    TerminalSubIdx: INT;
END_TYPE
```

Parameter

Name	Type	Description
CycleTime	LREAL	Task cycle time.
I_ratio	LREAL	Ratio of the current transformers.
ePowerTerminal	E_TcPfw_TctrlPowerTerminal	Type of power measurement terminal that is connected.
TerminalIdx	INT	Number of the Supplygroup which is assigned to it.
TerminalSubIdx	INT	For terminals containing more than three current measurement channels (EL3446), it must be specified whether the group is connected to the first three current measurement inputs (-> 1) or to the rear three current measurement inputs (-> 2).

3.7 Knowledge Base

3.7.1 Commissioning

3.7.1.1 Global

Commissioning

The PfwLib_TempControl.lib library is divided into six main structures.

- ST_TcPfw_TempCtrlInput
  This structure contains all linkable input variables.
- ST_TcPfw_TempCtrlOutput
  This structure contains all linkable output variables.
- ST_TcPfw_TempMparamFromHmi_Itf
  All machine parameters are stored in this structure.
- ST_TcPfw_TempPparamFromHmi_Itf
  All product parameters are stored in this structure.
- ST_TcPfw_SupplyParam
  Settings for PWM output are made in this structure.
- ST_TcPfw_TempToHmi_Itf
  This structure shows the current states of the temperature controller.

For successful commissioning of the temperature controller, the PfwLib_TempControl.lib library must be included in the project. The following steps must then be carried out:

Creating an instance of the function block FB_TempCtrlMainBody_TcPfw and declaring the variables

Variable	Short description	Example value
ConfigEnable	Signals the validity of the parameters.	TRUE
Callback_Enable	Enables background communication with the I/O terminals.	TRUE
Looptest_Enable	Activates the power monitoring of the heating tapes.	FALSE
tCycle	Cycle time (A cycle time unequal to the mains frequency is recommended).	0.025s
Simu_Enable	Starting a simulation.	FALSE
Simu_DisCharge	Reset simulation to initial value.	FALSE
Simu_DisCharge	Reset simulation to initial value.	FALSE

The temperature library contains two simulations:

- A Pt2 path that is active when the aaaPfwTempMparamFromHmi[..].TempSensTerm parameter contains a 0 (recommended simulation for the user).
- If the parameter **Simu_Enable** is set, a physical model of an extruder cylinder becomes active. This simulation is mainly intended for development purposes and should not be activated by the user.

Constant definition

Variable	Short description	Type	Example value
cnPfwTempCtrlFirst	Number of the first zone (usually 1).	INT	1
cnPfwTempCtrlLast	Number of the last zone.	INT	32
cnPfwAppSupplyFirst	Number of the first supply group.	INT	1
cnPfwAppSupplyLast	Number of the last supply group.	INT	3
cnPfwTempTrendFirst	cnPfwTempTrendFirst and cnPfwTempTrendLast specify the number of samples for trend acquisition.	INT	1
cnPfwTempTrendLast	cnPfwTempTrendFirst and cnPfwTempTrendLast specify the number of samples for trend acquisition.	INT	100
cnPfwTempTrend_sPH	Period value of the sampling time in ms.	INT	
cnPfwBoolOutSwapFirst	Initial index of the array out_SwappedDigitalOut.	INT	1
cnPfwBoolOutSwapLast	End index of the array out_SwappedDigitalOut.	INT	2 (When not in use)
cnPfwBoolInSwapFirst	Initial index of the array in_SwappedDigitalIn.	INT	1
cnPfwBoolInSwapLast	End index of the array in_SwappedDigitalIn.	INT	2 (When not in use)
cnPfwScopeSampleFirst	cnPfwScopeSampleFirst and cnPfwScopeSampleLast specify the number of samples for scope acquisition.	INT	
cnPfwScopeSampleLast	cnPfwScopeSampleFirst and cnPfwScopeSampleLast specify the number of samples for scope acquisition.	INT	
bPfw_UseTempControl	The feedback is active and thus the reading of actual values is possible.	BOOL	TRUE
bPfw_UseRtScope	The acquisition of scope data is active.	BOOL	FALSE
bPfw_UseTempTrend	The acquisition of trend data is active.	BOOL	FALSE
cnst_pfw_selRelAlarm	Via a TRUE, the selected setpoint is used for the relative alarms. With a FALSE the internally ramped setpoint is used.	BOOL	FALSE
cnPfwLoopTestTimer	Time in ms in which the power measurement measures.	LREAL	0.4 s
cnPfwLoopTestCycle	Time in ms which is available for the controller before the next zone is checked.	LREAL	9.999 s
cnst_PfwParamFilePath_CE	Folder path on a CE operating system.	STRING	
cnst_PfwParamFilePath_XP	Folder path on an XP/ Win7 operating system.	STRING	
cnst_PfwSubDir_Logging	Subfolder for log files.	STRING	
cnst_PfwSubDir_Product	Subfolder for product files.	STRING	
cnst_PfwSubDir_Machine	Subfolder for machine files.	STRING	
cnst_PfwSubDir_Supply	Subfolder for supply files.	STRING	
bPfw_UseEnergyRecording	Activation of a theoretical energy acquisition.	BOOL	FALSE
bPfwRunOn_WinCE	A TRUE activates the folder path for a CE operating system.	BOOL	TRUE

Please note that the constants of the timer as well as the alarm visualization must also be taken into account.

A heating zone in this library is an actuator (Solid State Relay - SSR or contactor), one or more heating tapes and a temperature sensor. If a calculated temperature is regarded as an actual value, this must be done in the application.

Sample project

The loading and saving of product and machine parameters as well as the general handling of the temperature library will be explained in more detail using the sample project. To start the application it is necessary to include the PfwLib_TempControl.lib and to adjust the path for saving and loading the parameters.

3.7.1.2 Application

Commissioning (parameterization)

The following table lists all parameters necessary for initializing the temperature controller.

After changes have been made to the parameters of the aaaPfwTempMparamFromHmi, aaaPfwTempPparamFromHmi, stPfwSupplyLineCfg structures, they must be saved via the respective structure. For this purpose, the following variables are provided in each structure:

- **SaveDelay:** If a number in ms is entered here, the library counts from this number to zero. At zero, the file is saved and at -1, saving is inactive.
- **SaveParam:** Triggers immediate saving.
- **LoadParam:** Loads the parameters from the previously defined path.

During saving, a backup file is always created, so that the validity of the data is ensured at all times.

In the structure aaaPfwTempMparamFromHmi[...] the machine data are initialized:

Variable	Short description	Example value	Type
ZoneName	Name of the zone	Z1	STRING
AbsoluteHigh	Temperature upper limit for alarm and shutdown	250 °C	LREAL
AbsoluteLow	Temperature lower limit for alarm and shutdown	40 °C	LREAL
ExtruderComp	TRUE: active; FALSE: inactive	FALSE	BOOL
KpCool	P gain from PID controller (heating)	... %/°C	LREAL
KpHeat	P gain from PID controller (cooling)	.. %/°C	LREAL
TdCool	The D-part (damping time) of the temperature controller (cooling)	... s	LREAL
TdHeat	The D-part (damping time) of the temperature controller (heating)	... s	LREAL
TnCool	Integral action time from PID controller (heating)	... s	LREAL
TnHeat	Integral action time from PID controller (cooling)	... s	LREAL
TvCool	The D-part (rate time) of the temperature controller (cooling)	... s	LREAL
TvHeat	The D-part (rate time) of the temperature controller (heating)	... s	LREAL
Overshoot	Overshoot after self-optimization	... °C	LREAL
Tracking_Td	Time constant for control with large setpoint changes	... s	LREAL
Ramping_Rate	Ramp slope in °C/min	... °C/min	LREAL
Ramping_Tolerance	Up to which setpoint step-change should a ramp be used?	5 °C	LREAL
dTmax	max. slope of the path for a step response	... °C/s	LREAL
SensorOffset	Offset temperature from temperature sensor	0 °C	LREAL
SettlingTime	path-specific time base	5 s	LREAL
SupplyLoad_Cooler	Cooling capacity	100 W	LREAL
SupplyLoad_Heater	Heating capacity	1000 W	LREAL
SupplyLoad_Tolerance	If the measured heating power deviates from this tolerance, an alarm is triggered.		LREAL
TuneEnd	Final temperature of self-tuning in relation to setpoint [%]	80%	LREAL
TuneKp	Setting factor for the Kp value	01. Feb	LREAL
TuneTd	Setting factor for the Td value	0.2	LREAL
TuneTn	Setting factor for the Tn value	2	LREAL
TuneTv	Setting factor for the Tv value	0.5	LREAL
TuneY	Power output during self-tuning	100%	LREAL
TuneTrackingTd	Setting factor for the TrackingTd value	0	LREAL
L_LoadIdle	Discharge resistance to the environment	2 W/°C	LREAL
ErrorHeatingFactor	Heating factor with control switched off	20 %	LREAL
fPwmStdMaxOnTime	Maximum heating on time when the setpoint is selected. At zero the MaxPWMOnTime of the SupplyLine is used. Value range 0.0...1.0	0.0	LREAL
fPwmMaxOnTime	Maximum heating on-time when one of the setpoints in the array is selected. At zero the MaxPWMOnTime of the SupplyLine is used. Value range 0.0...1.0	20 %	LREAL
fPwmMinOnTime	Minimum heating on time when the setpoint is selected. At zero the PwmMinOnTime of the SupplyLine is used. Value range 0.0...1.0	20 %	LREAL

Variable	Short description	Example value	Type
Weighting_C	Factor between the determined heating and cooling parameters		LREAL
fc_OnTime	Switch-on time of forced cooling		LREAL
fc_OffTime	Duration for normal regulation		LREAL
OutputSel_H	Heating signal selection (see E_TcPfw_TctrlOutSelect)	1	INT
OutputSel_C	Cooling signal selection (see E_TcPfw_TctrlOutSelect)	3	INT
TempSensTerm	Sensor terminal type must be selected by E_TcPfw_TerminalType.		
SensorType	Type of sensor must be selected by E_TcPfw_TempSensType.		
TermChannel	Sensor terminal channel	1	INT
ExtruderId	Extruder number	1	INT
ModuleId	Assigns a temperature group to zones	1	INT
ZoneId	Numbering of the zone of a machine	1	INT
SupplyId	Supply group to distribute the power evenly among the three phases	2	INT
CJ_CompMode			
CJ_CompZone			
SensTermSwapIdx			
HeaterSwapIdx			
CoolerSwapIdx			
InUse	TRUE: zone in use; FALSE: zone not in use The zone is only active in case of feedback from ST_TcPfw_TempToHmi_Itf[x].InUse. Requirement: ModuleId<>0; ZoneId<>0; SupplyId<>0	TRUE	
UseCooling	Zone has a cooling.	TRUE	BOOL
ExtruderCompEna	Activation of extruder compensation	FALSE	BOOL
TuneCooling	Self-optimization for cooling	FALSE	BOOL
Tune_IdleLoad	Self-optimization for IdleLoad determination. This would follow the heating power determination.	TRUE	BOOL
Autotune	Self-optimization for heating	FALSE	BOOL
Enable	Zone is active	TRUE	BOOL
Update	The user interface indicates that values in this structure have been changed.	FALSE	BOOL
EnaExtruderBlock	reserved	FALSE	BOOL
NoFanWhileTrackDown	Do not use a fan when cooling down.	FALSE	BOOL
Ena_TuneIdleLoad	Activate calculation of the discharge resistance.	FALSE	BOOL
LooptestUpdate	Activate power measurement in this zone.	FALSE	BOOL
EnableErrorHeating	Activate that the heating tape is supplied with a predefined power (ErrorHeatingFactor).	FALSE	BOOL
NoFanWhileTrackDown	reserved	FALSE	BOOL
Ena_TuneIdleLoad	reserved	FALSE	BOOL
OpenloopHeating	Set output with a predefined output	FALSE	BOOL
fc_Enable	Activation of forced cooling	FALSE	BOOL
HibernateI_Cool	Freezing of the I-part for heating	FALSE	BOOL
hibernateI_Heat	Freezing the I-part for cooling	FALSE	BOOL

Variable	Short description	Example value	Type
bSavingParams	Signal that machine data are currently being stored.		BOOL
bLoadParams	Loads the machine data	FALSE	BOOL
ReadBack	reserved	FALSE	BOOL

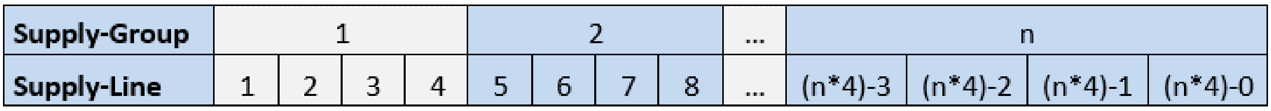
In the structure aaaPfwTempPparamFromHmi[...] the product data are initialized:

Variable	Short description	Example value	Type	Note
Setpoint	Setpoint during operation	180 °C	LREAL	must be parameterized
StandbySetpoint	Setpoint in standby	60 °C	LREAL	must be parameterized
Threshold_M	The inner negative tolerance limit, referred to the setpoint (threshold for the on-off controller)	-5 °C	LREAL	must be parameterized
Threshold_MM	The outer negative tolerance limit, referred to the setpoint	-10 °C	LREAL	must be parameterized
Threshold_P	The inner positive tolerance limit, referred to the setpoint (threshold for the on-off controller)	+5 °C	LREAL	must be parameterized
Threshold_PP	The outer positive tolerance limit, referred to the setpoint	+10 °C	LREAL	must be parameterized
Update	The user interface indicates that values in this structure have been changed, relative to the setpoint.		BOOL	optional
Variable	Short description	Example value	Type	Note

In the structure stPfwSupplyLineCfg[...] the settings for the PWM are initialized and the connection between a zone and the supply network is established.

This information is only important for heating power monitoring and zooning. If both are not used, it is sufficient to initialize the array with one element.

Plastics machines are sometimes designed to contain several temperature control groups (extruders1..n, heating channels, etc.). These are partly supplied separately. These supply units are called supply groups in the framework. Each supply group consists of four supply lines (phase 1..3 and multi-phase elements) . Several heating zones can be connected to one supply line. The first supply line of a group represents the supply for the heating zones, which are connected between L1 and the neutral conductor. The second for L2 and N and the third for L3 and N. The fourth supply line includes all heating zones that are connected between two outer conductors in star or delta.



Supply-Group	1				2				...	n			
Supply-Line	1	2	3	4	5	6	7	8	...	(n*4)-3	(n*4)-2	(n*4)-1	(n*4)-0

Variable	Short description	Example value		Parameterization level
fPwmCycleTime	PWM cycle time	0.1 s	LREAL	yes
fPwmMaxOnTime	PWM maximum switch-on time (related to cycle time)	0.9 for 90%	LREAL	yes
fPwmMinOnTime	PWM minimum switch-on time (related to cycle time)	0.1 for 10%	LREAL	yes
fPwmMaxOnC	PWM maximum switch-on time (related to the cycle time) for cooling	0.1 for 10%	LREAL	yes
fPwmMaxRampLoad	reserved		LREAL	optional
fActSupplyLoad	If a power measurement is performed, the current power can be read here.	... W	LREAL	optional
fActSupplyCurrent	If a power measurement is performed, the current currently measured can be read here.	... W	LREAL	optional
fSupplyLoad	The predicted total power of the supply group.	... W	LREAL	optional
fSupplyMatch	The ratio of fActSupplyLoad to fSupplyLoad		LREAL	optional
nPwmFactorC	For PWM cooling, nPwmFactorC is multiplied by "fPwmCycleTime" (reason: cooling is often controlled via contactors).	1	INT	must be parameterized

In the structure aaaPfwTempToHmi[...] the following display values are written back to the HMI:

Variable	Short description	
ActualTemp	Current temperature of the zone	LREAL
SupplyMatch	Current measured power ratio (actual power/target power)	LREAL
ActCurrent	Current measured from the zone	LREAL
FileErrId	Error number if loading or saving of the machines or product data fails.	DINT
ErrorId	Error number	WORD
ModuleId	Module number	INT
PowerLevel	Heating power output of the controller	LREAL
ZoneId	Zone number	INT
Cooling	The zone is cooling.	BOOL
Enable	The zone controls to its setpoint.	BOOL
Error	The zone is heating.	BOOL
Heating	The zone is in error state.	BOOL
FileErr	The zone could not successfully load or save the machine or product parameters.	BOOL
InUse	Zone is used. In the machine parameters must be: ModuleId<>0; ZoneId<>0; SupplyId<>0 and InUse:=TRUE	BOOL
OnStandBy	Standby temperature is active as setpoint.	BOOL
TuningActive	Self-optimization active	BOOL
TuningDone	Self-optimization successful	BOOL
IdleLoadActive	Calculation of the discharge resistance active	BOOL
IdleLoadDone	Calculation of the discharge resistance successful	BOOL
LooptestActive	Activation of the power control of the individual heating tapes	BOOL
ExtruderCompActive	Feedback that the extruder compensation is being taught.	BOOL
ExtruderCompDone	Feedback that the extruder compensation was successfully taught.	BOOL

3.7.1.3 Mapping

System Manager

The following variables must be linked for each temperature sensor:

- EL terminal (EL33xx):

Terminal variable	PLC variable	Comment
Value	in_PfwTempCtrlInput[...].EL_Sns_Data	Process value
Underrange	in_PfwTempCtrlInput[...].EL_SnsUnderrun	The temperature has fallen below the lower temperature range of the sensor.
Overrange	in_PfwTempCtrlInput[...].EL_SnsOverrun	The upper temperature range of the sensor has been exceeded.
Error	in_PfwTempCtrlInput[...].EL_SnsError	The channel of a terminal supplies an error.
WcState	in_PfwTempCtrlInput[...].EL_SnsWcState	If the value is 0, the terminal data is valid; if the value is 1, the data is invalid. The terminal variable must be linked to each PLC channel used by this terminal (multiple linking).
AdsAddr	in_PfwTempCtrlInput[...].EL_AdsAddr	Address for communication with the terminal. The terminal variable must be linked to each PLC channel used by this terminal (multiple linking).
State	in_PfwTempCtrlInput[...].EL_SnsState	General terminal status The terminal variable must be linked to each PLC channel used by this terminal (multiple linking).

- KL terminal (KL33xx):

Terminal variable	PLC variable	Description
Data In	in_PfwTempCtrlInput[...].KL_SnsData	Data input
State	in_PfwTempCtrlInput[...].KL_SnsState	Terminal state
Data out	out_PfwTempCtrlOutput[...].KL_SnsData	Data output
Ctrl	out_PfwTempCtrlOutput[...].KL_SnsCtrl	To register communication

- Via aaaPfwTempMparamFromHmi[..].OutputSel_C a cooling signal is selected in the PLC, which must be linked to the System Manager by out_PfwTempCtrlOutput[...].SelOutNeg.
- Via aaaPfwTempMparamFromHmi[..].OutputSel_H a heating signal is selected in the PLC, which must be linked to the System Manager by out_PfwTempCtrlOutput[...].SelOutPos.

3.7.1.4 Self-optimization

After the library is included and all parameters are predefined, a self-optimization should be performed. The following parameters are relevant for this purpose:

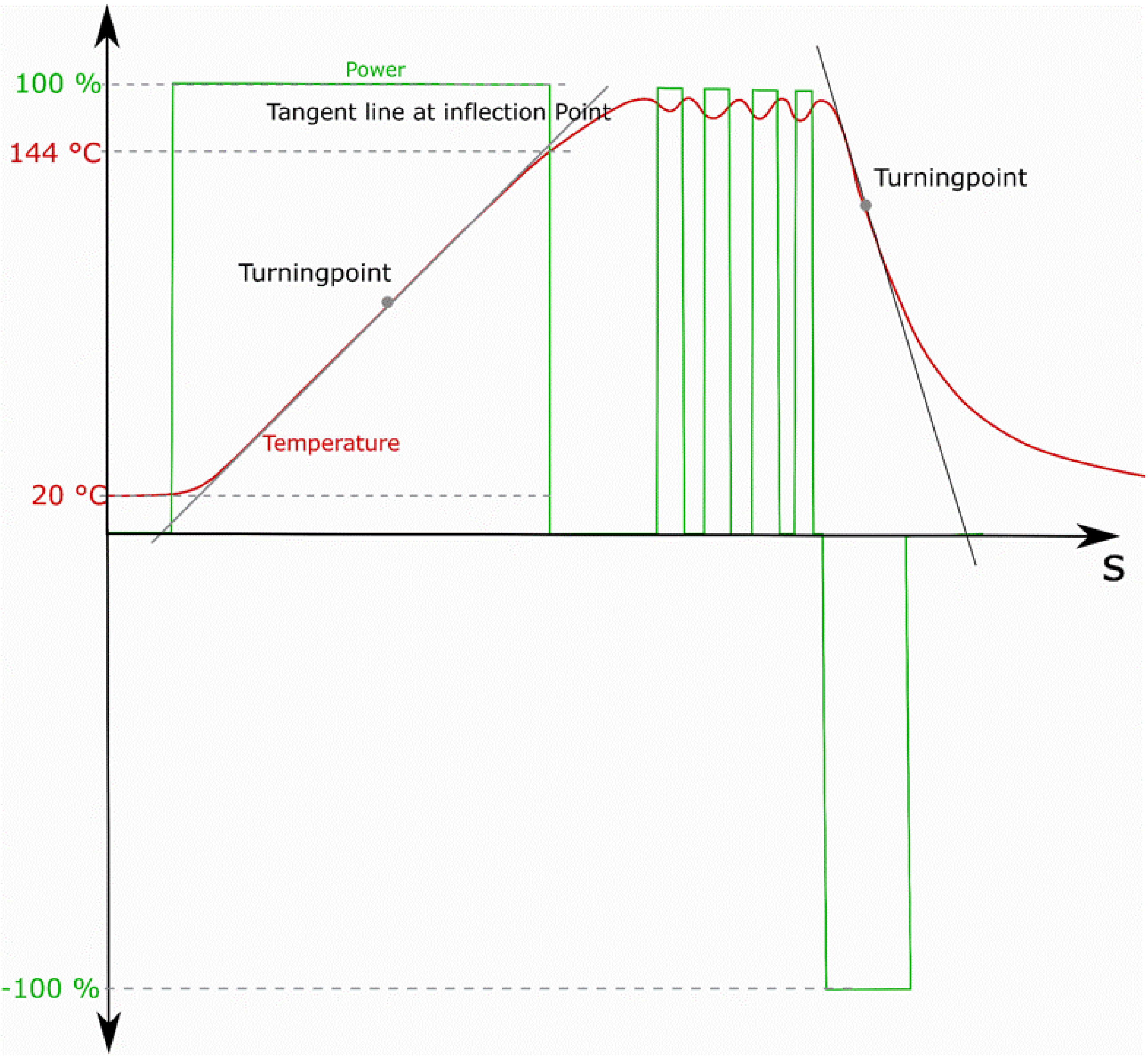
- **TuneCooling:** If the zone has cooling (UseCooling must be TRUE and the output type must be set OutputSel_H), separate parameters can be determined for the cooling.
- **Tune_IdleLoad:** A TRUE triggers the determination of the pre-control part during self-optimization.
- **TuneY:** Specifies the amount of heating power output during self-optimization.
- **TuneEnd:** After TuneEnd/100* setpoint the output is switched off and the overshoot is determined. It should be guaranteed that after overshooting the actual temperature is below the set temperature.

Self-optimization is started via the signal ST_TcPfw_TempMparamFromHmi_Itf[...].Autotune. Then ST_TcPfw_TempToHmi_Itf[...].TuningActive becomes TRUE. If this is not the case, it was aborted with error. After the autotune was successful, this is indicated in ST_TcPfw_TempToHmi_Itf[...].TuningDone via a TRUE. Self-optimization can be aborted at any time by setting ST_TcPfw_TempMparamFromHmi_Itf[...].Autotune to FALSE.

Prerequisite for successful self-optimization is that the controller is active (ST_TcPfw_TempMparamFromHmi_Itf[...].Enable) and has stabilized to standby temperature.

It is recommended that the actual temperature is close to the ambient temperature. The difference between the current temperature and the operating point temperature should be at least 40.0 °C.

After activating the autotune, the temperature is monitored for a certain time to check if it is stable. Subsequently, a control value in the amount of TuneY is output. If the actual temperature cuts ST_TcPfw_TempMparamFromHmi_Itf[...].TuneEnd*ST_TcPfw_TempPparamFromHmi_Itf[...].Setpoint, the setpoint output is switched off and the overshoot is observed.



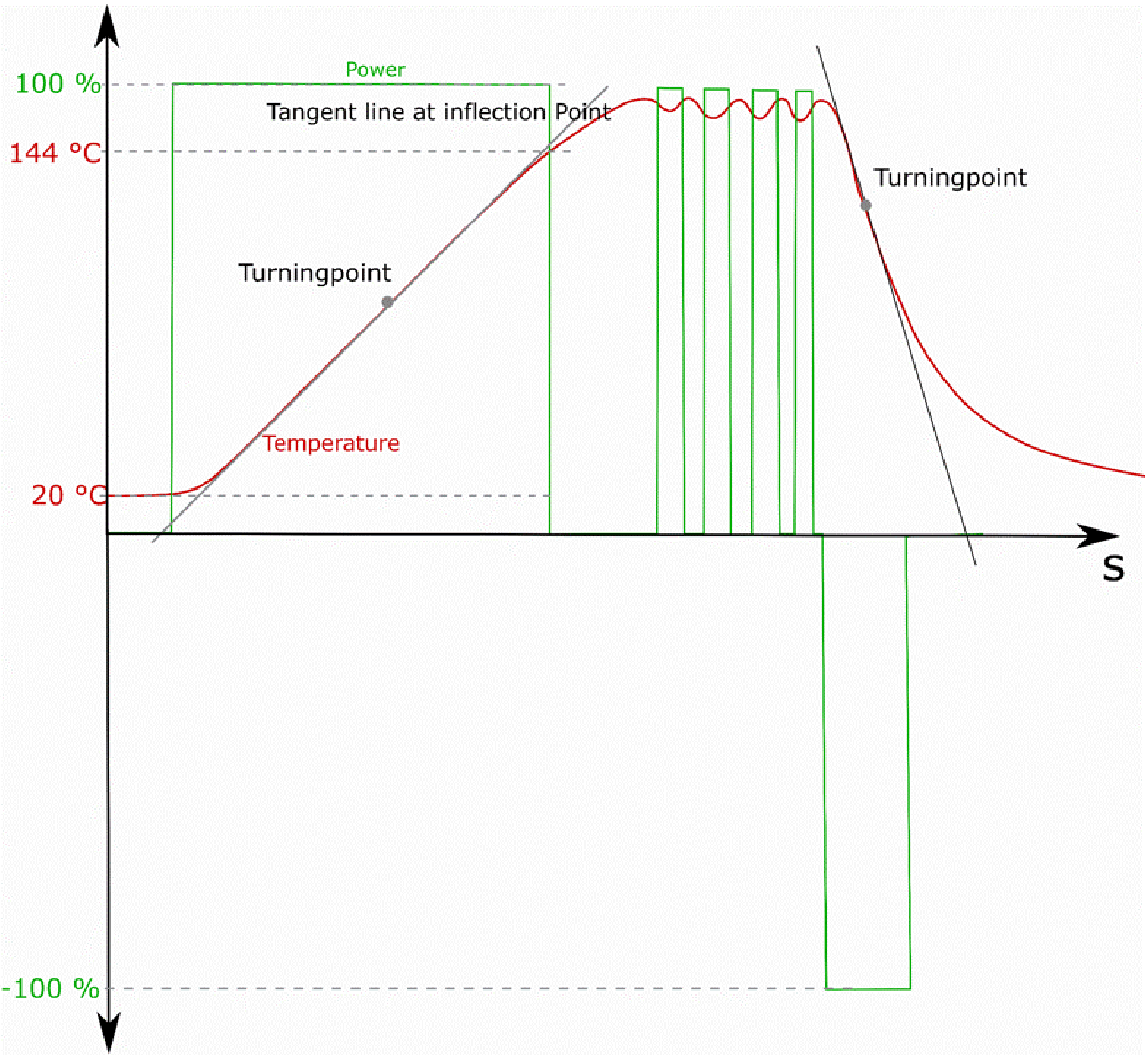
3.7.2 Alarm handling

Overview of the Plastics Processing Framework alarms

The following alarm conditions are continuously monitored by the temperature controller:

- Absolute temperatures (high and low):
  - the actual temperature is below aaaPfwTempMparamFromHmi[...].AbsoluteLow, the global flag aaaTempAlarm_AbsoluteLow is set to TRUE. As soon as the actual temperature exceeds the threshold, aaaTempAlarm_AbsoluteLow becomes FALSE again.

  - if the actual temperature is above aaaPfwTempMparamFromHmi[...].AbsoluteHigh, the global flag aaaTempAlarm_AbsoluteHigh is set to TRUE. As soon as the actual temperature falls below the threshold, aaaTempAlarm_AbsoluteHigh becomes FALSE again. As soon as aaaTempAlarm_AbsoluteHigh is active, the corresponding zone is switched off. After falling below the limit, the zone returns to regulated operation.
- Relative temperatures (in two bands around the setpoint):
  - if the actual temperature is below aaaPfwTempPparamFromHmi[...].Threshold_MM, the global flag aaaTempAlarm_LowLow is set to TRUE. As soon as the actual temperature exceeds the threshold, aaaTempAlarm_LowLow becomes FALSE again.
  - if the actual temperature is below aaaPfwTempPparamFromHmi[...].Threshold_M, the global flag aaaTempAlarm_Low is set to TRUE. As soon as the actual temperature exceeds the threshold, aaaTempAlarm_Low becomes FALSE again.
  - if the actual temperature is above aaaPfwTempPparamFromHmi[...].Threshold_PP, the global flag aaaTempAlarm_HighHigh is set to TRUE. As soon as the actual temperature falls below the threshold, aaaTempAlarm_HighHigh becomes FALSE again.
  - if the actual temperature is above aaaPfwTempPparamFromHmi[...].Threshold_P, the global flag aaaTempAlarm_High is set to TRUE. As soon as the actual temperature falls below the threshold, aaaTempAlarm_High becomes FALSE again.
- If the constant cnst_pfw_selRelAlarm is TRUE, the relative alarms refer to the entered set temperature, otherwise to the internally ramped set temperature.



- Error messages from the terminal:

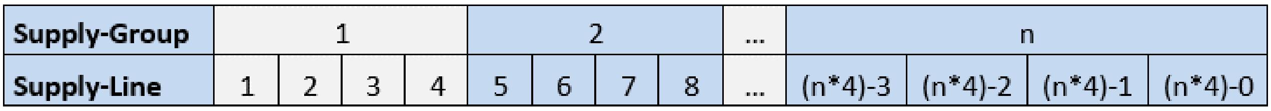
  - Are passed on via ST_TcPfw_TempToHmi_Itf[...].Error and ST_TcPfw_TempToHmi_Itf[...].ErrorId.
  - The associated zone is then switched off.
  - By activating the error heating in the machine parameters, the zone is further kept at temperature.
- Error messages during autotune:
  - Are passed on via ST_TcPfw_TempToHmi_Itf[...].Error and ST_TcPfw_TempToHmi_Itf[...].ErrorId.
- Misbehavior during heating:
  - If the actual value of a zone does not react to the setpoint change during heating up, an error is output and the zone switches off.

3.7.3 Heating current monitoring

In the Plastics Processing Framework, heating tape monitoring is called "Looptest". This allows to detect the following malfunctions of a heating zone:

- Condition of the actuators
  - Shorted Solid State Relays (SSR)
  - "Sticking" contactors
- Condition of the heating tapes
  - Partial and total failure
  - Performance losses due to aging

Plastics machines are sometimes designed to contain several temperature control groups (extruders1..n, heating channels, etc.). These are partly supplied separately. These supply units are called supply groups in the framework. Each supply group consists of four supply lines and one power measurement terminal. Several heating zones can be connected to one supply line. Each heating zone contains an actuator (solid state relay - SSR or contactor), one or more heating tapes and a temperature sensor. The first supply line of a group represents the supply for the heating zones, which are connected between L1 and the neutral conductor. The second for L2 and N and the third for L3 and N. The fourth supply line includes all heating zones that are connected between two outer conductors in star or delta.

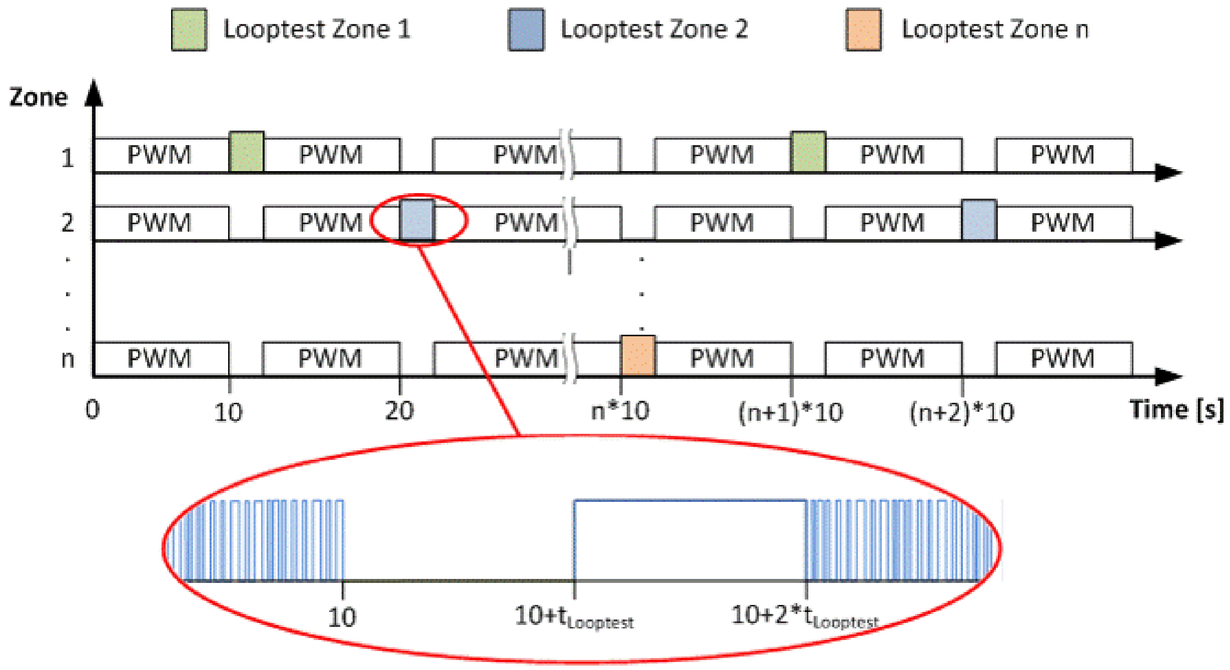


Supply-Group	1				2				...	n			
Supply-Line	1	2	3	4	5	6	7	8	...	(n*4)-3	(n*4)-2	(n*4)-1	(n*4)-0

In the parameters of a heating zone, it is now possible to set in which supply line the zone is located and thus to which supply group it belongs. Furthermore, the nominal output of the heating element can be set in the parameters. This information is necessary so that the Looptest can assign the zone, test it and interpret its measurement.

In the Plastic Processing Framework, there is one loop test per supply group. This passes through all heating zones. If it finds a zone that belongs to its supply group, it starts a measurement. For this purpose, all elements belonging to this group are switched off and the power is determined. This should be zero. If this is not the case, an actuator is defective. After a time $t_{Looptest}$ (default setting 200 ms) has elapsed, the heating element is switched on for the same time and the power is determined. This is compared with the nominal output of the heating zone. If it deviates by a set percentage value, an error is displayed. After that, the control is active for 10 seconds. The self-generated additional heating power is taken into account in the controller. After that, the next zone of the same supply group is searched.

The following figure illustrates this with an example. It is assumed that a plant has only one supply group, for example an extruder with n zones. The loop test is active and finds zone 1. As described before, all zones are switched off first and the power is measured. Zone 1 is then switched on for $t_{Looptest}$ and the power is measured again. In a cycle of 10 seconds, the next zone is tested using the same procedure.



The EL3403 calculates the power over an interval of 10 periods by default. The calculated power is then filtered using low-pass and high-pass filters. In addition, solid state relays, especially those that switch in zero crossing, have a not inconsiderable switching time. This leads to the fact that in some cases a usable power is measured only after 500 ms. With an interval of one period, the time could be reduced to 200 ms. Nevertheless, measuring errors of around 10% can occur under certain circumstances. If a correspondingly longer measuring time (e.g. 400 ms) is kept available, the accuracy increases considerably. An advantage of the xL3403 terminals is that the calculation of the power takes place within the terminal and thus no PLC computing time is required.

Implementation:

The following points must be observed for the implementation:

- FB_PowerMeasurement_TcPfw: This function block must be instantiated according to the number of terminals.
- ST_TcPfw_PowerMeasurement_Cfg: Must be created as ARRAY according to the terminal number.
- ST_TcPfw_PowerMeasurment_Ctrl: Must be applied as ARRAY according to the terminal number.
- ST_TcPfw_xL3403_State: Must be applied as ARRAY according to the terminal number. Even if no xL3403 is in use.

The mapping structures provided by the library for mapping to the power measurement terminal must be used and passed to the FB_PowerMeasurement_TcPfw function block as an address.

NOTE

The pointer address and the stored terminal type must match at all times. Otherwise there will be wrong memory accesses.

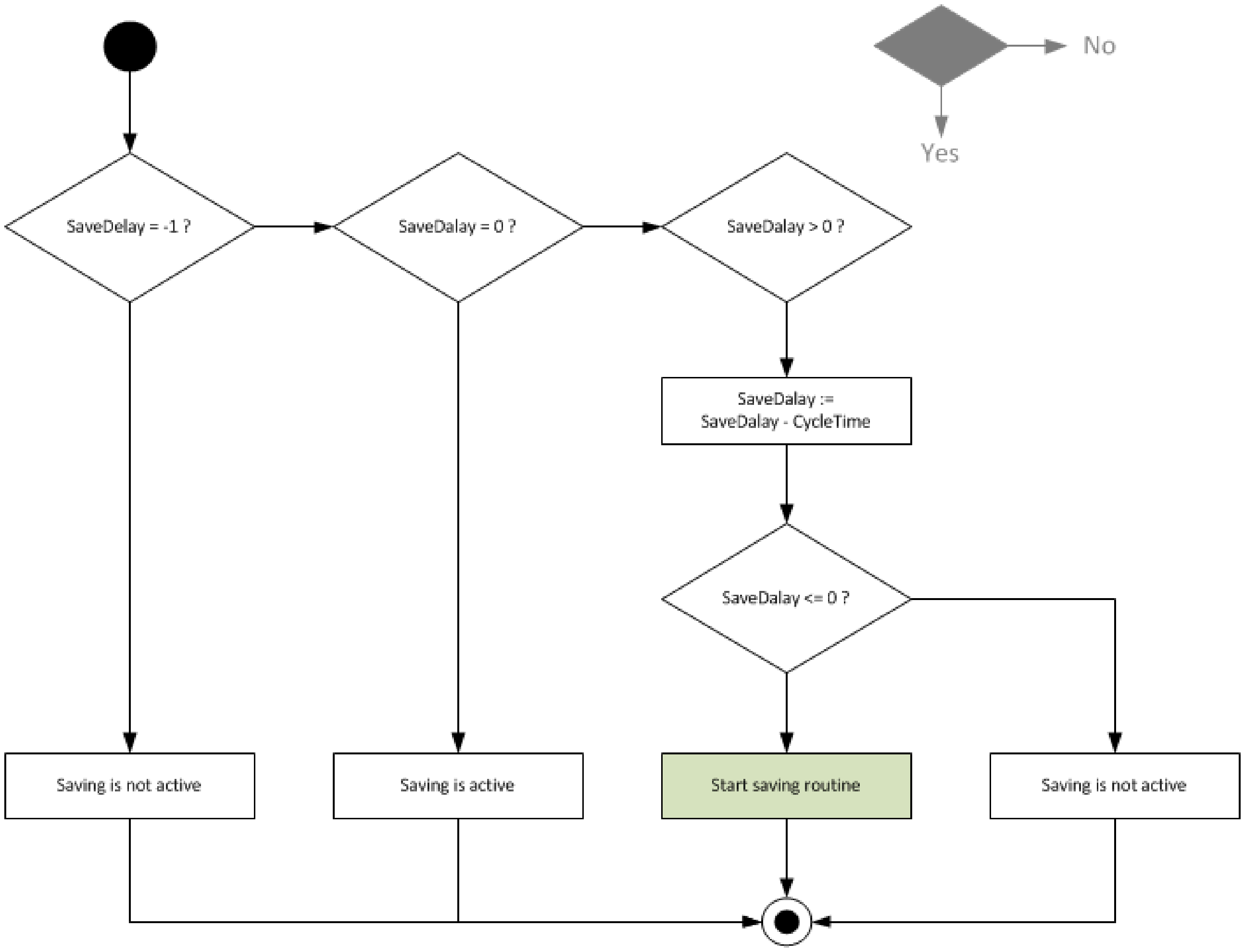
3.7.4 Parameters saving/loading

The structures ST_TcPfw_TempMparamFromHmi_Itf, ST_TcPfw_TempPparamFromHmi_Itf, ST_TcPfw_SupplyParam contain parameters that must be stored remanently. The following variables are stored in each structure for this purpose:

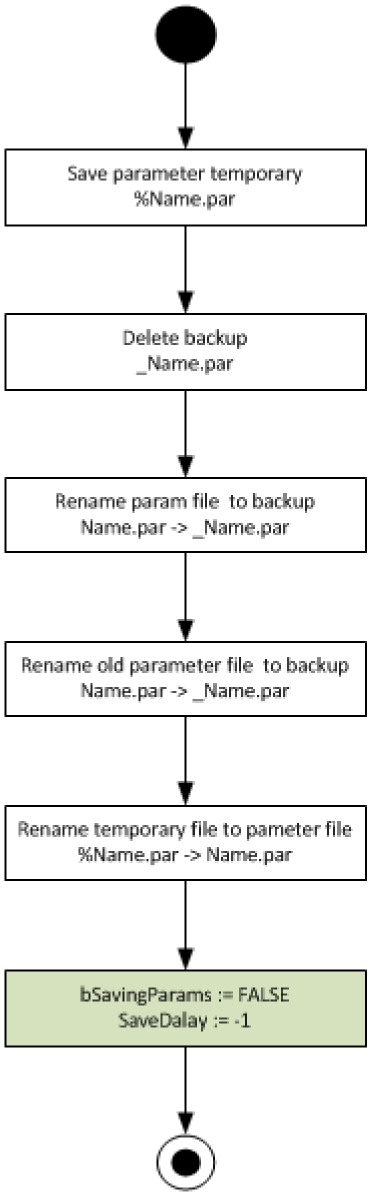
- **SaveDelay:** Time-delayed saving. After a parameter change, a time-delayed saving can be performed here via a number in milliseconds. Based on the entered value, the library automatically counts down this number to zero. At zero the saving routine is triggered internally and at -1 the saving delay is inactive.
- **bLodParams:** Via a TRUE the parameters of the corresponding structure can be loaded

- **bSavingParams:** Feedback that parameters are being saved.

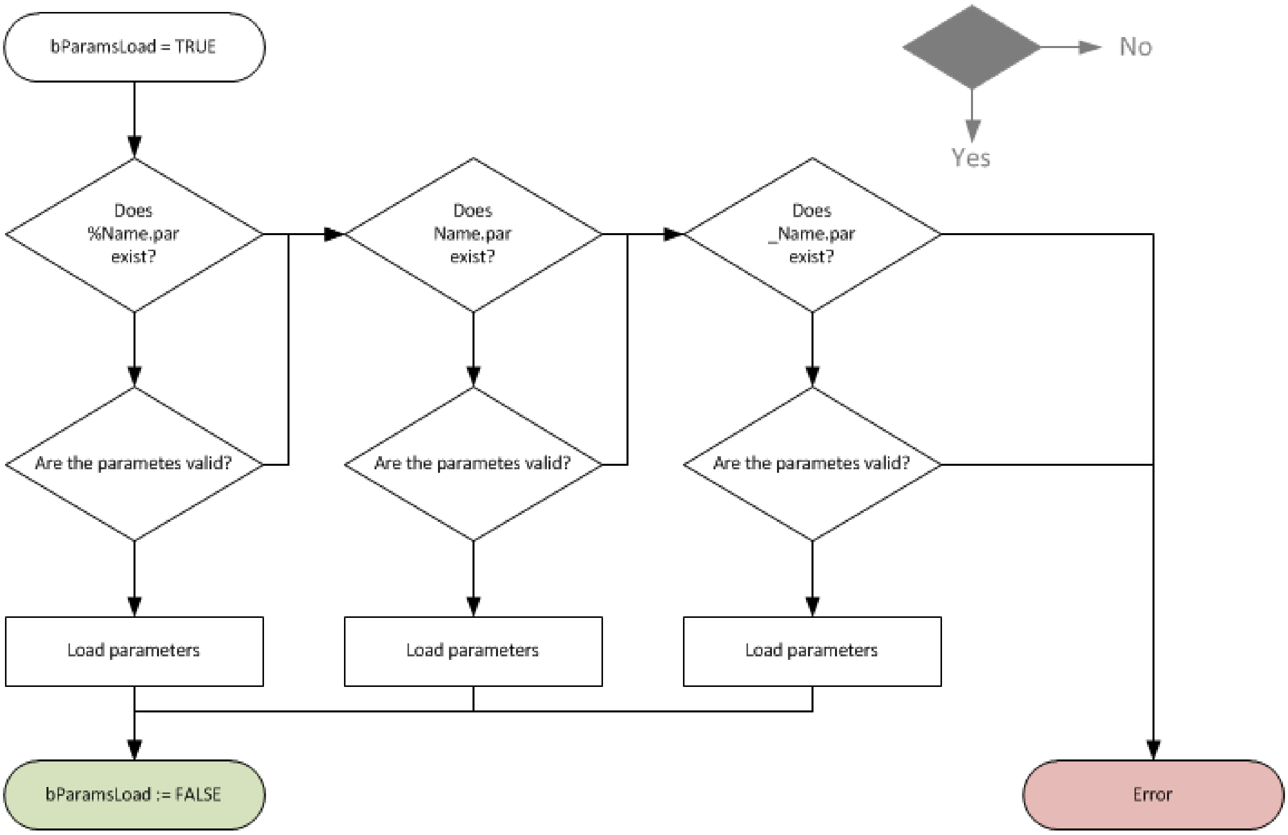
The following flow chart should make clear how the saving process works.



During saving, a temporary file is created first. Then the backup file is deleted and the original file is converted to a backup file. As a final step, the temporary file is converted into an "original file".
This mechanism ensures that there is always a completely saved parameter file.



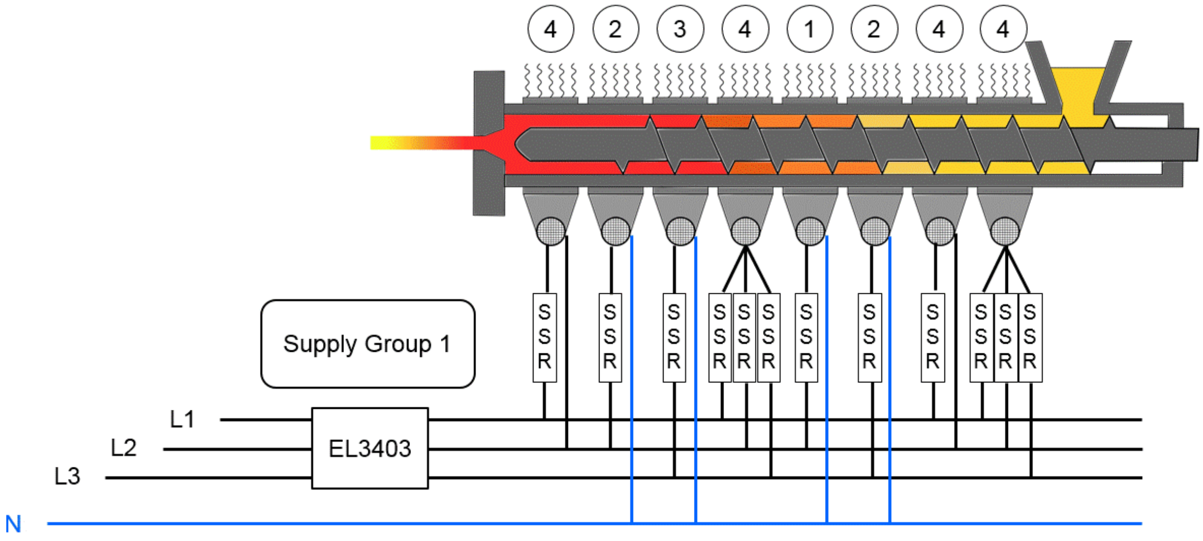
When loading a product via bLoadParams the following process checks whether the saved file is consistent.



3.7.5 Supply groups

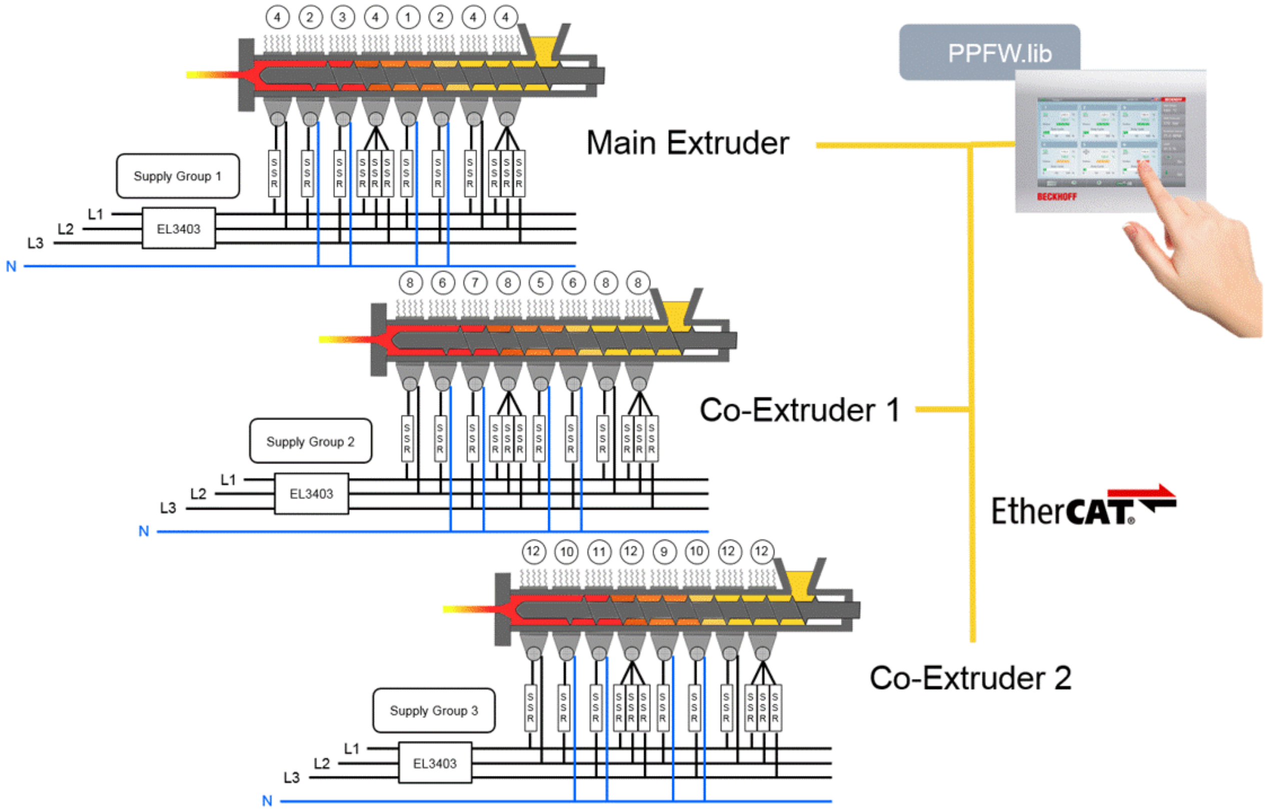
A supply line represents a supply unit. For heating power monitoring and for distributed switch-on (zoning), it must be known which zone is connected to which supply line.

As a rule, the supply consists of one, two or more phases of a three-phase network. Heating tapes connected between phase1 and neutral conductor must get a 1 in the MParam.SupplyLine. Heating tapes connected between phase 2 and neutral conductor are assigned a 2 in the Mparam.SupplyLine. Heating tapes connected between phase 3 and neutral conductor are assigned a 3 in the Mparam.SupplyLine. Multiphase heating tapes (two-phase or three-phase) are assigned a 4.

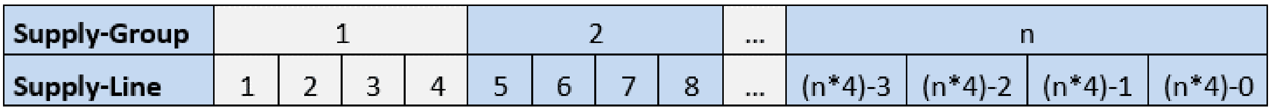


Heating tape	Supply Line
L1-N	1
L2-N	2
L3-N	3
Multiphase (L1 - L2; L2-L3; L1-L2-L3; etc)	4

If there is more than one supply for the heaters, there must be at least one power measurement terminal for each supply. This can be the case, for example, with several extruders that are operated on one control system.



The numbering of the subsequent supply groups (supplygroups) is consecutive. I.e. Supplygroup 2 has assigned SupplyLine 5,6,7,8.



Supply-Group	1				2				...	n			
Supply-Line	1	2	3	4	5	6	7	8	...	(n*4)-3	(n*4)-2	(n*4)-1	(n*4)-0

The constant cnPfwAppSupplyLast must be at least equal to the largest value of the Mparam.SupplyLine. This constant defines the size of the TempCtrl_ST_TcPfw_SupplyParam array. The SupplyParams represent essential properties of the output generation, like the PWM cycle time, minimum and maximum switch-on time.

3.7.6 FAQs

Name	Description
FAQ#1 [▸ 78]	What are the hardware requirements?
FAQ#2 [▸ 78]	What are the software requirements?
FAQ#3 [▸ 78]	Can a zone heat in case of sensor failure?
FAQ#4 [▸ 79]	How can I redirect the output of heating and cooling signals?
FAQ#5 [▸ 79]	How can I redirect the acquisition of the actual temperature?
FAQ#6 [▸ 79]	How does the Idle Load Tuning work?
FAQ#7 [▸ 79]	How does extruder compensation work?
FAQ#8 [▸ 79]	What should I do in case of malfunction?

FAQ#1: Hardware requirements

An IPC with x86 architecture and hardware FPU. Computing power requirements and fieldbus support must be clarified in each case. Examples:

- DIN-rail PCs of the types CX1010, CX1020 or CX1030.
- DIN-rail PCs of the types CX50xx.
- DIN-rail PCs of the types CX51xx.
- DIN-rail PCs of the types CX20xx.
- Control cabinet PCs of the families C41xx, C61xx, C62xx, C69xx.
- Panel PCs of the families CP62xx, CP72xx, CP67xx, CP22xx, CP32xx and CP37xx.
- Other PC families are usually suitable as well. This must be clarified in each individual case.

Alternatively, an IPC with iXP architecture and FPU emulation. Here, there may be restrictions on the number of zones and/or the cycle time.

- DIN-rail PCs of the types CX9010 or CX9020.
- Panel PCs of the families CP66xx and CP26xx.

A fieldbus architecture with suitable performance:

- Preferably EtherCAT
- Alternatively, one of the following fieldbuses (cycle times of 10 ms or less should be aimed for):
  - LightBus
  - RT Ethernet
  - Profibus with 12 MBaud

FAQ#2: Software requirements

A Microsoft operating system of the type Windows CE 6, Windows Embedded Compact 7, Windows XP, Windows XPe or Microsoft Windows Embedded Standard 7, 32-bit.

An executable and licensed TwinCAT system, released for at least TwinCAT PLC.

A licensed copy of the Plastics Temperature Control Framework Library PfwLib_TempControl.LIB, version V1.0.1 or higher.

A provided Plastics Framework AppExtension project PfwLib_TempControlAppExtension.PRO whose version matches the Plastics Temperature Control Framework Library version.

FAQ#3: Heating in case of sensor failure

In the event of a temperature sensor failure, it may be necessary to continue operating the machine. In such a case, the library can be made to estimate the power demand of the zone and output the corresponding heating signal. A number of preconditions have to be met:

- The signal type must be set to eTcPfwTcOut_PWM.
- The zone must be fully commissioned beforehand. Above all, the IdleLoad Tuning must have been done with sufficient accuracy.
- This option should not be enabled in adjacent zones. Otherwise, the risk of overheating of the heating tapes and the material increases due to the uncontrolled output of the power.

NOTE

This option must never be activated in zones that receive excess energy (friction, discharge from neighboring zones) from the process during operation.

To enable the option, set EnableErrorHeating to TRUE in the machine data. A value in the range 0...1 in ErrorHeatingFactor can be used to specify which portion of the estimated power should be output.

If the set temperature of the zone is changed, ErrorHeatingFactor does not need to be adjusted. The estimation will automatically adjust the required power.

FAQ#4: Redirecting heating and cooling signals

If a digital output of the PLC is defective, the operator can connect this digital output to another free output and then redirect the corresponding signal to the other output in the HMI. In order to use this feature, various settings must be made in the temperature library. Since the commissioning of the I/O redirection is somewhat more sophisticated, contact the support in this case. If you do not want to use this feature, the settings in the step-by-step commissioning are to be applied.

FAQ#5: Redirect actual temperature acquisition

If an analog input of the PLC is defective, the operator can connect this analog input to another free input and then redirect the corresponding signal to the other input in the HMI. In order to use this feature, various settings must be made in the temperature library. Since the commissioning of the I/O redirection is somewhat more sophisticated, contact the support in this case. If you do not want to use this feature, the settings in the step-by-step commissioning are to be applied.

FAQ#6: The Idle Load Tuning

The prerequisite for determining the base load is:

- the system must be stable (control deviation of less than 1 °C)
- the setpoint must be greater than 70 °C
- there must be no disturbances acting on the system (rotating screw).

The actual value is not excited during base load determination. Only internal calculations take place. The determination takes several seconds, after which the determined value is automatically included in the control.

FAQ#7: Extruder compensation

Prerequisite for determining the extruder compensation:

- the system must be stable (control deviation of less than 1 °C)
- the screw speed must have reached the working speed

After the extruder compensation has been determined, it is not active. Only when the parameter ExtruderCompEna from the machine parameters receives a TRUE, it also becomes effective. With correct extruder compensation, the actual temperature behaves much more smoothly when the screw is switched on or off than without compensation.

FAQ#8: Behavior in case of malfunction

Take screenshots of the machine and product parameters.

Record the behavior using the scope.

3.7.7 Global variables

Error Codes

Name	hex	dec	Description	Error-heating
dwTcPfwTempErrNoError	16#0000	0	No error.	
dwTcPfwTempErrIllegalValue	16#0706	1798	Parameter has a value that is not allowed. Example: TempSensTerm is not allowed. The selected sensor does not match the selected terminal.	x
dwTcPfwTempErrBusy	16#0708	1800	Terminal is already active. FB_TempCtrlCallback_TcPfw() reports this error if it finds the communication path busy.	x
dwTcPfwTempTuneErr_NoSigType	16#1001	4097	During autotuning this error is reported if OutputSel_H does not select an active signal.	
dwTcPfwTempTuneErr_Parameter	16#1002	4098	In autotuning, this error is reported when the tuning parameters are not allowed.	
dwTcPfwTempTuneErr_NoTravel	16#1003	4099	During autotuning, this error is reported if the setpoint step-change is <25 °C.	
dwTcPfwTempTuneErr_NoSettling	16#1004	4100	During autotuning, this error is reported if the actual temperature of the zone is not stable before the heating test (actual temperature fluctuates by >2 °C).	
dwTcPfwTempTuneErr_Aborted	16#1005	4101	This error is reported when autotuning has been aborted by operator intervention.	
dwTcPfwTempTuneErr_ShortOfPoints	16#1006	4102	During the heating or cooling test, an insufficient number of measuring points was determined.	
dwTcPfwTempTuneErr_NoTop	16#1007	4103	When evaluating the measuring points of a heating or cooling test, no inflection point (= point of maximum slope) was determined.	
dwTcPfwTempTuneErr_NoResponse	16#1008	4104	The measured values cannot be evaluated. The zone has not responded or the measured values are heavily disturbed.	
dwTcPfwTempErrNotSupport	16#4107	16647	FB_TempCtrlCallback_TcPfw() reports this error if the terminal used does not support a required functionality. Example: The linked terminal is not of the specified type.	
dwTcPfwTempErrSnsUnderrun	16#4450	17488	The terminal reports an undershooting of the measuring range.	x
dwTcPfwTempErrSnsOverrun	16#4451	17489	The terminal reports that the measuring range has been exceeded.	x
dwTcPfwTempErrSnsHdwFailed	16#4464	17508	The terminal reports an internal fault.	x
dwTcPfwTempErrDisconected	16#4FF0	20464	The connection to the terminal is interrupted. **Not all fieldbuses and I/O devices support connection monitoring.**	x
dwTcPfwTempErr_NoResponse	16#4FF1	20465	Actual temperature does not respond to heating.	x
dwTcPfwTempErrNotOperational	16#4FF2	20466	The terminal is not in an operable state.	x
dwTcPfwTempErrIoSwapCollision	16#4FF3	20467	I/O redirection causes a collision. The SensTermSwapIdx variable in the machine parameters is probably not configured correctly.	
dwTcPfwTempErrAdsSwapCollision	16#4FF4	20468	TermChannel redirection causes an ADS collision. The SensTermSwapIdx variable in the machine parameters is probably not configured correctly.	

Name	hex	dec	Description	Error-heating
dwTcPfwTempErrInheritedFault	16#4FF5	20469	Hardware problem in another channel.	
dwTcPfwTempErrOverCurrent	16#4FFE	20478	This error is generated when the deviation between the actual heating power and the set heating power is greater than the specified tolerance. See power measurement	
dwTcPfwTempErrUnderCurrent	16#4FFF	20479	reserved.	x
dwTcPfwTempErrSnsCommFailed	16#5000	20480	reserved.	x

Errorheating: if errorheating is active, heating power continues to be output for errors marked with x and the zone is kept at temperature.

Declared global constants

Name	Description
cnv_SupplyParam_TcPfw	Current version identifier of the ST_TcPfw_SupplyParam structure.
cnv_TempCtrl_Itf_TcPfw	Current version identifier of the ST_TcPfw_TempCtrl_Itf structure.
cnv_TempCtrlInput_TcPfw	Current version identifier of the ST_TcPfw_TempCtrlInput structure.
cnv_TempCtrlOutput_TcPfw	Current version identifier of the ST_TcPfw_TempCtrlOutput structure.
cnv_TempMparamFromHmi_TcPfw	Current version identifier of the ST_TcPfw_TempMparamFromHmi_Itf structure.
cnv_TempMparamFileVers	Only for stand-alone operation: Current version identifier of the structure of the optional parameter file.
cnv_TempPparamFromHmi_TcPfw	Current version identifier of the ST_TcPfw_TempPparamFromHmi_Itf structure.
cnv_TempToHmi_TcPfw	Current version identifier of the ST_TcPfw_TempToHmi_Itf structure.

Undeclared global constants

Name	Description	Type	Recommendation value/ maximum values
cnPfwTempCtrlFirst	Initial index to set the size of the array and thus the number of zones that can be controlled.	INT	1 (recommended)
cnPfwTempCtrlLast	End index to set the size of the array and thus the number of zones that can be controlled.	INT	512 (depending on CPU)
cnPfwScopeSampleFirst	Initial index to set the size of the array and thus the number of memory points in the scope.	INT	1 (recommended)
cnPfwScopeSampleLast	End index to set the size of the array and thus the number of memory points in the scope.	INT	32767 (max.)
cnPfwTempTrendFirst	Initial index to set the size of the array and thus the number of memory points in the trend.	INT	1 (recommended)
cnPfwTempTrendLast	End index to set the size of the array and thus the number of memory points in the trend.	INT	32767 (max.)
cnPfwAppSupplyFirst	Initial index to set the size of the array and thus the number of supply groups (usually 4; 1=phase 1, 2=phase 2, 3=phase 3 and 4 for multi-phase heating tapes).	INT	1
cnPfwAppSupplyLast	End index to set the size of the array and thus the number of supply groups (usually 4; 1=phase 1, 2=phase 2, 3=phase 3 and 4 for multi-phase heating tapes).	INT	4
cnPfwBoolOutSwapFirst	Initial index of the array out_SwappedDigitalOut (I/O redirection).	INT	1
cnPfwBoolOutSwapLast	End index of the array out_SwappedDigitalOut (I/O redirection).	INT	2 (when not in use)
cnPfwBoolInSwapFirst	Initial index of the array in_SwappedDigitalIn (I/O redirection).	INT	1
cnPfwBoolInSwapLast	End index of the array in_SwappedDigitalIn (I/O redirection).	INT	2 (when not in use)

Global variables of the framework

Name	Description
aaaPfwTempMparamFromHmi	The machine parameters of the zones.
aaaPfwTempPparamFromHmi	The product parameters of the zones.
aaaPfwTempToHmi	The interfaces of the zones to the HMI.
aaaTempAlarm_AbsoluteHigh	Collective message: Exceeding of the alarm threshold by at least one zone.
aaaTempAlarm_AbsoluteLow	Collective message: At least one zone falls below the alarm threshold.
aaaTempAlarm_High	Collective message: Exceeding of the positive inner tolerance threshold by at least one zone.
aaaTempAlarm_HighHigh	Collective message: Exceeding of the positive outer tolerance threshold by at least one zone.
aaaTempAlarm_Low	Collective message: Exceeding of the negative inner tolerance threshold by at least one zone.
aaaTempAlarm_LowLow	Collective message: Exceeding of the negative outer tolerance threshold by at least one zone.
aaaTempFault_Reset	Collective command: Reset of possible error states for all zones.
bPfwTempLinksInitDone	reserved
in_PfwTempCtrlInput	The input process images of the zones.
out_PfwTempCtrlOutput	The output process images of the zones.
stPfwSupplyLineCfg	The parameters of the supply groups.
stPfwTempCtrl	The runtime data of the zones.
out_SwappedDigitalOut	For the redirection of digital output signals. This array provides the interface for linking the digital outputs. (I/O redirection)
in_SwappedDigitalIn	For redirecting digital input signals. This array provides the interface for linking the digital inputs. (I/O redirection)

4 PLC timer

4.1 Overview

Via the weekly timer it is possible to automatically transfer zones to different operating states. Memory and load routines are provided so that the set times are also available after a restart.

The following table gives an overview of the provided function blocks and their meaning.

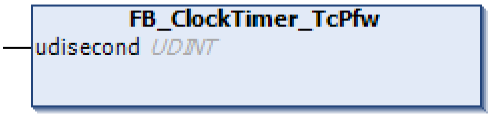
Name	Description
FB_ClockTimer_TcPfw() [▶ 85]	Switches the outputs on or off according to the set time.
FB_ClockTimerParamSave_TcPfw() [▶ 85]	Saves the timer settings.
FB_ClockTimerParamLoad_TcPfw() [▶ 86]	Loads the timer settings.

Data types: Structure types

Name	Description
ST_TcPfw_ClockTimerCam [▶ 88]	Is a substructure of the structure ST_TcPfw_ClockTimerItf. This is used to enter the switch-on and switch-off time.
ST_TcPfw_ClockTimerItf [▶ 87]	The structure contains the interfaces for setting the timer.

4.2 Function blocks

4.2.1 FB_ClockTimer_TcPfw()



The actual timer is implemented in this function block. It must be called cyclically and enables or disables the corresponding output ST_TcPfw_ClockTimerItf.Q in the set time.

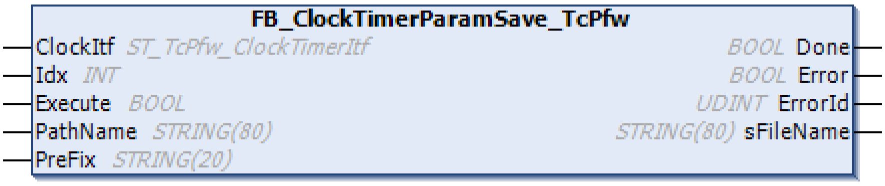
Syntax

```
VAR_INPUT
    udisecond    : UDINT:=0;
END_VAR
```

Inputs

Name	Type	Description
udisecond	UDINT	Time of the task in microseconds

4.2.2 FB_ClockTimerParamSave_TcPfw()



This function block saves the current settings of the timer as a binary file.

Syntax

```
VAR_IN_OUT
    ClockItf   : ST_TcPfw_ClockTimerItf;
END_VAR
VAR_INPUT
    Idx: INT;
    Execute    : BOOL:=FALSE;
    PathName   : STRING(80);
END_VAR
VAR_OUTPUT
    Done       : BOOL:=FALSE;
    Error      : BOOL:=FALSE;
    ErrorId    : UDINT:=0;
END_VAR
```

Inputs

Name	Type	Description
Idx	INT	Timer number
Execute	BOOL	The storage process is started with a rising edge.
PathName	STRING	Path name where to save the timer settings.

Inputs/outputs

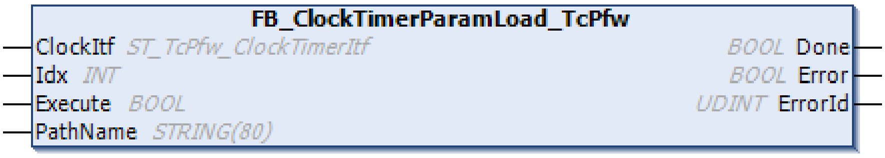
Name	Type	Description
ClockItf	ST_TcPfw_ClockTimerItf	

Outputs

Name	Type	Description
Done	BOOL	Feedback that saving was successful.
Error	BOOL	Error while saving timer settings.
ErrorId	UDINT	Error number

A positive edge at the Execute input activates the saving process. After a successful saving process, a Done is present for one cycle. If loading is not successful, an error with error number is returned.

4.2.3 FB_ClockTimerParamLoad_TcPfw()



This function block loads the saved settings of the timer.

Syntax

```
VAR_IN_OUT
    ClockItf: ST_TcPfw_ClockTimerItf;
END_VAR
VAR_INPUT
    Idx      : INT;
    Execute :  BOOL:=FALSE;
    PathName: STRING(80);
END_VAR
```

```
VAR_OUTPUT
    Done    : BOOL:=FALSE;
    Error   : BOOL:=FALSE;
    ErrorId : UDINT:=0;
END_VAR
```

Inputs

Name	Type	Description
Idx	INT	Timer number
Execute	BOOL	The loading process is started with a rising edge.
PathName	STRING	Path name where the timer settings can be loaded.

/Inputs/outputs

Name	Type	Description
ClockItf	ST_TcPfw_ClockTimerItf	Current settings of the timer as binary file.

Outputs

Name	Type	Description
Done	BOOL	Feedback that loading was successful.
Error	BOOL	Error while loading timer settings.
ErrorId	UDINT	Error number

Behavior of the function block:

A positive edge at the Execute input activates the loading process. After a successful loading process, a Done is present for one cycle. If loading is not successful, an error with error number is returned.

4.3 Structures

4.3.1 ST_TcPfw_ClockTimerItf

This structure must be declared in the global variables of the application. The structure contains the interfaces for setting the timer.

In this structure the day of the week is selected as well as the time; when the timer is active and when it is inactive.

Syntax

```
TYPE ST_TcPfw_ClockTimerItf:
(* location PfwLib_Common.PRO *)
(* last modification: 08.07.2008 *)
STRUCT
(*
see cnv_ClockTimerItf_TcPfw for format information
*)
    Q  : ARRAY[cnPfwClockTimerCamFirst..cnPfwClockTimerCamLast] OF BOOL:=FALSE;

    Day: ARRAY[1..7,cnPfwClockTimerCamFirst..cnPfwClockTimerCamLast] OF ST_TcPfw_ClockTimerCam;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
Q	ARRAY OF BOOL	Is a one-dimensional array in which the number of outputs per timer is specified. This allows a zone to assume up to 9 different operating states per day.
Day	ARRAY OF ST_TcPfw_ClockTimerCam	Is a two-dimensional array in which the first number indicates the day of the week (1=Monday, 2=Tuesday, ..., 7=Sunday) and the second number indicates the number of the timer.

4.3.2 ST_TcPfw_ClockTimerCam

Is a substructure of the structure ST_TcPfw_ClockTimerItf. This is used to enter the switch-on and switch-off time.

Syntax

```
TYPE ST_TcPfw_ClockTimerCam:
(* location PfwLib_Common.PRO *)
(* last modification: 05.06.2008 *)
STRUCT
(*
see cnv_ClockTimerCam_TcPfw for format information
*)
    On  : ARRAY[1..3] OF INT;
    Off : ARRAY[1..3] OF INT;
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
On	ARRAY OF INT	After the time entered in On by the operator, the corresponding output becomes active. In the first place On[1] is the hour with the value range 0...23, in the second place On[2] the minute with the value range 0...59 and in the third place On[3] the second with the value range 0...59.
Off	ARRAY OF INT	After the time entered in Off by the operator, the corresponding output becomes inactive. In the first place On[1] is the hour with the value range 0...23, in the second place Off[2] the minute with the value range 0...59 and in the third place Off[3] the second with the value range 0...59.

4.4 Knowledge Base

4.4.1 Commissioning

These commissioning instructions assume that the temperature controller is fully commissioned.

In the first step it is necessary that an array of type ST_TcPfw_ClockTimerItf is created in the global variables. The array size reflects the number of weekly timers. Subsequently, the following constants are to be created:

Constant definition

Variable	Short description	Example value	Maximum values
cnPfwAppClockTimerFirst	Initial index to set the size of the array and thus the number of weekly timers.	1	1
cnPfwAppClockTimerLast	End index to set the size of the array and thus the number of weekly timers.	2	Limited by computing power and memory
cnPfwClockTimerCamFirst	Initial index to set the size of the array and thus the number of channels per weekly timer.	1	1
cnPfwClockTimerCamLas3	End index to set the size of the array and thus the number of channels per weekly timer.	3	9

Parameterization

In order to transfer the correct time to the timer, the following structure must be observed:

The structure ST_TcPfw_ClockTimerItf is divided into Q and Day, where Day is a two-dimensional array. The first index indicates the number of the day of the week (1=Monday, 2=Tuesday,3=Wednesday, 4=Thursday, 5=Friday, 6=Saturday, 7=Sunday) and the second index indicates the number of the corresponding channel. This two-dimensional array is again divided into an On-Array and an Off-Array. Both On-Array and Off-Array has three elements for:

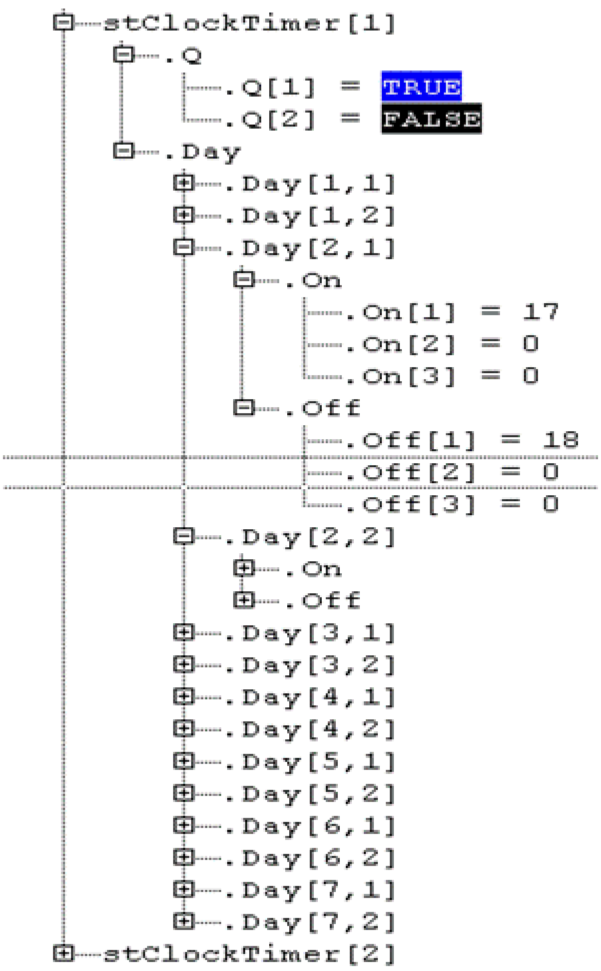
- On/Off[1]=hour in the value range 0...23,
- On/Off[2]=minute in the value range 0...59,
- On/Off[3]=second in the value range 0...59.

This is used to set the switch-on and switch-off time and to set output Q accordingly.

Parameterization

A very simple application has been prepared as a sample project.

- All program lines referring to the weekly timer are marked with (*ClockTimer*).
- If different times are entered into the array during the running program, the corresponding output will be active at the corresponding times.
- **Example:**
    - On Tuesday from 17:00 to 18:00 standby / the output stClockTimer[1].Q[1] must activate the controller in the application.



```
⊟ stClockTimer[1]
    ⊟ .Q
        .Q[1] = TRUE
        .Q[2] = FALSE
    ⊟ .Day
        ⊞ .Day[1,1]
        ⊞ .Day[1,2]
        ⊟ .Day[2,1]
            ⊟ .On
                .On[1] = 17
                .On[2] = 0
                .On[3] = 0
            ⊟ .Off
                .Off[1] = 18
                .Off[2] = 0
                .Off[3] = 0
        ⊟ .Day[2,2]
            ⊞ .On
            ⊞ .Off
        ⊞ .Day[3,1]
        ⊞ .Day[3,2]
        ⊞ .Day[4,1]
        ⊞ .Day[4,2]
        ⊞ .Day[5,1]
        ⊞ .Day[5,2]
        ⊞ .Day[6,1]
        ⊞ .Day[6,2]
        ⊞ .Day[7,1]
        ⊞ .Day[7,2]
⊞ stClockTimer[2]
```

4.4.2 Global variables

Error Codes

No error codes are defined. If an error code is issued, it must be consulted in the Information System because it is an error of another library.

Declared global constants

Name	Description
cnv_ClockTimerParamFileVers	Current version identifier
cnv_ClockTimerItf_TcPfw	Current version identifier of the ST_TcPfw_ClockTimerItf structure.
cnv_ClockTimerCam_TcPfw	Current version identifier of the ST_TcPfw_ClockTimerCam structure.

Undeclared global constants

Name	Description	Maximum values
cnPfwClockTimerCamFirst	Initial index to set the size of the array and thus the number of channels per weekly timer.	1
cnPfwClockTimerCamLast	End index to set the size of the array and thus the number of channels per weekly timer.	Limited by computing power and memory
cnPfwAppClockTimerFirst	Initial index to set the size of the array and thus the number of timers.	1
cnPfwAppClockTimerLast	End index to set the size of the array and thus the number of timers.	9

4.4.3 FAQs

Name	Description
FAQ#1 [▸ 91]	Which time inputs are allowed?
FAQ#2 [▸ 91]	How is the time calculated? Is summer/winter time taken into account? Are leap years taken into account?
FAQ#3 [▸ 91]	How do I activate a time over several days?
FAQ#4 [▸ 91]	Can I specify multiple switching times via a timer?

FAQ#1: Time inputs

A 24h time specification is necessary, i.**e.**:

- Hour: value range 0...23
- Minute: value range 0...59
- Second: value range 0...59

FAQ#2: How is the time calculated?

The time is determined from the time in the computer.

It is important that this time is set correctly.

Since the computer recognizes a summertime/wintertime changeover, this is also recognized in the control.

Since the computer knows the leap years, these are also recognized in the control.

FAQ#3: Activate time over several days

In the structure ST_TcPfw_ClockTimerCam the start time is entered at **On** and at **Off** the time 23:59:59 must be entered. On the following day, 00:00:00 is entered at **On** and **Off** is entered when the system is to be switched off again. In this case, the heating is not turned off for a second at midnight, but remains active.

FAQ#4: Specify multiple switching times via a timer

It is possible to create 9 channels per weekly timer. A switch-on time and a switch-off time are specified for each channel, as well as an output.

5 PLC alarm visualization

5.1 Overview

Ready-made function blocks are provided for displaying, disabling, deleting and managing messages (alarms, warnings, etc.). The temperature control itself only provides an ErrorID and an error flag in case of an error.

Messages are entered and activated by the function block FB_MsgAppend_TcTvA() in an array of the type ST_TcTvA_Alarm_Itf (alarm history). If the cause of a message has been eliminated, this message can be disabled via **Reset**. The deactivation of a message takes place via the function block FB_MsgDeactivate_TcTvA(). Afterwards the message can be deleted by calling the function block FB_MsgClearPending_TcTvA() in the alarm history. So that no empty lines are created in the alarm history by deleted messages, the function block FB_MsgGarbageCollect_TcTvA() must be called cyclically.

Data types: function blocks

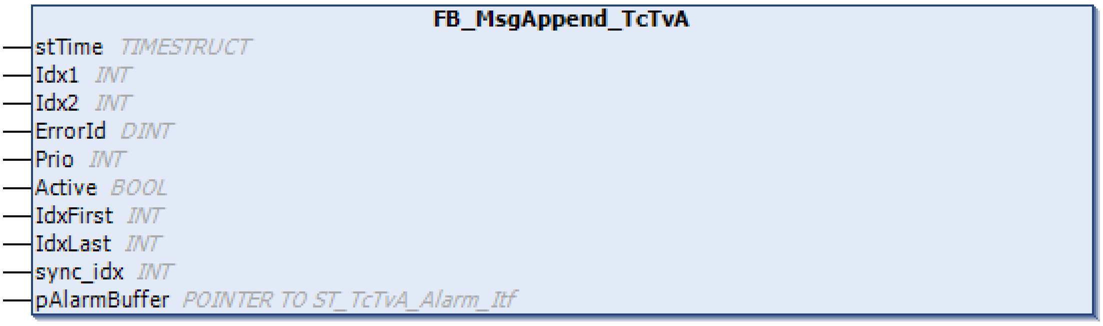
Name	Description
FB_MsgAppend_TcTvA() [▸ 92]	Activates an active error message in the array of type ST_TcTvA_Alarm_Itf.
FB_MsgClearPending_TcTvA() [▸ 93]	This function block deletes inactive errors from the error history (array of type ST_TcTvA_Alarm_Itf).
FB_MsgClearSignal_TcTvA() [▸ 94]	This function block deletes the message.
FB_MsgDeactivate_TcTvA() [▸ 95]	Disables an inactive error message in the array of type ST_TcTvA_Alarm_Itf.
FB_MsgGarbageCollect_TcTvA() [▸ 95]	Must be called cyclically and re-sorts the error history (array of type ST_TcTvA_Alarm_Itf).
FB_MsgUpdateTime_TcTvA() [▸ 96]	This function block determines the Windows system time and returns it to the application.

Data types: Structure types

Name	Description
ST_TcTvA_Alarm_Itf [▸ 97]	Creating an array of this structure results in the alarm history.

5.2 Function blocks

5.2.1 FB_MsgAppend_TcTvA()



FB_MsgAppend_TcTvA

```
stTime  TIMESTRUCT
Idx1  INT
Idx2  INT
ErrorId  DINT
Prio  INT
Active  BOOL
IdxFirst  INT
IdxLast  INT
sync_idx  INT
pAlarmBuffer  POINTER TO ST_TcTvA_Alarm_Itf
```

This function block appends active alarms to an array of type ST_TcTvA_Alarm_Itf. To get as much information as possible about the error in the visualization, the parameters listed below can be given to the error message.

Syntax

```
VAR_IN_OUT
    stTime       : TIMESTRUCT;
END_VAR
VAR_INPUT
    Idx1         : INT:=0;
    Idx2         : INT:=0;
    ErrorId      : DINT:=0;
    Prio         : INT:=0;
    Active       : BOOL:=TRUE;

    IdxFirst     : INT;
    IdxLast      : INT;
    sync_idx     : INT:=0;
    pAlarmBuffer: POINTER TO ST_TcTvA_Alarm_Itf;
END_VAR
```

Inputs

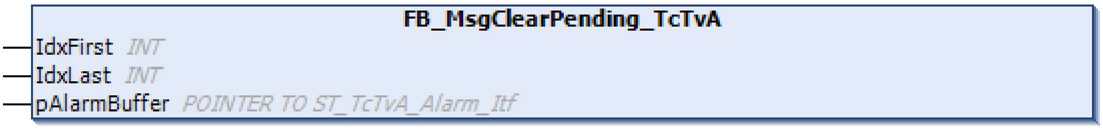
Name	Type	Description
Idx1	INT	Zone number or drive number of the faulty zone.
Idx2	INT	Module number of the faulty zone.
ErrorId	DINT	If necessary, the error number can also be read as a plain text message from an .xml file.
Prio	INT	The priority of the error, where 3=Alarm 2=Warning 1=Note 0=Empty.
Active	BOOL	Active indicates that the error is active.
IdxFirst	INT	The first index in the alarm buffer.
IdxLast	INT	The last index in the alarm buffer.
sync_idx	INT	The address of the alarm buffer with the type ST_TcTvA_Alarm_Itf.
pAlarmBuffer	POINTER TO ST_TcTvA_Alarm_Itf	The address of the alarm buffer with the type ST_TcTvA_Alarm_Itf.

Inputs/outputs

Name	Type	Description
stTime	TIMESTRUCT	The time when the error occurred.

After a successful call of this function block, there must be another entry at the first free position in the alarm history.

5.2.2 FB_MsgClearPending_TcTvA()



This function block deletes all inactive messages from the alarm history.

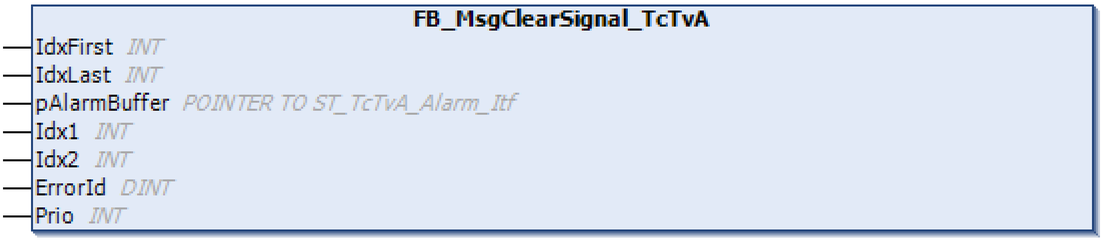
Syntax

```
VAR_INPUT
    IdxFirst     : INT;
    IdxLast      : INT;
    pAlarmBuffer: POINTER TO ST_TcTvA_Alarm_Itf;
END_VAR
```

Inputs

Name	Type	Description
IdxFirst	INT	The first index in the alarm buffer.
IdxLast	INT	The last index in the alarm buffer.
pAlarmBuffer	POINTER TO ST_TcTvA_Alarm_Itf	The address of the alarm buffer with the type ST_TcTvA_Alarm_Itf.

5.2.3 FB_MsgClearSignal_TcTvA()



This function block disables the boolean expression ST_TcTvA_Alarm_Itf.Signal.

Syntax

```
VAR_INPUT
    IdxFirst: INT;
    IdxLast: INT;
    pAlarmBuffer: POINTER TO ST_TcTvA_Alarm_Itf;

    Idx1    : INT:=0;
    Idx2    : INT:=0;
    ErrorId: DINT:=0;
    Prio    : INT:=0;
END_VAR
```

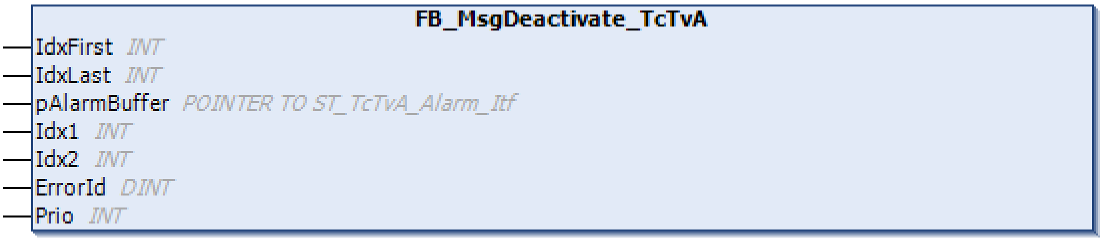
Inputs

Name	Type	Description
IdxFirst	INT	The first index in the alarm buffer.
IdxLast	INT	The last index in the alarm buffer.
pAlarmBuffer	POINTER TO ST_TcTvA_Alarm_Itf	The address of the alarm buffer with the type ST_TcTvA_Alarm_Itf.
Idx1	INT	Zone number of the faulty zone.
Idx2	INT	Module number of the faulty zone.
ErrorId	DINT	The error number
Prio	INT	The priority of the error, where 3=Alarm 2=Warning 1=Note 0=Empty.

Behavior of the function block:

If an alarm with prio:=3 is present in the alarm buffer, this alarm additionally sets the signal ST_TcTvA_Alarm_Itf.Signal:=TRUE. This can be used to control a horn, for example. By calling this function block all ST_TcTvA_Alarm_Itf.Signal:=FALSE, which turns off the message.

5.2.4 FB_MsgDeactivate_TcTvA()



This function block disables an inactive message in the alarm history (type: ST_TcTvA_Alarm_Itf). The message is not deleted from the alarm history.

Syntax

```
VAR_INPUT
Idx1        : INT:=0;
Idx2        : INT:=0;
ErrorId     : DINT:=0;
Prio        : INT:=0;

IdxFirst    : INT;
IdxLast     : INT;
pAlarmBuffer: POINTER TO ST_TcTvA_Alarm_Itf;
END_VAR
```

Inputs

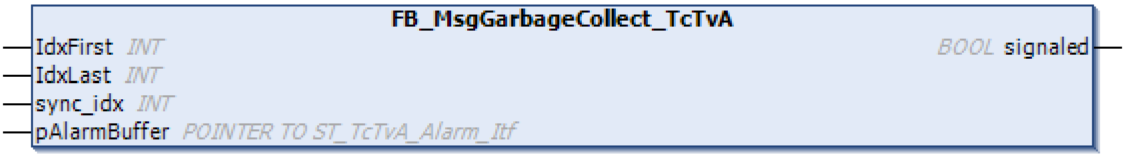
Name	Type	Description
Idx1	INT	Zone number of the faulty zone.
Idx2	INT	Module number of the faulty zone.
ErrorId	DINT	The error number
Prio	INT	The priority of the error, where 3=Alarm 2=Warning 1=Note 0=Empty.
IdxFirst	INT	The first index in the alarm buffer.
IdxLast	INT	The last index in the alarm buffer.
pAlarmBuffer	POINTER TO ST_TcTvA_Alarm_Itf	The address of the alarm buffer with the type ST_TcTvA_Alarm_Itf.

Behavior of the function block:

Each FB_MsgAppend_TcTvA() function block also includes a FB_MsgDeactivate_TcTvA() that resets the message. To ensure that exactly the right message is reset, the deactivation must be called with the same properties (Idx1, Idx2, ErrorID, Prio) as the activation.

After a successful deactivation ST_TcTvA_Alarm_Itf.active:=FALSE. The variable ST_TcTvA_Alarm_Itf.Bitmap depends on ST_TcTvA_Alarm_Itf.Prio and whether ST_TcTvA_Alarm_Itf.active:=TRUE or FALSE.

5.2.5 FB_MsgGarbageCollect_TcTvA()


FB_MsgGarbageCollect_TcTvA
IdxFirst INT
IdxLast INT
sync_idx INT
pAlarmBuffer POINTER TO ST_TcTvA_Alarm_Itf
BOOL signaled

This function block checks whether the alarm history (array of type ST_TcTvA_Alarm_Itf) contains empty cells and moves subsequent messages up.

Syntax

```
VAR_INPUT
IdxFirst    : INT;
IdxLast     : INT;
sync_idx    : INT:=0;
pAlarmBuffer: POINTER TO ST_TcTvA_Alarm_Itf;
END_VAR
VAR_OUTPUT
signaled    : BOOL;
END_VAR
```

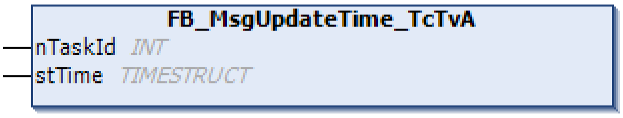
Inputs

Name	Type	Description
IdxFirst	INT	The first index in the alarm buffer.
IdxLast	INT	The last index in the alarm buffer.
sync_idx	INT	Reserved; with 0 the array is automatically searched for spaces.
pAlarmBuffer	POINTER TO ST_TcTvA_Alarm_Itf	The address of the alarm buffer with the type ST_TcTvA_Alarm_Itf.

Outputs

Name	Type	Description
signaled	BOOL	Signals whether a message is present.

5.2.6 FB_MsgUpdateTime_TcTvA()



This function block determines the Windows system time and returns it to the application.

```
VAR_INPUT
    nTaskId: INT;
END_VAR
VAR_IN_OUT
    stTime : TIMESTRUCT;
END_VAR
```

Inputs

Name	Type	Description
nTaskId	INT	Task in which the alarm handling takes place.

Inputs/outputs

Name	Type	Description
stTime	TIMESTRUCT	Windows system time.

5.3 Structures

5.3.1 ST_TcTvA_Alarm_Itf

Creating an array of this structure results in the alarm history. Each new, active and inactive message is temporarily stored in this array.

Syntax

```
TYPE ST_TcTvA_Alarm_Itf :
(* last modification: 04.10.2008 *)
STRUCT
(*
==========================================
message data
see cnv_ItfStructType_TvA for format definition
==========================================
*)
sTime: STRING(23);

ErrorId: DINT:=0;

Count  : INT:=0;
Bitmap : INT:=0; (* 5=AlarmInaktiv 4=AlarmAktiv 3=WarningInaktiv 2=WarningAktiv 1=Note 0=Empty *)
Prio   : INT:=0; (* 3=Alarm 2=Warning 1=Note 0=Empty *)
Idx1   : INT:=0; (* module or drive no. *)
Idx2   : INT:=0; (* zone no. *)
i_align: ARRAY[1..3]OF INT;

Active : BOOL:=FALSE;
Pending: BOOL:=FALSE;
Signal : BOOL:=FALSE;
b_align: ARRAY[1..5]OF BOOL;
(**)
END_STRUCT
END_TYPE
```

Parameter

Name	Type	Description
sTime	STRING	The cycle time (in seconds) of the PWM signal generator.
ErrorId	DINT	Error number (this can be used to read a plain text message from an .XML file).
Count	INT	Number of the alarm memory.
Bitmap	INT	This can be used to visualize the state in an error message. Where 5=Alarm inactive, 4=Alarm active, 3=Warning inactive, 2=Warning active, 1=Note, 0=Empty.
Prio	INT	Indicates the type of alarm. A distinction is made between Alarm=3, Warning=2, Note=1, Empty=0.
Idx1	INT	The module or drive number should be specified here (but can be used arbitrarily).
Idx2	INT	The zone number is to be specified here (but can be used as desired).
i_align	ARRAY OF INT	INTEGER Alignment.
Active	BOOL	By calling the function block FB_MsgAppend_TcTvA() an error is added to the alarm history and activated via this Bool. If the error is corrected, this Boolean is deactivated by the function block FB_MsgDeactivate_TcTvA().
Pending	BOOL	Alarms that are no longer active (Active:=FALSE) can be deactivated via the function block FB_MsgClearPending_TcTvA().
Signal	BOOL	For messages with Prio:=5 additionally Signal:=TRUE, whereby a horn can be activated. By calling FB_MsgClearSignal_TcTvA() Signal:=FALSE.
b_align	ARRAY OF BOOL	BOOL Alignment.

5.4 Knowledge Base

5.4.1 Commissioning

There are two application examples for alarm handling commissioning:

- minimal application to describe only the alarm handling.
- Application with complete temperature library for alarm handling of the temperature library.

In both applications important places for alarm handling are marked by (*TVAlarm*).

Basic structure of the Example_TvAlarm application for alarm handling

The project has the following arrays:

- **stVisuTempAlarm:**
  This array is of type ST_VisuAlarm (an application specific structure) and is used as a buffer in case there are several program parts generating an error at the same time (e.g. several zones on one extruder), as well as for presetting prio, Idx1 and Idx2 to each error number.
- **aaaPfwTempMessageHmi:**
  This array is of type ST_TcTvA_Alarm_Itf and is the array that must be used for visualization. Here the individual errors are listed in order.
- **FB_InitAlarmMessage:**
  In this function block exactly one error code is assigned to each cell of the array stVisuTempAlarm.
- **F_initTempMsg:**
  This function is called in the FB_InitAlarmMessage function block and is responsible for initializing the error code in the stVisuTempAlarm array.
- **FB_MsgCheck:**
  This function block must be called cyclically. All library function blocks are called from here. This means:
  - active messages are entered into the event system as active,
  - inactive messages are entered into the event system as inactive,
  - inactive messages can be deleted.
- **FB_MsgCheckTemp:**
  In this function block all messages that have become active in this cycle are entered at the corresponding positions in the array stVisuTempAlarm.

Since the PfwLib_Processing.lib library was included in this example, various constants had to be declared for the temperature control. But these are unimportant for the actual alarm handling and only serve to compile the project. In the source code these places are marked accordingly.

Basic structure of the Example_TvAlarm_TempCtrl application for alarm handling

Compared to the example above, this alarm handling is much more sophisticated due to the different zones of temperature control and due to the many possible error messages.

The project have the following arrays:

- **stVisuTempAlarm:**
  This array is of type ST_VisuAlarm (an application-specific structure) and serves as a buffer in case several alarms become active in one cycle. In the array you have the possibility to record 30 different errors for one zone.
- **aaaPfwTempMessageHmi:**
  This array is of type ST_TcTvA_Alarm_Itf and must be used for visualization.
- **FB_InitAlarmMessage:**
  In this function block exactly one error code is assigned to each cell of an array stVisuTempAlarm.
- **F_initTempMsg:**
  This function is called in the FB_InitAlarmMessage function block and is responsible for initializing the error code in the stVisuTempAlarm array.

- **FB_MsgCheck:**
  This function block must be called cyclically. All library function blocks are called from here. This means
  - active messages are entered into the event system as active,
  - inactive messages are entered into the event system as inactive,
  - inactive messages can be deleted.
- **FB_MsgCheckTemp:**
  In this function block all messages that have become active in this cycle are entered at the corresponding positions in the array stVisuTempAlarm.

5.4.2 Global variables

Error Codes

No error codes are defined. If an error code is output, it must be consulted in the Beckhoff information system, because it is an error of another library.

Name	Description
cnv_ItfStructType_TvA	Current version identifier of the ST_TcTvA_Alarm_Itf structure.

5.4.3 FAQs

Name	Description
FAQ#1 [▶ 99]	Is a distinction made between alarm, warning and information?
FAQs [▶ 99]	Can other errors, other than those from specific libraries, be logged?

FAQ#1: Is there a distinction between alarm, warning and information?

There are three different message levels, which can be set under ST_TcTvA_Alarm_Itf.Prio:

- prio:=3 means alarm and activates ST_TcTvA_Alarm_Itf.signal at the same time. This output can be linked to a horn, for example.
- prio:=2 is a warning.
- prio:=1 is an information.

FAQ#2: Can other errors besides those from specific libraries be logged as well?

If the additional error codes are included in the alarm management, it is possible to display these messages as well.

More Information:
www.beckhoff.de/tf8540

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